Controlling information in Probabilistic Systems

Engel Lefaucheux

UNIVERSITE BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1
COMUE **U**NIVERSITE **B**RETAGNE **L**OIRE

Ecole Doctorale N°601
Mathèmatique et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

« Engel LEFAUCHEUX »

« Controlling Information in Probabilistic Systems »

Thèse présentée et soutenue à RENNES , le 24 Septembre 2018
Unité de recherche : IRISA, Equipe SUMO

Rapporteurs avant soutenance :
Stefan KIEFER, Associate Professor at Oxford
Antonin KUCERA, Professor at Masaryk University

Composition du jury :

Président :	Sophie PINCHINAT Professeur à l'Université Rennes 1
Examinateurs :	Stefan KIEFER, Associate Professor at Oxford
	Antonin KUCERA, Professor at Masaryk University
	Nihal PEKERGIN, Professeur à l'Université Paris-Est Créteil
	Wiesław ZIELONKA Professeur à l'Université Paris 7
Dir. de thèse :	Nathalie BERTRAND, Chargée de recherche INRIA
Co-dir. de thèse :	Serge HADDAD, Professeur de l'ENS Paris-Saclay

En essayant continuellement on finit par réussir. Donc : plus ça rate, plus on a de chance que ça marche.
Les Shadoks

(By trying over and over, one finally succeeds. Thus: the more you fail, the more you have a chance that it will work.)

Contents

Résumé en français

Introduction

De nombreux systèmes critiques doivent satisfaire une spécification donnée. Cette spécification peut comprendre des critères de sécurité, des mesures de rendement ou d'autres conditions. Le processus de vérification, dont le but est de vérifier que le système satisfait une spécification, peut être réalisé de différentes manières. Chacune ayant ses avantages et inconvénients, le choix de la méthode à utiliser dépend à la fois du système qui est étudié et de la spécification. Une des méthodes de vérification possibles est de réaliser des batteries de tests sur le système. Cette méthode est notamment utile lorsqu'on ne connait pas le fonctionnement interne d'un système (le code d'un programme par exemple). Le choix des tests est alors basé sur la spécification [GTWJ03]. Au contraire, si l'on a accès au système complet, un *modèle* formel et opérationnel du système peut être construit. Ce modèle peut ensuite être étudié avec des méthodes dédiées.

La complexité de certains systèmes peut rendre la construction du modèle difficile. Celle-ci peut être accompli grâce à une analyse du code du système, en réalisant des tests spécifiques dans certaines configurations du système permettant de déterminer comment il évolue (par exemple en surchargeant le processeur d'un ordinateur afin d'observer comment le programme réagit face à ce type de pression), etc. Quand elle est possible, cette approche a de nombreux avantages :

- Lors de la conception d'un système, si le prototype actuel n'accompli pas les objectifs voulus, il doit être modifié. Construire de nouveaux prototypes jusqu'à en obtenir un qui soit satisfaisant est coûteux. Il est plus économique et simple, de modifier un modèle jusqu'à ce que celui-ci satisfasse la spécification, et seulement alors de construire le système associé.

- Un modèle est souvent conçu afin de vérifier plusieurs propriétés. Si l'on désire vérifier de nouvelles propriétés à une date ultérieure, utiliser le modèle existant peut suffire. Dans le cas contraire, il n'est pas forcément nécessaire de construire entièrement un nouveau modèle. Il peut être suffisant de raffiner le modèle actuel, en y incorporant les informations appropriées. Utiliser un tel raffinement réduit fortement la complexité de la nouvelle étude.

- Enfin, si le modèle est suffisamment proche de la réalité, il permet une analyse

précise du système. Ce n'est pas le cas lors d'utilisation de batteries de tests, qui ne couvrent qu'une partie des situations possibles. Le même souci existe pour d'autres méthodes telles que la vérification statistique de modèle [Bar14].

Il existe de nombreux formalismes permettant de représenter un système. Plus celui-ci est complexe (représentation du temps, de l'aléatoire...), plus on peut modéliser de systèmes et de spécifications et plus l'étude de ce formalisme est difficile. Notamment, les formalismes possédant une composante probabiliste tels que les chaînes de Markov [KS60] ont de nombreuses applications. En effet, certains systèmes nécessitent des probabilités pour être représentés de façon précise. C'est le cas des systèmes contenant des comportements intrinsèquement probabilistes, par exemple un programme utilisant l'aléatoire afin de briser les symétries. Le hasard peut aussi être une conséquence de l'interaction du système avec l'environnement dont le comportement n'est pas entièrement prévisible. De plus, les probabilités permettent de représenter les incertitudes d'un modèle construit de façon approchée, par analyse statistique par exemple. Enfin, l'utilisation de probabilités élargit l'ensemble des propriétés qui peuvent être spécifiées en permettant de les quantifier. Par exemple, si un système (qui n'est pas un système critique) peut commettre des erreurs, mais que celles-ci ont peu de chances d'avoir lieu, ceci peut suffire pour satisfaire la spécification.

Le choix du formalisme utilisé détermine également quelles sont les informations accessibles aux utilisateurs : lorsque l'on construit un modèle, les différents évènements qui peuvent avoir lieu dans le système et leurs effets sont décrits ; certains de ces évènements ont lieu de façon interne et ne sont donc pas observables par un utilisateur extérieur. Le contrôle de l'information transmise par un système a vu son importance augmenter ces dernières années à cause de l'omniprésence des instruments électroniques communicants. Certaines informations du système doivent être maintenues secrètes (les mots de passe par exemple) alors que d'autres doivent être rendues publiques (les erreurs du système par exemple). Les problèmes liés à l'*observation partielle* peuvent être groupés en trois familles selon le type d'objectif à accomplir : (1) la planification sous observation partielle, (2) la dissimulation d'information à l'observateur et (3) la récupération d'information.

Le *diagnostic* est l'un des problèmes principaux de cette troisième catégorie. Le terme diagnostic vient du domaine médical dans lequel il désigne l'identification d'une maladie à partir de symptômes. Dans la communauté des systèmes à évènements discrets, cette identification est appliquée à des systèmes dynamiques (les centrales électriques, les chaînes de production...). Dans cette approche, une exécution du système est observée et on essaie de détecter si un évènement particulier, appelé la *faute*, a eu lieu. La faute ne représente pas forcément une défaillance du système, cependant cette terminologie est utilisée principalement car une irrégularité est l'un des évènements les plus importants à détecter durant une exécution. En effet, celles-ci menacent la sureté et la disponibilité du système. Ceci peut provoquer des dégâts catastrophiques à la fois en termes économiques et humains. L'étude des fautes est également justifiée du fait que tout système peut, et en fait va, faire une erreur. En effet, les systèmes que l'on construit sont de plus en plus complexes et ont des interactions de plus en plus

importantes avec l'environnement. Il est donc extrêmement difficile de ne pas introduire d'erreurs lors de la conception d'un système et il est presque impossible de prédire toutes les actions que l'environnement aura sur le système. Enfin, des fautes auront lieu à cause du vieillissement des composants du système.

Comme les fautes sont dangereuses, inévitables et potentiellement difficiles à identifier, une méthode automatique de détection est nécessaire. Par ailleurs, cette méthode doit être précise car stopper un système à cause d'un faux positif est coûteux et elle doit être réactive de façon à ce que les fautes soient repérées avant tout dommage sérieux. En réaction à une faute, on peut soit (1) essayer d'optimiser le comportement du système durant la période préalable à la faute [EMT16], ce qui est particulièrement utile pour des systèmes dont les composants sont régulièrement remplacés, ou (2) essayer de détecter la faute de façon à réagir à son occurrence. Comme l'on désire réagir aussi vite que possible, prédire l'occurrence de la faute permettrait de réagir avant même que le système ne commette l'erreur. Cette question est étudiée dans le contexte des problèmes de prédiction [GL09]. Cependant, il est rare qu'un système permette une prédiction efficace des fautes. Détecter les fautes *a posteriori* est plus plausible. L'étude du diagnostic soulève deux problèmes importants : comment décider si un système est diagnostiquable, ce qui est appelé *diagnostiquabilité*, et, dans le cas positif, comment construire un *diagnostiqueur* la fonction réalisant le diagnostic et qui satisferait potentiellement des conditions supplémentaires sur la taille de la mémoire utilisée, le délai de détection, etc. Dans le domaine des systèmes à évènements discrets, le diagnostic a été défini initialement pour des systèmes finis tels que les *systèmes à transitions étiquetées partiellement observables* [SSL$^+$95] puis a été étendu à de nombreux modèles complexes (*e.g.* les réseaux de Petri [CGLS12, BHSS18], les systèmes à pile [MP09], etc.) et cadres (*e.g.* décentralisés [DLT00], distribués [HC94]). De plus, plusieurs travaux rassemblés sous le nom diagnostic actif étudient comment contrôler le système pour en assurer la diagnostiquabilité [SLT98, TT07, CT08, CP09].

Notre but dans ce document est d'étudier le diagnostic de systèmes probabilistes. Par conséquent, la première question qui se doit d'être abordée est le choix du formalisme (probabiliste) à utiliser. On doit notamment déterminer si, en plus des probabilités, le modèle est partiellement contrôlable, s'il peut représenter une infinité d'états différents ou s'il doit décrire efficacement des comportements concurrents. Par ailleurs, comme le diagnostic est un problème d'observation partielle, le modèle doit indiquer quelle observation est associée à une exécution.

Dans un deuxième temps, il va nous falloir établir les différents problèmes que nous allons étudier. De nombreuses notions de diagnostic ont déjà été définies, chacune accomplissant un objectif différent. Nous devons donc présenter un ensemble cohérent de notions qualitatives et quantitatives appropriées englobant les définitions importantes déjà établies. De plus, les définitions formelles des problèmes que nous étudions doivent être choisies prudemment. En effet, comme ces problèmes mélangent observation partielle, probabilités et, dans certains cas, contrôle, ils ont de fortes chances d'être indécidables. Une petite modification dans la définition peut faire la différence entre un problème pouvant se résoudre efficacement et un problème indécidable.

Une fois les notions de diagnostic définies, notre but sera d'établir les complexités

précises de la diagnostiquabilité et de la synthèse des diagnostiqueurs pour chaque notion, ceci, dans les différents formalismes que nous aurons choisis. Nous chercherons également à déterminer comment modifier un système afin qu'il devienne diagnostiquable. Ceci donne deux approches: une approche passive, de vérification, et une active, de contrôle.

Définitions du modèle et du diagnostic

Le modèle le plus utilisé dans ce document est celui des systèmes probabilistes à transitions étiquetées partiellement observables (dont le nom anglais est abrégé en pLTS). Un pLTS est formellement défini par un tuple $\langle Q, q_0, \Sigma, T, \mathbf{P}\rangle$ où Q est un ensemble dénombrable d'états, q_0 est l'état initial, Σ est un ensemble d'évènements pouvant avoir lieu dans le système, T est un ensemble de transitions indiquant comment un évènement de Σ affecte l'état courant du système et $\mathbf{P}$ donne la probabilité de chaque transition. Afin de représenter l'observation partielle du système, l'ensemble d'évènements Σ est partitionné entre évènements observables Σ_o et évènements inobservables Σ_u.

Une exécution ρ du système est une suite d'états et de transitions liant deux états consécutifs. Grâce à $\mathbf{P}$, on peut attribuer une probabilité à toute exécution finie. Ensuite, et en utilisant des résultats de théorie de la mesure, on peut définir une mesure de probabilité sur l'ensemble des exécutions infinies. Par ailleurs, à toute exécution on peut associer une observation qui est la projection sur Σ_o de la séquence d'évènements étiquetant ses transitions. On suppose que le pLTS est convergent ce qui signifie que toute exécution infinie possède une observation infinie.

Example 0.1. *Considérons le pLTS représenté dans la figure 0.1. Une utilisation normale de la machine à café est donnée par exemple par l'exécution* $\rho = q_0$ `pièce` q_1 `sucre` q_1 `café` q_0. *Cependant dans l'état* q_1 *une erreur représentée par l'évènement '***f***' peut avoir lieu, menant à l'état* f_1 *à partir duquel on ne peut plus obtenir de café. Cet évènement a cependant une faible probabilité d'avoir lieu. L'exécution normale* ρ *a pour probabilité le produit des probabilités des transitions empruntées, c'est-à-dire* $1 \times 0.29 \times 0.7 = 0.203$. *Un comportement fautif de la forme* $\rho' = q_0$ `pièce` q_1 `sucre` q_1 **f** f_1 *a probabilité* 0.0029 *d'avoir lieu.*

Comme dans l'exemple précédent, les exécutions du modèle peuvent être fautives ou correctes. Ceci est indiqué par la présence (ou absence) de la faute (l'évènement **f**) à l'intérieur de celle-ci. Le but du diagnostic est d'utiliser l'observation d'une exécution pour déterminer si celle-ci est correcte ou fautive. Comme plusieurs exécutions différentes peuvent posséder la même observation, une exécution est surement fautive (resp. surement correcte) si toute les exécutions partageant la même observation sont fautives (resp. correctes). Sinon, l'exécution est ambiguë. Si une exécution est surement correcte ou surement fautive, un verdict peut être rendu. Le souci vient des exécutions ambiguës.

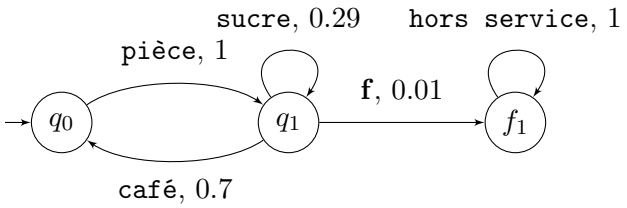


Figure 0.1: Un pLTS représentant une machine à café. q_0 est l'état initial, ce qui est représenté par la flèche entrante. Les transitions entre les états sont étiquetées par l'évènement provoquant cette transition ainsi que par la probabilité que cette transition soit prise.

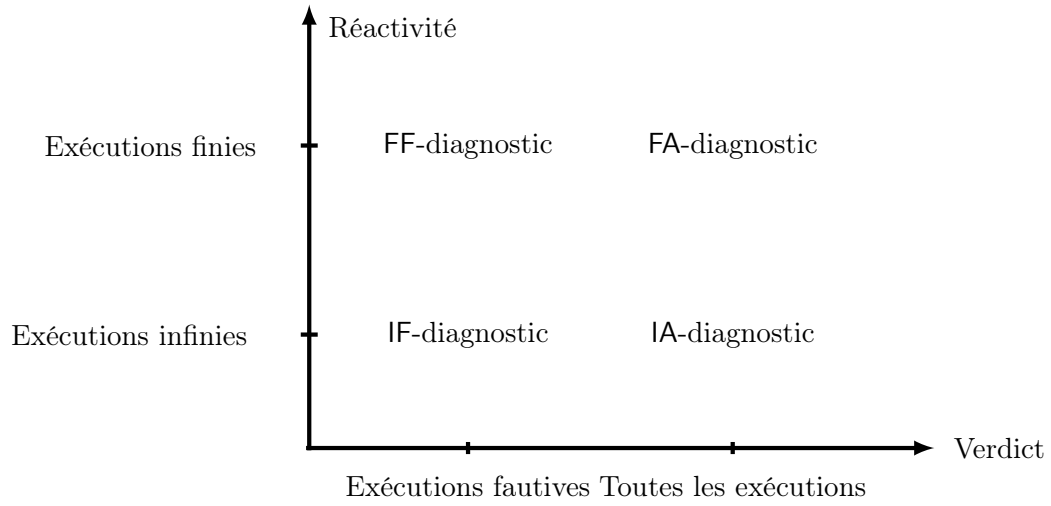


Figure 0.2: Résumé des variantes du diagnostic exact.

Plusieurs définitions de diagnostiquabilité peuvent être proposées. Pour des systèmes non probabilistes, la définition originelle de diagnostiquabilité requiert que toute séquence fautive devienne finalement surement fautive [SSL+95]. Ainsi, toute faute est finalement détectée. Cette condition est trop forte pour des systèmes probabilistes car un système pourrait être déclaré non diagnostiquable à cause d'une exécution de probabilité nulle. Une adaptation possible est de demander qu'avec probabilité 1 une exécution fautive devienne surement fautive [TT05]. Nous appelons cette notion FF-diagnotiquabilité. Cette notion ignore l'ambiguïté des exécutions correctes. Le système peut donc rester ambigü infiniment, ce que l'on peut vouloir éviter. La diagnostiquabilité peut être étendue aux exécutions correctes en requérant que la probabilité des séquences ambigües (fautives et correctes) converge vers 0. Cette notion se nomme FA-diagnostiquabilité (le A signifie "all" alors que F signifie fautif). Pour ces deux notions de diagnostiquabilité, l'ambiguïté est résolue sur des exécutions finies (le premier F signifi-

ant justement fini). Si l'on autorise l'ambiguïté à être résolue lorsque la séquence devient infinie, la FF- et la FA-diagnotiquabilité deviennent la IF- et la IA-diagnotiquabilité. Les quatre notions sont résumées dans la figure 0.2.

Ces notions de diagnostiquabilité ne permettent aucune erreur de verdict. Ce sont des notions dites exactes. Il peut cependant être intéressant d'affaiblir cette condition. Considérons le pLTS de la figure 0.3. Toute exécution fautive est ambiguë. Cependant, de par le choix des probabilités, une exécution fautive a plus de chances de produire un 'b' qu'un 'a' et inversement pour les exécutions correctes. Par conséquent, en comparant le nombre de 'b' et de 'a' dans l'observation, on peut déduire avec forte probabilité si l'exécution est correcte ou fautive. Nous considérons donc plusieurs notions de diagnostiquabilité dites approchées dans ce document (l'une d'entre elle ayant été introduite dans [TT05] et pour laquelle seule une condition suffisante avait été donnée).

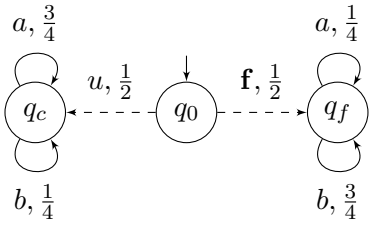


Figure 0.3: Quand un diagnostic approché est nécessaire.

La formalisation du modèle et des notions de diagnostiquabilité est réalisée dans le chapitre 2. Nous expliquons maintenant comment ces notions de diagnostiquabilité peuvent être étudiées.

Vérification de la diagnostiquabilité

L'étude d'un problème commence par une analyse sémantique (développées dans le chapitre 3) de façon à bien le comprendre. Dans le cas de la diagnostiquabilité, cette analyse sémantique prend tout d'abord la forme d'une étude des liens entre les différentes notions de diagnostiquabilité. Certains sont assez clairs. Par exemple, les notions de FA- et IA-diagnostiquabilité considèrent les exécutions fautives ainsi que les exécutions correctes alors que FF- et IF-diagnostiquabilité ne considèrent que les notions fautives. Donc un système FA-diagnostiquable (resp. IA-) est FF-diagnostiquable (resp. IF). Similairement, observer des exécutions infinies donne plus d'informations que leurs préfixes finis, donc les notions de diagnostiquabilité finies impliquent leur équivalent infini. De façon intéressante, si on ne s'intéresse qu'aux exécutions fautives, on a une réciproque partielle : si le système est à branchement fini, la FF-diagnostiquabilité est équivalente à la IF-diagnostiquabilité.

La deuxième étape d'une analyse sémantique est de déterminer des caractérisations efficaces des notions étudiées. Avoir des contraintes sur le système étudié permet d'avoir

des caractérisations plus simple. Nous étudions donc d'abord le cas des systèmes ayant un nombre d'états fini.

Étude des systèmes finis

Les systèmes représenté par un modèle ayant un nombre d'états fini possèdent des propriétés extrêmement utiles pour le diagnostic. Notamment, on sait qu'avec probabilité 1, une exécution atteint une composante strictement connexe terminale (CSCT) du graphe induit par le pLTS. Comme la diagnostiquabilité s'intéresse à des comportements "avec probabilité 1", l'étude peut se concentrer sur les CSCT du système. Par ailleurs, il existe une méthode simple de déterminisation de l'automate induit par le pLTS. Cette déterminisation est très utile pour caractériser l'ambiguïté d'une exécution. En effet, l'automate déterminisé associe à chaque séquence d'observations l'ensemble d'états du pLTS pouvant être atteint par des exécutions associées à cette séquence. Par conséquent, en supposant sans perte de généralité que les états du pLTS sont partitionnés entre états fautifs (atteint par une exécution fautive) et états corrects, une exécution surement fautive est une exécution dont la séquence d'observation mène à un état de l'automate déterminisé ne possédant que des états fautifs. Observons ceci sur un exemple. La figure 0.4 représente un pLTS qui est FF-diagnostiquable ainsi que IA-diagnostiquable mais qui n'est pas FA-diagnostiquable. En effet, toute faute est suivie finalement par un 'b' révélant la faute. L'observation a^ω est donc au contraire associée à une exécution surement correcte, mais tout préfixe fini de cette exécution est ambigu. Donc toute exécution correcte finie est ambiguë.

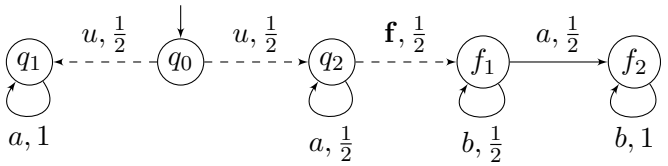


Figure 0.4: Un pLTS qui est IA et FF-diagnostiquable mais n'est pas FA-diagnostiquable.

Le pLTS de la figure 0.4 respecte la partition entre états fautifs (les f_i) et états corrects (les q_i) mentionnée plus tôt. Nous représentons l'automate déterminisé induit en figure 0.5. Un état de cet automate atteint par l'observation w contient deux ensembles : nous séparons les états du pLTS pouvant être atteint par une exécution correcte de ceux atteints par une exécution fautive, de plus on ne considère que les exécutions terminant par un évènement observable. Les états doublement entourés ne contiennent soit aucun état correct, soit aucun état fautif. Les observations menant à ces états correspondent donc à des exécutions non ambiguës.

En observant cet automate et au vu de notre remarque antérieure sur les CFCT, on pourrait penser que le diagnostic de ce système est simple étant donné que le seul état pour lequel le verdict ne peut être rendu ne fait pas parti d'une CFCT de l'automate déterminisé. Cependant, si les CFCT sont atteintes avec probabilité 1 dans le pLTS, ce

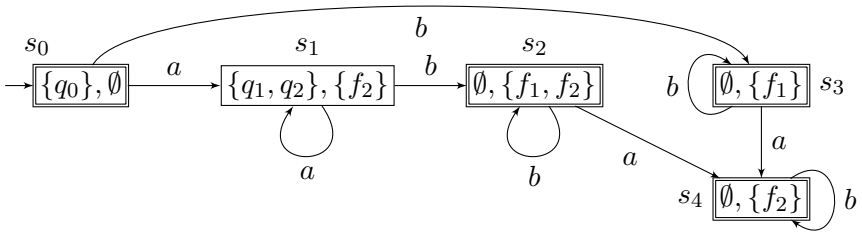


Figure 0.5: L'automate déterminisé associé au pLTS de la figure 0.4.

n'est pas forcément le cas dans l'automate déterminisé. Afin de regagner cette propriété tout en conservant les informations données par l'automate déterminisé, on réalise le produit synchronisé sur les évènements observables du pLTS et de l'automate. Celui-ci est représenté dans la figure 0.6.

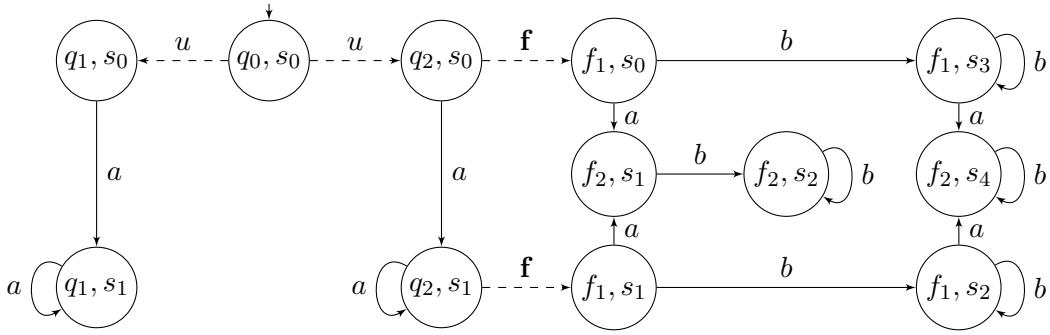


Figure 0.6: Produit synchronisé du pLTS de la figure 0.4 et de son automate déterminisé. Les probabilités sont omises pour faciliter la lisibilité.

Le produit synchronisé conserve le comportement probabiliste du système. C'est-à-dire, qu'il y a une bijection entre les exécutions du pLTS et celles du produit, les deux exécutions ayant la même observation, correction et probabilité. Par conséquent, la diagnostiquabilité du pLTS est équivalente à celle de son produit synchronisé. Par contre, ce dernier possède plus d'informations car la composante des états correspondants à l'automate déterminisé indique si une exécution finissant dans cet état est surement fautive, surement correcte ou ambigüe. Observons les CFCT de ce système. Il y en a 3, toutes réduites à un état: (q_1, s_1), (f_2, s_2) et (f_2, s_4). Les ensembles s_2 et s_4 impliquent que les exécutions atteignant la deuxième et troisième CFCT sont surement fautives. s_1 en revanche, montre une ambiguïté dans la première CFCT. Comme s_1 est associé à un état correct q_1, cette ambiguïté n'existe que pour des exécutions correctes. Ce produit synchronisé montre donc que le pLTS est bien FF-diagnostiquable, mais pas FA-diagnostiquable.

En utilisant le produit synchronisé, une caractérisation peut être établie pour toutes les notions de diagnostiquabilité exacte[1]. Cette caractérisation peut ensuite être vérifiée en espace polynomial. La principale source de complexité de l'algorithme est la déterminisation qui produit un automate de taille au plus exponentielle en la taille du pLTS. Utilisant une réduction du problème de l'universalité du langage généré par un automate non déterministe, on montre que les notions de diagnostiquabilité exacte sont PSPACE- difficiles. Elles sont donc PSPACE-complètes pour les pLTS finis.

En ce qui concerne le diagnostic approché, seule une notion, construite en modifiant subtilement celle introduite dans [TT05], est décidable. La décidabilité est montrée en réduisant le problème à un nombre au plus quadratique d'instances du problème de la distance 1 de deux chaînes de Markov étiquetées, problème qui a été montré décidable en PTIME dans [CK14]. Ceci donne *in fine* un algorithme PTIME. Les résultats d'indécidabilité quant à eux sont montrés grâce à des réductions du problème du vide des automates probabilistes [Paz71].

Construction des diagnostiqueurs

Le but de l'étude du diagnostic est la détection automatique de la faute. Cette détection est réalisée par un diagnostiqueur qui observe le système et donne son verdict. Formellement, un diagnostiqueur est une fonction $D : \Sigma_o^* \to \{?, \top, \bot\}$. Un verdict ? ne fournit aucune information, un verdict $\top$ déclare l'exécution actuelle fautive et un verdict $\bot$ fournit une information relative à la correction de l'exécution. Un diagnostiqueur a trois caractéristiques principales: *verdict*, *sureté* et *réactivité*. Le verdict formule la nature de l'information que le diagnostiqueur doit fournir au cours d'une exécution (détection de fautes uniquement, ou aussi de la correction de l'exécution par exemple). La sureté formule quand le diagnostiqueur peut émettre son verdict. Dans le cas du diagnostic exact, la sureté requiert que si le diagnostiqueur produit un verdict, celui-ci est correct. Ce n'est pas forcément le cas dans le cadre du diagnostic approché. La réactivité exprime à quelle régularité le diagnostiqueur doit fournir des informations sur le statut de l'exécution courante.

Les notions de diagnostiquabilité ayant été présentées sous la forme d'un problème de décision, la troisième étape de l'étude sémantique consiste à établir le lien entre chaque notion de diagnostiquabilité et l'existence d'un diagnostiqueur avec un verdict, une sureté et une réactivé donnés. La preuve permettant d'établir le lien entre diagnostiquabilité et existence d'un diagnostiqueur est constructive. Par conséquent, nous disposons d'un algorithme permettant de construire automatiquement un diagnostiqueur pour chaque système diagnostiquable. Cependant, le diagnostiqueur en question utilise une mémoire non bornée. En vue d'une possible implémentation, il est préférable de pouvoir se limiter à des diagnostiqueurs à mémoire finie. Un tel diagnostiqueur est représenté par un automate déterministe sur Σ_o enrichi d'un verdict $(M, \Sigma_o, m_0, \mathsf{up}, D_{fm})$ où M est un ensemble d'états de la mémoire avec m_0 l'état initial, up est la fonction de tran-

[1]Pour la IA-diagnostiquabilité, une information supplémentaire est nécessaire. L'automate déterminisé que l'on construit possède un troisième ensemble permettant de partitionner les états fautifs en deux groupes, selon le moment où la faute a été commise.

sition, mettant à jour la mémoire du diagnostiqueur et $D_{fm} : M \to \{?, \top, \bot\}$ associe un verdict à chaque état de la mémoire. La taille d'un tel diagnostiqueur à mémoire finie est le nombre d'états de la mémoire qu'il possède. La fonction de mise à jour peut être étendue à des séquences d'observations de façon inductive : pour ε le mot vide $w \in \Sigma_o^*$ et $a \in \Sigma_o$, $\mathsf{up}(m, \varepsilon) = m$ et $\mathsf{up}(m, wa) = \mathsf{up}(\mathsf{up}(m, w), a)$. Si un diagnostiqueur à mémoire finie n'est pas un diagnostiqueur selon la définition établie plus haut, il en induit un qui est définit par $D(w) = D_{fm}(\mathsf{up}(m_0, w))$ pour tout $w \in \Sigma_o^*$.

Example 0.2. *La figure 0.7 représente un diagnostiqueur à mémoire finie qui ne fournit aucune information (verdict ?) initialement puis déclare une faute (verdict $\top$) dès qu'un 'b' est observé.*

Considérons le pLTS de la figure 0.4. La faute est identifiée dès qu'un 'b' est observé. Par conséquent, le diagnostiqueur induit par le diagnostiqueur à mémoire finie de la figure 0.7 peut être utilisé pour détecter les fautes de ce pLTS. Ce diagnostiqueur ne détecte pas les exécutions correctes, il correspond à la notion de FF*-diagnostiquabilité. On l'appelle donc un* FF*-diagnostiqueur.*

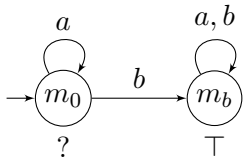


Figure 0.7: Exemple de diagnostiqueur à mémoire finie. Le verdict donné par D_{fm} dans un état de la mémoire est indiqué sous l'état.

Pour toute notion de diagnostic exact, un diagnostiqueur peut être construit à partir de l'automate déterminisé qui a été utilisé pour vérifier la diagnostiquabilité. Le diagnostiqueur peut donc avoir une taille exponentielle. C'est malheureusement inévitable : certains pLTS n'admettent pas de diagnostiqueurs de taille sous-exponentielle. Pour le diagnostic approché, aucune borne sur la taille de la mémoire n'existe dans le cas général. En d'autres mots, on ne peut pas toujours construire de diagnostiqueur à mémoire finie. Pire encore, déterminer s'il existe un diagnostiqueur à mémoire finie est un problème indécidable.

Ces résultats, reposant fortement sur la limitation à un nombre d'états finis pour le modèle, sont rassemblés dans le chapitre 4. Nous allons maintenant discuter de ce qui peut être fait pour lever cette restriction.

Étude des systèmes infinis

De nombreux systèmes réels nécessitent un nombre infini d'états pour être décrit de façon précise. Afin de rendre une analyse possible, on ne peut pas utiliser directement un pLTS infini. On a besoin d'un modèle de plus haut niveau, capable de représenter de façon finie un pLTS infini. De nombreux formalismes permettent ceci. Les automates à

pile et les réseaux de Petri notamment représentent deux classes orthogonales de LTS infinis. Le choix du formalisme est très important car plus un formalisme est expressif, plus les problèmes seront compliqués à résoudre. Les automates à pile et les réseaux de Petri sont trop puissants par exemple, toutes les notions de diagnostiquabilité exacte étant indécidables dans ces deux formalismes.

Nous nous sommes donc intéressés à un formalisme légèrement plus faible, plus précisément une sous-classe des automates à pile : les automates à pile visibles. Les automates à pile sont des automates enrichis d'une pile qui leur permet de conserver de l'information au cours d'une exécution. Les transitions de l'automate disponibles dépendent de l'état courant ainsi que de la tête de la pile. Une fois sélectionnée, une transition peut soit (1) modifier la tête de la pile (transition locale), (2) ajouter un nouveau symbole en tête de pile (transition d'empilement) ou (3) retirer la tête de pile actuelle, s'il y en a une (transition de dépilage). La sémantique d'un automate à pile probabiliste est un pLTS infini dont les états représentent l'état actuel de l'automate à pile ainsi que le contenu de la pile. Ce pLTS peut être infini car la taille de la pile n'est pas bornée. La restriction aux automates à pile visible requiert que l'ensemble des évènements est partitionné selon le type de transitions auquel ils correspondent, $\Sigma = \Sigma_\sharp \cup \Sigma_\flat \cup \Sigma_\natural$. Les évènements de $\Sigma_\sharp, \Sigma_\flat$ et $\Sigma_\natural$ sont respectivement associés à des transitions d'empilement, de dépilage et des transitions locales. De plus, $\Sigma_\sharp \cup \Sigma_\flat \subseteq \Sigma_o$. Un observateur voit donc quand un élément est ajouté ou retiré de la pile. La taille de la pile est donc connue à tout moment, son contenu par contre peut être inconnu.

Example 0.3. *La figure 0.8 donne un exemple d'automate à pile probabiliste. Une exécution démarre dans l'état q_0 avec pour contenu de pile le symbole $\perp_0$. Celui-ci est appelé élément de fond de pile et ne peut pas être retiré ou modifié. Le reste de la pile n'est composé que d'un certain nombre de γ. La seule transition d'empilement est* in *et les transitions de dépilage sont* out *et* abort.

Ce système reçoit un certain nombre d'ordres qu'il note dans la pile. Puis il commence à servir ses clients, chaque ordre reçu reçoit donc une réponse soit sous la forme d'un out*, soit de façon fautive sous la forme d'un* abort*. Enfin, il retourne à son point initial. Les deux exécutions données en exemple présentent un comportement correct et fautif pour deux ordres reçus.*

Il y a une claire partition entre transitions d'empilement, de dépilage et locales. Par conséquent si in*,* out *et* abort *sont observables, cet automate à pile probabiliste est visible.*

La restriction aux automates à pile probabilistes visibles limite peu l'expressivité du formalisme, mais donne des propriétés supplémentaires utiles au modèle, notamment elle permet de réaliser une déterminisation de l'automate à pile. On ne peut cependant pas faire comme dans les systèmes finis et étudier les CFCT du produit du modèle et de son automate déterminisé. Notamment car ces CFCT peuvent ne pas exister (cas où aucune borne sur la taille de la pile n'existe). Une autre forme de caractérisation de la diagnostiquabilité est donc nécessaire. Pour ce faire, une logique nommée pathL a été introduite et des formules caractérisant plusieurs notions de diagnostiquabilité exacte ont été établi. Ensuite, en utilisant notamment l'automate à pile déterminisé, ces formules ont pu être traduites en des formules de pLTL, une logique connue et pour laquelle des

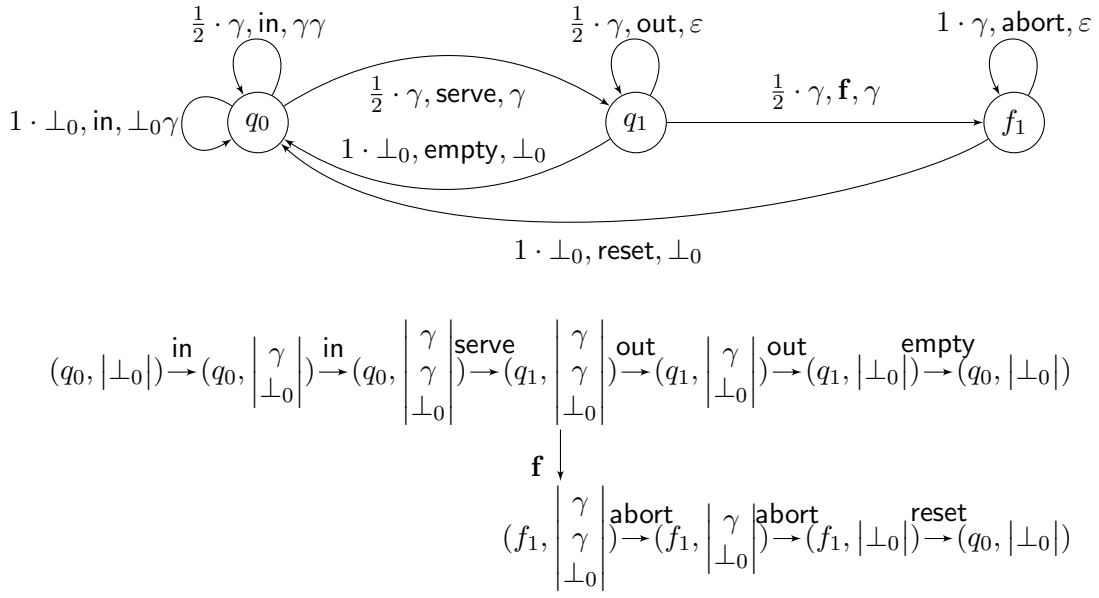


Figure 0.8: Un automate à pile probabiliste et deux de ses exécutions finies.

algorithmes de vérification existe pour les automates à pile. En utilisant ces résultats nous avons obtenu des algorithmes EXPSPACE pour les notions de diagnostiquabilité caractérisées (la borne inférieure prouvée étant EXPTIME). La FA-diagnostiquabilité, qui requiert la détection des séquences fautives et correctes en temps fini n'a cependant pas pu être caractérisée. En fait, des résultats de non-expressivité ont été établi pour montrer que cette notion ne pouvait pas être exprimée en pathL.

Ces résultats sont développés dans le chapitre 5.

Contrôle d'un système

Les pLTS donnent une représentation passive d'un système. Ainsi, étudier le diagnostic sur des pLTS est purement un travail de vérification. Il ne permet donc pas de questionner efficacement comment modifier le système afin de le rendre diagnostiquable. Afin d'étudier ce genre de problème, on ajoute une forme de contrôle dans le pLTS. Le formalisme obtenu, appelé CLTS partitionne l'ensemble des évènements observables en évènements contrôlables Σ_c et évènements incontrôlables Σ_e. Après chaque observation, un contrôleur choisit un ensemble d'évènements autorisés excluant potentiellement certains évènements contrôlables. Ceci limite donc les transitions pouvant être prise par le système.

Example 0.4. *Un exemple de CLTS est représenté dans la figure 0.9. Initialement, deux évènements de poids* 1 *sont possibles et forcément autorisés par le contrôle. Par*

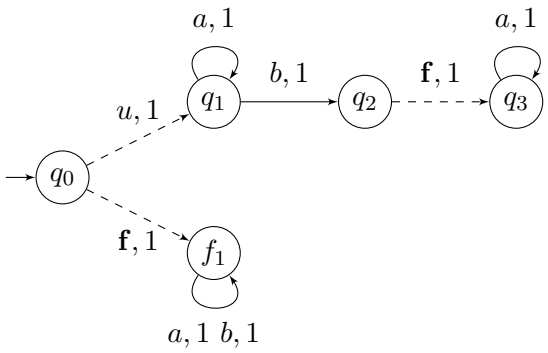


Figure 0.9: Un exemple de CLTS. Les probabilités sont remplacées par des poids. Le seul évènement contrôlable est 'b'.

conséquent chacune a une probabilité $\frac{1}{2}$ *d'être prise. En* q_1*, 'b' peut être interdit par le contrôleur. Si c'est le cas, la transition étiquetée par 'a' est choisie avec probabilité 1.*

Ainsi, si le contrôleur autorise tous les évènements en permanence, l'exécution $q_0uq_1aq_1bq_2$ *a probabilité* $1/8$*. S'il interdit 'b' en permanence, cette exécution a probabilité* 0*. Finalement, s'il n'autorise un 'b' qu'après l'observation d'un 'a', l'exécution a probabilité* $1/4$*.*

Le contrôle est formellement défini par l'utilisation de stratégies. Une stratégie $\pi : \Sigma_o^* \mapsto \mathsf{Dist}(2^\Sigma)$ est une fonction associant à une séquence d'observations une distribution probabiliste sur les ensembles d'évènements autorisés. Par ailleurs, si l'ensemble $\Sigma^\bullet$ est sélectionné par la stratégie, on a $\Sigma_u \cup \Sigma_e \subseteq \Sigma^\bullet$. En d'autres mots, la stratégie ne peut exclure que des évènements contrôlables. Un CLTS $\mathcal{C}$ équipé d'une stratégie π génère un pLTS infini dénoté $\mathcal{C}_\pi$.

Example 0.5. *Considérons le CLTS* $\mathcal{C}$ *représenté dans la figure 0.9. Il y a deux ensembles d'évènements possibles à autoriser:* Σ *et* $\Sigma \setminus \{b\}$ *que nous abrégeons en* Σ^-*. Définissons la stratégie* π *par* $\pi(a^n) = p_n \cdot \Sigma^- + r_n \cdot \Sigma$ *avec* $p_n + r_n = 1$ *pour tout* $n \in \mathbb{N}$ *et* $\pi(w) = 1 \cdot \Sigma$ *sinon. C'est-à-dire, après avoir observé* a^n*, avec probabilité* p_n *l'ensemble* Σ^- *est autorisé par la stratégie et avec la probabilité complémentaire* Σ *est autorisé. Le pLTS généré* $\mathcal{C}_\pi$ *est infini. Une partie de celui-ci est représenté en figure 0.10.*

Expliquons la distribution de probabilité à la sortie de la configuration $(\varepsilon, q_1, \Sigma)$*. Les deux transitions sortant de* q_1 *ont le même poids, comme elles sont toutes deux autorisées par la stratégie, la probabilité de chaque transition est* $\frac{1}{2}$*. Comme 'a' et 'b' sont observables, un nouveau contrôle est choisi. Si un 'b' est observé, de par la définition de* π*, le nouveau contrôle est* Σ*. Par contre, si un 'a' est observé, le nouveau contrôle est* Σ^- *avec probabilité* p_1 *et* Σ *sinon. Il y a donc trois transitions sortantes de* $(\varepsilon, q_1, \Sigma)$ *ayant pour probabilité respectivement* 0.5*,* $0.5p_1$ *et* $0.5r_1$*.*

Nous discutons maintenant des questions abordées pour les CLTS.

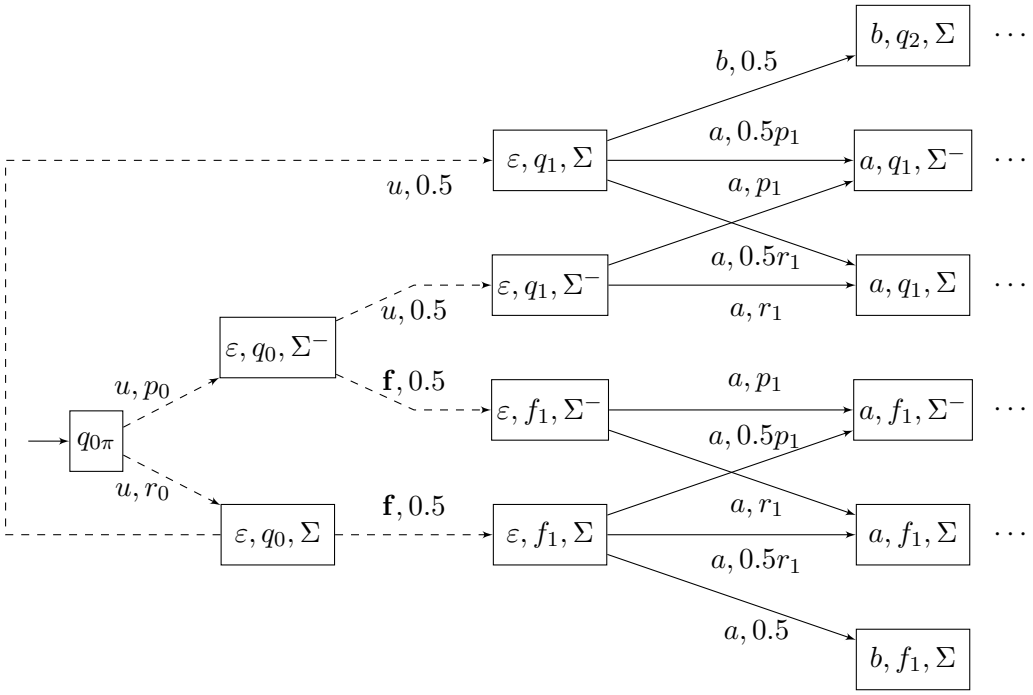


Figure 0.10: Un pLTS obtenu en contrôlant le CLTS de la figure 0.9.

Diagnostic actif et dégradation

Lorsque l'on étudie le diagnostic d'un CLTS, la question n'est plus si le CLTS est diagnostiquable (ce qui n'a techniquement pas de sens en soi) mais s'il existe une stratégie telle que le pLTS induit est diagnostiquable. Cette problématique a été étudiée pour des systèmes probabilistes dans [BFH$^+$14]. Afin de déterminer s'il existe une stratégie satisfaisant la notion de diagnostiquabilité qu'ils étudient, ils traduisent le problème du diagnostic en une condition de Büchi pour un processus de décision Markovien partiellement observable. Le problème peut ensuite être résolu avec des techniques connues. La stratégie obtenue a de bonnes propriétés : elle est notamment ce qu'on appelle "basée sur la croyance", ce qui implique entre autres que le pLTS généré est fini.

Leur travail soulève un souci important : afin de rendre le système diagnostiquable, la stratégie peut faire des choix problématiques comme forcer l'occurrence d'une faute. Bien que ceci permette de détecter le mauvais comportements du système, cela va à l'encontre du but initial du diagnostic qui est de pouvoir utiliser un système fonctionnel. Ils introduisent donc le problème du diagnostic sûr. Celui-ci demande s'il existe une stratégie satisfaisant à la fois le diagnostic et assurant une probabilité positive aux exécutions correctes. Cette notion est malheureusement indécidable dans le cas général et un algorithme NEXPTIME est donné dans le cadre limité des stratégies à mémoire finie (*i.e.* pour lesquelles le pLTS engendré est fini).

Continuant sur cette idée, nous avons introduit de nouvelles notions permettant

de mesurer la dégradation du système. Ces notions ne s'assurent pas que le système reste correct infiniment avec une probabilité positive comme le diagnostic sûr, mais demandent que, si faute il y a, celle-ci puisse être retardée fortement. Le diagnostic longtemps correct et le diagnostic fortement résistant sont deux telles notions, mesurant différemment le délai à imposer à l'occurrence de la faute. Ces deux notions sont impliquées par le diagnostic sûr et sont différentes quand appliquées à des pLTS infinies (*i.e.* il existe des pLTS infinis satisfaisant chacune des notions sans satisfaire l'autre).

Parmi les notions de dégradation que nous avons introduites, certaines sont quantitatives, d'autres, comme le diagnostic longtemps correct et le diagnostic fortement résistant, sont qualitatives. Les notions quantitatives sont toutes indécidables, même limitées à des stratégies à mémoire finie. Au contraire, des algorithmes ont pu être établis pour les notions qualitatives. Ceux-ci procèdent en deux étapes. Tout d'abord et en enrichissant les états du CLTS comme fait dans le cas passif grâce à une déterminisation du CLTS, on identifie l'ensemble des états du CLTS enrichi que l'on peut visiter tout en respectant la diagnostiquabilité. Cette méthode permet en fait de construire la stratégie la plus permissive assurant la diagnostiquabilité du système. Dans un second temps, on étudie le CLTS réduit aux états accessibles sous cette stratégie et on identifie comment restreindre la stratégie afin de rester suffisamment longtemps dans une exécution correcte. Cette étude peut se réaliser en EXPTIME. Le diagnostic actif étant EXPTIME-difficile, les notions qualitative de dégradation introduites sont donc EXPTIME-complètes dans le cas général. Nous avons également montré comment réduire à EXPTIME la complexité du diagnostic sûr limité aux stratégies à mémoire finie. Sous cette restriction, le diagnostic sûr est donc EXPTIME-complet également.

Ces travaux sont présentés dans le chapitre 6.

Assurer l'opacité d'un système

Dans cette thèse, le diagnostic est le problème d'observation partielle auquel nous avons prêté le plus d'attention. D'autres problèmes d'observation partielle sont également intéressants à étudier, notamment l'opacité que nous étudions formellement dans le chapitre 7. Le but de l'opacité est de cacher une information à l'observateur. En conséquent, sur bien des aspects cette notion apparait comme un dual du diagnostic.

Formellement, un système possède deux types d'exécutions : publiques ou secrètes. Pour déterminer, si une exécution est secrète, on pourrait faire comme pour le diagnostic et utiliser un évènement particulier qui, s'il est présent dans une exécution, la rend secrète. De façon équivalente, ceci peut être représenté en partitionnant l'ensemble des états du système en états publics et états secrets et en considérant ces derniers absorbants. Une exécution est secrète ici si elle visite un état secret. C'est cette seconde option que nous utilisons pour l'opacité. Parmi les exécutions secrètes, certaines révèlent le secret. Ce sont celles pour lesquelles toute exécution ayant la même observation est secrète. Notons le parallèle avec le diagnostic : en considérant les exécutions secrètes comme fautives, une telle exécution serait appelée surement fautive. Lorsqu'on étudie l'opacité d'un système, nous désirons mesurer à quel point le secret est révélé, c'est-à-dire quelle est la mesure de probabilité des exécutions révélant le secret. Cette mesure

est appelée la révélation.

L'opacité a été étudiée dans un contexte passif, ainsi que dans un contexte actif. Le contrôle utilisé dans l'étude de l'opacité est fortement différent de celui utilisé pour le diagnostic. En effet, le contrôleur est une fonction associant à une exécution (et non à une séquence d'observations) une distribution sur un ensemble d'action. Chaque action correspond à une distribution de probabilité sur un ensemble de transitions. Ce formalisme donne beaucoup plus de puissance au contrôleur. Il connait précisément quelle est l'exécution actuelle, et son choix n'est pas forcément limité à quels évènements contrôlables il autorise, mais à un ensemble d'actions décrivant potentiellement des choix plus complexes.

Autre différence avec les CLTS, dans le cadre de l'opacité, les observations ne sont plus mises sur les transitions, mais sur les états. Pour noter ces différences, on parlera d'OMC pour les systèmes passifs et d'OMDP pour les systèmes contrôlables.

Example 0.6. *Considérons l'OMC représenté en figure 0.11. L'observation associée avec chaque état est indiquée à côté de celui-ci. Les états secrets sont indiqués en grisé. En supposant que o_1 et o_2 sont deux observations autre que ε, toute exécution contenant au moins 3 observations révèle le secret. La révélation est donc de* 1.

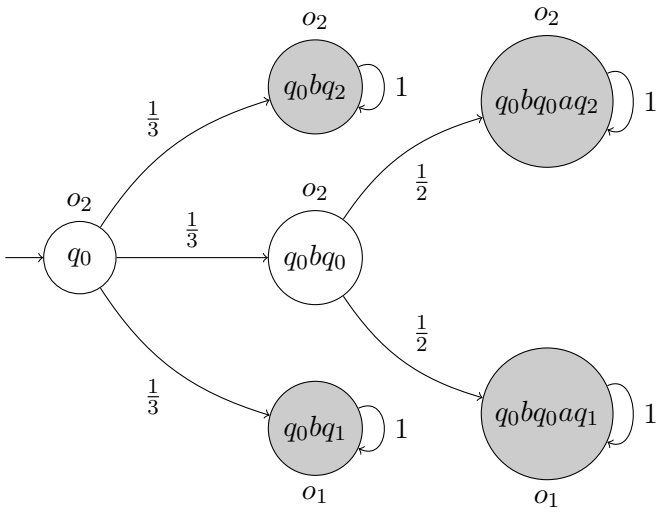


Figure 0.11: Un exemple d'OMC.

Considérons maintenant l'OMDP de la figure 0.12. Dans l'état initial q_0, deux actions sont possibles. Si l'action 'a' est choisie, l'exécution entre en q_1 avec probabilité $\frac{1}{2}$ et en q_2 avec la même probabilité. Si 'b' est choisie, tous les états ont une probabilité $\frac{1}{3}$ d'être atteint.

Définissons la stratégie π choisissant initialement l'action 'b', puis toujours l'action 'a'. L'OMC induit par ce OMDP contrôlé par la stratégie π est celui représenté dans la figure 0.11. Ainsi, en utilisant la stratégie π, on assure une révélation de 1. *En utilisant une stratégie π' qui sélectionne 'b' à tout les coups, la révélation n'aurait été que de $\frac{1}{2}$.*

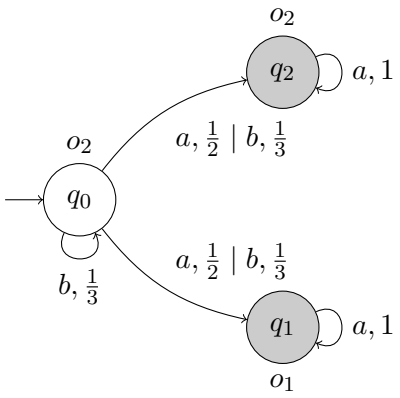


Figure 0.12: Un example d'OMDP. Les transitions sont étiquetées par des paires d'action et de la probabilité de prendre cette transition si cette action est choisie.

Intuitivement, lorsqu'on contrôle un système pour augmenter la révélation, on réalise une analyse du pire cas pour le système. Dans la réalité, ce pire cas est atteint si, par exemple, le contrôle est effectué par un virus ayant accaparé certaines fonctionnalités du système. L'inverse, quand le contrôle cherche à minimiser la révélation, est aussi intéressant à étudier. Cela représente, par exemple, le cas où un concepteur possède quelques degrés de liberté dans son système et désire choisir l'option qui maximisera l'opacité du système. Ces deux problèmes semblent symétriques au premier abord, mais leur analyse est en fait extrêmement différente.

Pour la maximisation de la révélation, une simplification est possible au niveau des stratégies : les stratégies déterministes sont suffisantes. C'est-à-dire, afin de maximiser la révélation, on peut se contenter de considérer des stratégies qui associent à chaque exécution finie non pas une distribution sur les actions, mais directement une action. Malgré cette simplification cependant, presque tous les problèmes sont indécidables. La seule question importante que l'on peut résoudre est : étant donné une OMDP M, existe t-il une stratégie π telle que l'OMC induite M_π a une révélation de 1. Ce problème peut se traduire en un problème d'accessibilité avec probabilité 1 dans un processus de décision Markovien partiellement observable, problème pour lequel des algorithmes efficaces existent. Le processus construit est de taille exponentielle et l'algorithme résolvant le problème d'accessibilité est en EXPTIME. Par conséquent une application directe donne un algorithme 2EXPTIME. Cependant, une analyse précise montre qu'une seule exponentielle est nécessaire. En effet, la complexité de l'algorithme vient principalement de l'utilisation d'une forme de déterminisation du processus de décision Markovien partiellement observable. Hors, celle-ci est déjà nécessaire à la transformation de l'OMDP vers le processus et n'a donc pas à être répétée. Le problème est également EXPTIME-difficile (réduction depuis les jeux de sécurité à information partielle), il est donc EXPTIME-complet.

Pour la minimisation, la situation est différente, d'une façon surprenante. Tout

d'abord, on ne peut pas se limiter à des stratégies déterministes. La capacité du contrôleur à agir de façon randomisée est importante pour rendre le système opaque. Pour autant, les problèmes sont beaucoup plus facile à résoudre : la révélation exacte de l'OMDP peut être calculée. En effet, bien que l'on ne puisse pas utiliser de stratégies déterministes, on peut se limiter à un type de stratégies particulier, que nous avons nommé quasi-déterministe. Ces stratégies choisissent une action et lui associent une probabilité proche de 1, puis partagent le reste de la probabilité sur toutes les autres actions. N'utiliser que des stratégies de cette forme permet de réduire le problème de la minimisation de la révélation à un problème d'accessibilité dans les processus de décision Markovien. Contrairement au cas de la maximisation, ceux-ci sont totalement observables et plus de problèmes sont décidables dans ce cas. Notamment, minimiser la probabilité d'accessibilité lorsque l'observation est complète peut être réalisé en temps polynomial. Donc comme le processus que l'on construit est de taille exponentielle, on obtient un algorithme EXPTIME. Le problème n'est cependant pas prouvé EXPTIME-difficile. La meilleure borne inférieure dont l'on dispose est PSPACE et est obtenue par réduction de la validité d'une formule booléenne quantifiée (QBF).

Conclusion

Cette thèse présente principalement une analyse des problèmes liés au diagnostic de systèmes probabilistes. Sa première contribution est de rassembler en un tout cohérent les différentes définitions existantes sur ce problème. Ceci permet à la fois de donner une base solide à la recherche présentée ici, et de servir de fondations à toute recherche future sur ce sujet.

En deuxième point, cette thèse explique comment vérifier les notions de diagnostiquabilité définies pour différents systèmes, que ceux-ci soient constitués d'un nombre d'états fini ou infini. Pour les systèmes infinis, la décidabilité de certaines notions reste ouverte et certains algorithmes ne sont pas prouvés optimaux. Il reste donc du travail à réaliser dans cette direction.

Le cas des systèmes contrôlables a enfin été étudié. Pour ceux-ci, un autre angle de questionnement a été utilisé : il ne s'agit plus seulement de déterminer la diagnostiquabilité du système. En premier lieu, il s'est agi de combiner le diagnostic avec une limitation de la dégradation du système. Combiner ces deux problèmes ne fait sens que pour des systèmes actifs : pour un système passif, les deux problèmes peuvent être vérifiés séparément, au contraire, dans des systèmes actifs, il peut exister pour chaque propriété une stratégie la vérifiant, mais aucune stratégie ne satisfait les deux simultanément. Dans un second temps, il a été question de l'analyse de l'opacité, une autre notion de contrôle de l'information produite par un système.

Part I

Introduction

Chapter 1

General Introduction

Model-based verification. Many critical systems must fulfil a given specification. This specification may include security criteria, efficiency measures or other kinds of requirements. The verification process, which checks if the system respects the specifications, can be performed in several manners. Each one having its pros and cons, the choice of the best method to use strongly depends on the kind of systems to be verified and on the specification. One of the possible method of verification is to perform tests on the system. If one wants to test a program for example, yet does not have access to the internal structures or working of it, the tests can be realised through specification-based testing [GTWJ03]. In this method, one considers the program as a black-box and focuses on the specification in order to determine which inputs are the most likely to show a failure of the system. If one has access to the content of the program, one can build a formal and operational *model* of the system. Then this model can be analysed via dedicated methods.

Building the model may be difficult for some complex systems. It can be done by analysing the code of the system, by making specific tests in its different states to understand its evolution (for example by overloading the CPU to see how a program reacts faced to this kind of stress), etc. When possible, this approach has many benefits:

- When designing a system, if the current prototype does not satisfy our goals, it must be modified. Building iteratively a new prototype until one gets a good result is expensive. It is easier and cheaper to modify a model of the system until it satisfies the requirements and only then implement the system.

- A model is often built for checking several properties. If one wishes to check for additional properties at a later date, verifying the existing model may be sufficient. If not, one does not necessarily have to build a fully new model. One only needs to refine the existing one with the appropriate, missing information, which strongly reduces the complexity.

- Finally, if the model is close enough to reality, then it allows for an accurate analysis of the system. This is not the case when using arrays of tests which only

partially cover the range of possibilities. The same issue exists for other methods such as statistical model checking [Bar14].

Probabilistic models. There exist several different formalisms for representing a system. The more complete the formalism is (by adding time, multiple players...), the more systems and specifications can be described in it, but also the more complex it is to study.

In particular, stochastic models such as Markov chains [KS60] or Markov decision processes [Put94] have many applications. There are some systems that require probabilities in order to be accurately represented. For example, they can be used to represent systems that contains inherent random behaviours. This occurs in any program using randomisation in order to break symmetries for instance, such as the algorithms dealing with the consensus problem [Agu10]. The randomisation also appears in the processes used in the consensus problem as one might represent the possibility that these processes fail with some probability. Another example of application of stochastic models is the case of systems that face unpredictable behaviours from the environment. This can be the case for a server that receives requests. These requests have randomised content and their timing of arrivals can also be random. The latter requires to mix probabilities and time in the model, as in stochastic timed automaton [BBB$^+$14]. Moreover, probabilities can also be used in the model to represent the uncertainty created when the modelling is done through a statistical analysis.

Using probabilities also enlarges the set of properties that can be specified by giving a measure on the runs of the system. For example, if a non-critical system possesses failures, yet one can determine that they are not likely to occur, this may be enough. Let us also consider a security example: if an attacker tries a password and discovers that it is wrong, he technically gets an information, however this information is not important enough to be worrisome. With probabilities, the specification can quantify the properties of the system the designer wants to verify. Moreover, even a qualitative quantification is useful as it allows to neglect behaviours that are present in the model, yet have a zero probability of occurring.

Paradigms of partial observation. Another important component of a system that can be modelled is related to the observation available to the users: when one builds a model, one describes the different actions that can be taken by the system, however, these actions may be internal and are not necessarily visible by an external observer. Managing the information exchanged with a system has shown increasing importance in recent years due to the omnipresence of communicating electronic devices. Some of the actions of the system may need to be kept private (passwords) while others must be made public (failures). The problems raised by partial observation can be grouped in three families thanks to the different types of goals the system has: (1) planning under partial observation, (2) hiding information from the observer and (3) getting information from the system.

This first category appears for example when studying games. In a game of poker, a player has to select a decision based on some observations (his cards) and on some

partial information (his opponents choices), moreover probabilities are involved in order to determine the likelihood to draw a specific card. Such a case falls under the study of works such as [BGG09] where the authors look for almost-surely winning or positively winning strategies in stochastic games. Partially observable Markov decision process (POMDP) is the stochastic one-player (also called one and a half player) special case of the above (See [CDH10] for algorithms to achieve qualitative objectives in POMDP). POMDP have been extensively used in the IA community, for example in order to plan the actions of a moving robot [KLC98].

Many works focus on hiding information from an attacker. For instance code obfuscation [BGI+01], albeit in a complete observation setting. In a partial observation setting, many theoretical hiding problems are gathered under the general name "*opacity*". An opaque system hides an information by ensuring the observation given by a secret behaviour of the system will be identical to the observation triggered by a non-secret behaviour. A practical example is given in [ABCP13] where the authors investigate how to hide the position of a cellphone user (by randomising the position declared by the phone) while getting relevant answers to location-based requests. In a theoretical and stochastic setting, [BKM12, BMS15] defined probabilistic measures of the opacity of a model. The studied model is *passive*: once defined, one cannot modify its behaviour in order to ensure better properties. A form of control is quickly introduced in [BMS15] and expanded in [BCS15, BKMS16, BKMS18]. More precisely, the authors of [BCS15] investigate Markov decision processes (MDP) with or without partial observation and secrets given by the infinite language of an automaton (using various accepting conditions). In [BKMS16, BKMS18], a model in between MDP and POMDP is used: the control is realised as in an MDP (thus the controller uses complete information) but the winning condition (opacity) uses partial observation and is thus more related to POMDP problems. One issue with these approaches is that they all rely on the black-box hypothesis: it is assumed that the opponent does not know how the non-determinism of the system is resolved. This hypothesis simplifies the problem, but is unrealistic in many cases. For example, the control within the system could be the result of a virus implanted by the attacker. In this case, it is natural for the attacker to know how the virus is implemented.

On the opposite, if the goal is to get information from the system, the first question to answer is to determine the kind of information that must be detected. One possibility is to determine, given a set of partially observable systems, which system is producing the current observation. In [CK14], the authors investigate how far two labelled Markov chains are one to the other in terms of the probabilities of the observed behaviours. They use a distance to measure the importance of the difference between the two models and approximate (or in some cases compute) this distance. The identification of a system can be applied to other questions such as the identification of its initial state, the equivalence of Markov chains (when the distance is equal to 0, see [DHR08]) or the monitoring of hidden Markov chains (HMC). In this last example, one monitor observes a random run of one of two given HMC and must determine which HMC is the origin with appropriately high probability. The case of the monitor required to be correct with probability 1 on infinite sequences was introduced in [SZF11] and solved using the

distance of [CK14] in [KS16]. Instead of identifying the current system, one can wish to obtain a specific information from the observations of the current system. We develop this kind of problems in the next part due to their importance within this thesis.

Diagnosis. *Diagnosis*, from the greek "*δĭάγνωσĭς*" which means "to distinguish" or "to discern", is by the definition of the wiktionary the "identification of the nature and cause of something (of any nature)". This describes many different problems in various domains. In medicine, a doctor analyses the symptoms to deduce the illness causing them. There has been multiple works in order to automatise this kind of diagnosis such as the rule-based expert system MYCIN [BS84]. Another approach in the medical domain can be found in computer-aided diagnosis [DMK+99] where the computer analyses medical image such as radios of a patient and points towards the abnormalities it detects. Due to the non-negligible number of false positives and negatives, these works are far from replacing the experts opinion. Forms of diagnosis can also be found in network management for example, where it is more often called *fault management*. More precisely, fault management has two aspects. The first one, passive, consists in receiving messages from the devices on the network and if an alarm was sent, to understand the cause and react to it. The second one takes an active step by considering that a failing device may not be able to detect its own fault and warn the system. Thus, the fault manager will interact regularly to check the behaviour of the devices. Fault managing therefore requires to do tests, diagnosis and possibly reparation. These diagnosis notions deal with systems that can be extremely complex but that are most often static. One wants to identify the current status of the system from what it is emitting currently, the evolution of the system is not monitored. Diagnosis, as seen in the discrete event systems community, focus in contrast on dynamic systems.

Diagnosis of discrete event systems. For many systems (power systems, manufacturing systems...) one needs to take into account the evolution of the system when analysing it. Such systems can be analysed with the approach from the discrete event systems community. In this approach, while the system is running, one follows a run of the system and tries to deduce the occurrence (or absence of) of a specific event called the *fault*. While one may want to detect any kind of important action of the system, the term "fault" is chosen both to correspond to the name "diagnosis" which, as shown in the previous examples, is mostly used to detect failures, and because faults are often one of the most important elements to detect within a run. They threaten the safety and availability of the system. In many of the systems listed above, a safety issue may lead to catastrophic damages both in terms of economic and human loses. The study of faults in particular is also justified by the fact that every system may, and will, fail. Indeed, the systems we build are increasingly complex and have increasingly intricate interactions with the environment. It is thus extremely difficult when designing a system not to introduce errors and it is almost impossible to predict every reaction the environment will have to the system. Finally, at the very least, failures will occur because of components ageing.

As faults are dangerous, unavoidable and potentially hard to detect (especially in large-scale complex systems), one needs an automated way to detect them. Moreover this method has to be accurate as stopping a system due to a false positive is costly and it must be reactive so that the failure is detected before too many damages were done. In order to react to the fault, one may either (1) wish to optimise the behaviour of a system in the delay before the occurrence of a fault [EMT16], which is particularly useful for systems which components are automatically replaced on a regular basis, thus hopefully before the occurrence of any fault, or (2) try to detect the fault. As one wants to react quickly to the fault, predicting its occurrence before the system even enters a faulty behaviour would be very efficient. This view is the one studied in *prediction* problems [GL09]. However, enabling prediction is a very strong requirement for a system. Detecting the fault *a posteriori* is more likely. The study of diagnosis raises two important issues: deciding whether the system is diagnosable which is called *diagnosability* and, in the positive case, synthesising a *diagnoser* possibly satisfying additional requirements about memory size, detection delays, etc. In the discrete event system context, diagnosis was first defined for finite systems such as *partially observable Labelled Transition Systems* [SSL$^+$95] then was extended to numerous more complex models (*e.g.* Petri nets [CGLS12, BHSS18], pushdown systems [MP09], etc.) and settings (*e.g.* decentralised [DLT00], distributed [HC94]). Also, several contributions, gathered under the generic term of active diagnosis, focus on enforcing the diagnosability of a system [SLT98, TT07, CT08, CP09].

Useful techniques. By observing the previously mentioned works on diagnosis, it appears that some methods and results are recurrent. Let us mention and explain some of them here.

Diagnosability is an *hyper property*, it cannot be checked by analysing every run of the system separately. On the contrary, some runs are *faulty* (contain the fault) while the others are *correct* and we want to compare the observations triggered by faulty runs to the ones produced by correct runs. A key object (*e.g.* see [JHCK01, YL02]) used to decide diagnosability is the *twin-plant*: a new model is built by making the product of the initial model with itself. A run of the twin-plant consists of a pair of runs of the model. As one wants to compare the observations of the runs, the product is made so that the two runs that are followed simultaneously have the same sequence of observations. This way, one can determine if there exist two runs with the same sequence of observations and some appropriate properties only by checking a single run of the twin-plant. Another often used construction (see [SSL$^+$95]) is the *belief* construction. This construction can be seen as an expanded twin-plant or as a form of determinisation of the model: instead of following a pair of runs, the belief automaton instead follows every possible run. More precisely, we keep every state that could be reached with the current sequence of observations, sometimes enhanced with some additional information. This construction is more useful than the twin-plant as it keeps much more information, however it is of exponential size w.r.t. the size of the original model while the twin-plant is only quadratic. The belief automaton by itself gives some information, but can also be used to enrich the initial model. For example,

assume the belief automaton of a stochastic model was built, one can now build the product of this model with its belief automaton [Var99, BK08]. The result, thanks to the determinism of the belief automaton, has the same stochastic behaviour as the original model, however configurations now contain two information: (1) the current state of the associated run in the initial model and (2) the current belief, *i.e.* the set of states that could be reached with the current sequence of observations.

Enriching a model this way is very useful to apply *model-checking* results. Model checking consists in, given a model of a system, verifying if it satisfies a property, often given by a logical formula. The two most famous logics used are LTL (linear temporal logic) [Pnu77] and CTL (computational tree logic) [Eme90]. The first one focuses on properties of individual run while the second is mostly interested in branching properties. As diagnosis is not a branching property we only discuss LTL here. In LTL, one can encode formulae about the future of a run, *e.g.*, a condition will eventually be true, a condition will remain true until another one becomes true, etc. The basic components of a formula are propositional variables whose truth value depends, in our framework, on the current state of the model. The more information is contained within a state, the more precise the use of propositional variables can be. How to verify that a "simple" model satisfies an LTL formula is known for a long time [Var96][1]. Complications occur however when the model is more complex or when the property one wants to check requires more expressive power than what LTL can offer. For stochastic specifications, LTL was extended to pLTL. The extension allows to quantify the measure of the paths satisfying a given LTL formula (the probabilistic operator cannot be nested in the formula). Verifying these formulae is more difficult in terms of complexities. It has been studied both for finite systems [CY95] or infinite ones [EY12]. One can refer to [BK08, Chapter 10] for details about the model checking of probabilistic systems. One important point is that the main source of the complexity of the algorithms is the size of the formula. For example, in [EY12], the qualitative model checking of a recursive Markov chain (a model of infinite-state stochastic system) is PSPACE in the size of the model (and can drop to PTIME under some restrictions) but is EXPTIME in the formula. When studying diagnosis, most problems can be expressed with a simple and fixed formula, which means that the part of the complexity depending on then size of the formula is not our concern.

Another set of techniques that can be used are the results known for POMDP. They have two main interests. First, when studying diagnosis on active systems, some problems can be translated into POMDP problems for which there exist efficient algorithms (as done in [BFH$^+$14]). The second interest of POMDP comes in fact mostly from probabilistic automata (PA), which are a subclass of POMDP. Many problems are known to be undecidable for PA [Paz71, GO10] and due to the simplicity of the PA model, these problems are often easier to use to prove undecidability than problems for POMDP (such as the policy-existence problem under the infinite-horizon average reward criterion [MHC03]).

[1] One technique is to obtain a Büchi automaton that is equivalent to the model and another one that is equivalent to the negation of the property. The intersection of the two non-deterministic Büchi automata is empty if the model satisfies the property.

Finally, as there exist many problems using partial observation, the first thing to do when studying a new notion is to check if there already exists a similar problem for which an analysis was realised. In the positive case, one only has to establish a translation. A relevant example of this was mentioned earlier when we stated that [CK14] was used to solve a monitoring problem. The authors of [CK14] established a polynomial algorithm in order to determine if the distance of the language of two labelled Markov chains is equal to 1. In other words, to decide if, almost surely, given an infinite observed sequence, one can determine which labelled Markov chain emitted it. The algorithm relies on the existence of algorithms to detect when two systems have exactly the same language (distance 0) and that if the models are not at distance 1, then a "part" of them is at distance 0. While diagnosis focuses mostly on finite runs and this problem considers infinite runs, strong links can be identified.

Challenges and objectives. As our global goal is to perform model-based verification, the first question that needs to be tackled in this thesis is the choice of the formalism. This formalism has to include probabilities as we want to be able to quantify the specification. But some points are still open: we must determine whether the model incorporates non-determinism, represents infinitely many states or expresses efficiently concurrent behaviours for example. Moreover, we intend to work on partial observation problems. This requires the model to select what observation is associated with a run.

Our second issue lies in the choice of the problems to focus on. Many notions of diagnosis have been defined over the years with different goals in mind. We have to find a set of appropriate qualitative/quantitative diagnosis notions that encompasses the important, already known, notions that focus on realistic relevant issues, and that is coherent as a whole. Moreover, the formal definitions of the problems we work on must be carefully chosen. Indeed, mixing partial observation, probabilities and control quickly leads to undecidability results (as is the case for PA, mentioned earlier). A slight modification of the definitions may strongly modify the complexity. For example we will see a case, where inverting two quantifiers turn a problem of complexity PTIME to an undecidable one.

Once the properties are defined, our goal is to establish precisely the complexities of verifying if the model satisfies the chosen specification. We are also interested in determining how to modify the system so that it satisfies the properties. This gives two main approaches, a passive one associated with observation and an active one associated with observation and control.

Outline. This thesis is organised as follows.

- In Chapter 2, we introduce notations useful all along the document. While it does not contain any result *per se*, it includes the definitions of the notions of diagnosis that we introduced. The choice of appropriate definitions is already a contribution. In the second part of this chapter, we present a state of the art on the diagnosis problem.

- In Chapter 3, we realise a semantical analysis of the problems of diagnosability we defined in Chapter 2. More precisely, we first present the notion of diagnoser, *i.e.* the function realising the diagnosis, and prove the equivalence between the existence of a diagnoser and the diagnosability of a system. We also establish the relations between the various notions of diagnosability and, when possible, characterise these notions. The semantical analysis of a problem as done here is very important as understanding the problems is the first step to solving them. This chapter is based on [BHL14, BHL16a, BHL16b].

- In Chapter 4, we focus on our simplest model, representing finite stochastic systems. Using the finiteness, we strengthen our characterisations of the diagnosability notions and use them in order to establish algorithms to decide the problems when possible or to prove undecidability in the opposite case. We also show how to build diagnosers using finite memory. This chapter develops contributions from [BHL14, BHL16a].

- In Chapter 5, we turn to systems with infinitely many states. We cannot use the characterisations obtained in Chapter 4, but the results of Chapter 3 still hold. We study different models and clearly observe the increase in difficulty compared to finite-state models. We still manage to obtain decidability results for one model. This chapter extends [BHL16b].

- In Chapter 6, we consider controllable systems. Using the control, one can ensure properties for the system. However, controlling the system with one objective in mind can have negative side-effects. For example, ensuring diagnosability can increase the likelihood of faults within the system. We are therefore interested in combining multiple objectives, one of them being diagnosability. This chapter is based on [BHL17b].

- In Chapter 7, we instead focus on another partial observation problem: opacity, which, on many aspects, appears like a dual of diagnosability. We introduce the notion and explain the impact opacity has on the choice of the framework: we consider here active systems as is done in Chapter 6, however the type of control is different as the controller is not interpreted in the same manner. We define multiple measures of opacity and explain how to maximise or minimise them when possible. The results of this chapter were published in [BHL17a].

Other works. In order to limit the number of frameworks and problems to define and to give a better coherence to the thesis, some of the works we realised are not detailed in this document. We give a short description of these results here.

The results of [BHL14] serve as a foundation to our analysis of diagnosability. They are thus, for the most part, necessary for this thesis and are therefore developed in Chapter 3 and Chapter 4. However, this work also contains contributions on *prediction* and *prediagnosis*. Prediction describes the ability to detect the fault before its occurrence. It had been shown to be in NLOGSPACE for logical systems [GL09]. While using

probabilities usually increases the difficulty, we gave an NLOGSPACE algorithm for the prediction problem in stochastic systems. The authors of [CK15] present a similar result with a notion called "prognosis". While prediction is limited to the detection of faults before their occurrence, diagnosis is itself limited to their detection after the occurrence. That is why we also introduced prediagnosis where one is allowed to either predict or diagnose the fault. Any predictable or diagnosable system is thus prediagnosable, but the converse does not hold. We showed that prediagnosability (*i.e.* the problem of deciding if a system is prediagnosable) is PSPACE-complete.

For the diagnosability notions studied in this thesis, faults are permanent: once a fault is triggered, any following behaviour is faulty. However, one may want to consider faults that are only temporary. For example, a model could contain the possibility of a reparation. This raises many new problems: one may wish to detect the fault before the reparation, to count the number of faults occurring within the system, etc. These questions were investigated in [FHLM18] in a non-stochastic setting. While diagnosability of permanent faults is known to be in NLOGSPACE for logical systems, we showed that with repairable faults, it becomes PSPACE-complete. We also discussed multiple methods to count faults and presented among other things an NLOGSPACE algorithm to decide if one can count the number of faults while having a delay of at most one count of fault.

One of the non-stochastic framework where diagnosability was studied is Petri nets (PN) [CGLS12, BHSS18]. For bounded PN, the usual method to solve diagnosability is to build the reachability graph of the net, which is an automaton structure representing the behaviour of the PN and then to use existing results on this kind of models [JHCK01, YL02]. The issue with this method is the size of the reachability graph. In order to face this issue, many works try to abstract the graph in order to reduce its size while keeping the relevant information. In a partial observation setting, this led to the introduction of the *basis reachability graph* [CGS09]. In [LGS18], we showed how to extend this abstraction to unbounded PN, calling the new object "Basis coverability graph". We established some results about the properties of the basis coverability graph and explained how to use it to solve diagnosability of unbounded PN.

Chapter 2

Preliminaries

The first step of the theoretical analysis of a problem is the definition of the framework. The chosen framework possesses different properties depending on the specificities of the system one wishes to study. Some frameworks are thus better adapted to represent concurrency, to express infinite-state systems... In the first section of this Chapter, we introduce the main definitions that are used throughout the thesis. More precisely, in Subsection 1.1, we recall some definitions and results of descriptive set theory. We then define in Subsection 1.2 the main probabilistic model used in this document and define a probabilistic measure. In Subsection 1.3, we explain how partial observation can be added to this model. These definitions give a framework for various problems linked to partial observation. In this section, we also give definitions related to diagnosis, which is the main partial observation problem studied during this thesis. Diagnosis corresponds to a family of questions, focusing on the identification of a faulty behaviour within the system. We explain in Subsection 1.4 how the notion of fault can be formalised in our model. Finally, in Subsection 1.5, we discuss different notions of diagnosability. These notions are used to capture when and how the faulty (or correct) behaviour of the system must be identified, and with which accuracy. Section 2 finally presents a state of the art on diagnosis.

1 Framework

Let us start with a few general notations. We denote by $\mathbb{N}$ the set of natural numbers, $\mathbb{Q}$ the set of rational numbers and $\mathbb{R}$ the real numbers. For a finite alphabet Σ, we denote by Σ^* (resp. Σ^ω) the set of finite (resp. infinite) words over Σ. ε represents the empty word.

1.1 Descriptive set theory

Descriptive set theory [Mos80, Chapter 1] defines and studies classes of "well-behaved" sets. These sets having good properties, they have applications in many areas. In this thesis, they have two main applications. First, they are used to evaluate the complexity of some problems. This is achieved using a hierarchy ranking these sets: the higher a

set is in the hierarchy, the more complex it is. Therefore if we can express a property with a set, the problem complexity can be related to the set complexity. Secondly, these sets form a building block of the formal definition of the stochastic behaviour of our model.

Let us first recall some standard facts about Borel sets.

Definition 2.1. *Given a space X, the set Y is a* topology *on X if*

- $X \in Y$ *and* $\emptyset \in Y$,
- Y *is stable by union,* i.e. *given $(O_i)_{i\in I}$ a family of elements of Y, $\cup_{i\in I} O_i \in Y$,*
- Y *is stable by finite intersection,* i.e. *given $(O_i)_{i\in I}$ a finite family of elements of Y, $\cap_{i\in I} O_i \in Y$,*

We call the sets in Y the open *sets.*

Example 2.1. *Consider the space of infinite words over two letters $\{a,b\}^\omega$. On such a space, the usual topology uses the notion of cylinder. Given a finite word w of $\{a,b\}^*$, the cylinder of w is the set of infinite words extending w, $\mathsf{Cyl}(w) = w\{a,b\}^\omega$. One obtains a topology by choosing the set of cylinders as its basic components and by adding the sets created through union and finite intersection. $\{a,b\}^\omega$ belongs to this topology as it is the cylinder of ε, the empty set can be obtained as the intersection of the cylinders of 'a' and of 'b'.*

Under this topology, the set of words containing at least k 'a', for $k \in \mathbb{N}$ is an open set as it is a countable union of cylinders.

Given a space X and a topology Y on X, we define the *Borel hierarchy* (represented in Figure 2.1) as the three classes of sets Σ_n^0, Π_n^0 and Δ_n^0 obtained inductively by

- $\Sigma_1^0 = Y$ is the set of open sets;
- $\forall n \geq 1$, a set belongs to Π_n^0 if its complement belongs to Σ_n^0
- $\forall n \geq 2$, O is in Σ_n^0 if there exists a family $(O_i)_{i\in I}$ of elements of Π_{n-1}^0 such that $O = \cup_{i\in I} O_i$;
- $\forall n \geq 1$, $\Delta_n^0 = \Pi_n^0 \cap \Sigma_n^0$.

The sets in Π_1^0 are called *closed*, the sets Σ_2^0 and Π_2^0 are respectively called F_σ and G_δ. A *Borel set* is a set belonging to some level of the Borel hierarchy. The set of Borel sets is denoted $\mathfrak{B}$.

Example 2.2. *Continuing Example 2.1, the set composed of the single word a^ω is closed. Indeed, its complement is $\cup_{n\in\mathbb{N}} \mathsf{Cyl}(a^n b)$ which is an open set. The set of words with infinitely many 'a' is neither open, nor closed. It however belongs to G_δ. Indeed, it is the complement of the set of words ending by b^ω. As this set of words is an infinite union of closed set (each closed set containing a single infinite word), it belongs to F_σ, thus its complement is a G_δ set.*

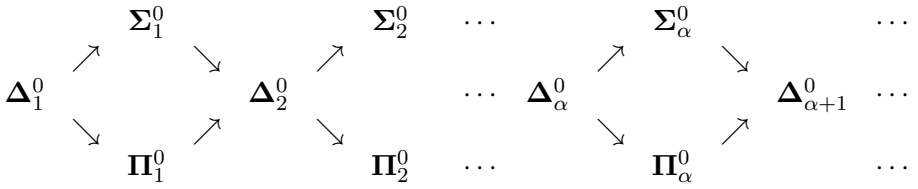


Figure 2.1: Representation of the Borel hierarchy.

In our models, we need to measure the probabilities of certain events. This is done thanks to Carathéodory's extension theorem [ADD99]. Before stating this theorem, we need to introduce some definitions used in measure theory.

Definition 2.2. *Given a space X, a* ring of sets *R of X is a subset of the powerset of X satisfying:*

- $\emptyset \in R$;
- *R is closed under pairwise union,* $\forall A, B \in R, A \cup B \in R$;
- *R is closed under relative complements,* $\forall A, B \in R, A \setminus B \in R$.

Definition 2.3. *Given a space X, a* σ-algebra *S of X is a subset of the powerset of X satisfying:*

- $\emptyset \in S$;
- *S is closed under countable union,* $\forall (A_i)_{i\in\mathbb{N}} \in S, \bigcup_{i\in\mathbb{N}} A_i \in S$;
- *S is closed under complement,* $\forall A \in S, S \setminus A \in S$.

By definition, a σ-algebra is a ring of sets. Observe that the set of Borel sets $\mathfrak{B}$ is a σ-algebra. More precisely, it is the σ-algebra *generated* by the open sets (*i.e.* it is the smallest σ-algebra containing the open sets).

Definition 2.4. *Given a ring of sets R on a space X, a* pre-measure *μ on R is a function $\mu : R \mapsto [0, +\infty]$ such that:*

- $\mu(\emptyset) = 0$;
- *for all countable family of sets of R pairwise disjoint, $(A_i)_{i\in\mathbb{N}}$, we have*

$$\mu\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mu(A_n).$$

A pre-measure μ is called σ-finite *if there exist a countable number of sets $A_1, A_2 \cdots \in R$ such that $X = \bigcup_{k=1}^{\infty} A_k$ and for all $k \in \mathbb{N}, \mu(A_k) < \infty$.*

If R is a σ-algebra, then μ is called a measure.

A measure μ is called inner regular *if for every set E, we have*

$$\mu(E) = \sup\{\mu(F) \mid F \subseteq E \wedge F \text{ is a closed set}\}.$$

In the current work, we require a measure, yet only a pre-measure can efficiently be defined. Fortunately, Carathéodory's extension theorem allows to bridge the gap.

Theorem 2.1 ([ADD99])**.** *Let R be a ring on a space X, μ be a pre-measure on R. There exists a measure μ' extending μ on the σ-algebra generated by R. Moreover, if μ is σ-finite, then μ' is unique and also σ-finite.*

Example 2.3. *Let us continue Example 2.1 and consider the ring of sets R generated by the topology based on the cylinders. We define the pre-measure μ on this ring as the only pre-measure satisfying $\forall n \in \mathbb{N}, w \in \{a,b\}^n, \mu(\mathsf{Cyl}(w)) = \frac{1}{2^n}$.*

One could interpret the represented system as an infinite number of coin flips, each result ('a' or 'b') having $\frac{1}{2}$ probability. μ gives the probability of cylinders (a given finite number of flips) and finite unions of them.

The pre-measure μ is σ-finite as $\mu(\{a,b\}^\omega) = 1$. According to Carathéodory's extension theorem, there is thus an unique measure μ' extending μ on the σ-algebra generated by R: the Borel sets. Observe that μ' is inner regular.

1.2 Probabilistic Labelled Transition Systems

We now define the model that represent the system. The choice of the model depends on the properties one wishes to have. Petri nets [Dia09] for example, efficiently represent concurrent systems. Another possibility is to use automata structures as in the seminal work of [SSL+95] formally defined by:

Definition 2.5. *A* labelled transition system *(LTS) is a tuple $\mathbb{A} = \langle Q, q_0, \Sigma, T\rangle$ where:*

- *Q is a countable set of states with $q_0 \in Q$ the initial state;*
- *Σ is a finite set of events;*
- *$T \subseteq Q \times \Sigma \times Q$ is a set of transitions;*

Informally, the states of Q represent the different configurations the system can be in, an event of Σ is an action that can be taken by the system (sending a request, activating a component...) and T describes how this action affects the system.

Formally, we write $q \xrightarrow{a} q'$ when there exists a transition $(q, a, q') \in T$; this transition is then said to be *enabled* in state q. We assume all LTS we consider are *live*, *i.e.* in every state of the LTS at least one transition is enabled. This ensures the system will not reach a deadlock position and stop activating events. A *run* ρ of an LTS $\mathbb{A}$ is a (finite or infinite) sequence $\rho = q_0 a_0 q_1 \dots$ such that for all $i \geq 0$, $q_i \in Q$, $a_i \in \Sigma$ and when q_{i+1} is defined, $q_i \xrightarrow{a_i} q_{i+1}$. A run thus represents the evolution of a system over time. The notion of run can be generalised, starting from an arbitrary state q. Given

an LTS $\mathbb{A}$, we write $\Omega^{\mathbb{A}}$ for the set of all infinite runs starting from q_0. We only write Ω when the LTS $\mathbb{A}$ is clear from context. When it is finite, ρ ends in a state that we denote $\mathsf{last}(\rho)$ and its *length*, denoted by $|\rho|$, is the number of events occurring in it. Given a finite run $\rho = q_0 a_0 q_1 \dots q_n$ and a (finite or infinite) run $\rho' = q_n a_n q_{n+1} \dots$ starting in $\mathsf{last}(\rho)$, we call concatenation of ρ and ρ' the run $\rho\rho' = q_0 a_0 q_1 \dots q_n a_n q_{n+1} \dots$. The run ρ is then a *prefix* of $\rho\rho'$, which we denote by $\rho \preceq \rho\rho'$. The *cylinder* generated by a finite run ρ consists of all the infinite runs that extend ρ: $\mathsf{Cyl}(\rho) = \{\rho' \in \Omega \mid \rho \preceq \rho'\}$. The sequence associated with $\rho = q a_0 q_1 \dots$ is the word $\sigma_\rho = a_0 a_1 \dots$, and we write indifferently $q \overset{\rho}{\Rightarrow}$ or $q \overset{\sigma_\rho}{\Longrightarrow}$ (resp. $q \overset{\rho}{\Rightarrow} q'$ or $q \overset{\sigma_\rho}{\Longrightarrow} q'$) for an infinite (resp. finite) run ρ starting in q (resp. and ending in q'). A state q is *reachable* (from the initial state q_0) if there exists a run ρ such that $q_0 \overset{\rho}{\Rightarrow} q$, which we alternatively write $q_0 \Rightarrow q$. The language of an LTS $\mathbb{A}$ consists of all infinite words that label runs of $\mathbb{A}$ and is formally defined as $\mathcal{L}^\omega(\mathbb{A}) = \{\sigma \in \Sigma^\omega \mid \exists\ q_0 \overset{\sigma}{\Rightarrow}\}$. A *bottom strongly connected component* (BSCC) of an LTS is a strongly connected component from which no state outside of the BSCC are reachable.

Example 2.4. *Consider the LTS represented in Figure 2.2. It contains three states.* q_0 *is the initial state, representing the machine waiting to receive an order. When such an order occurs the run takes the transition labelled by the 'coin' event and enters the second state. In this second state, the machine is preparing the coffee, it has the possibility to add sugar, and after a certain amount has been added, it returns to its initial state by giving a coffee. In this operating state, the machine can also commit an error, event '***f***', leading to a faulty state* f_1*. In* f_1*, the machine cannot serve coffee any more and sends an 'out of order' signal. A normal use of this system by a consumer is given for example by the run* $\rho = q_0$ *coin* q_1 *sugar* q_1 *coffee* q_0*.*

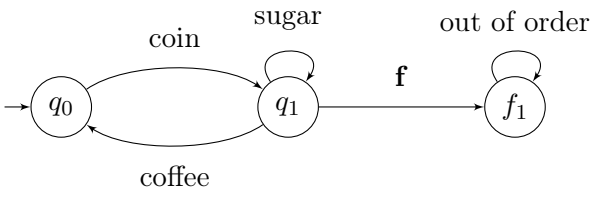


Figure 2.2: An LTS representing a coffee machine. q_0 is the initial state, which is represented by the incoming arrow. Transitions between two states are labelled by the event associated with the transition.

In order to represent the unpredictability of the environment and to neglect events that have a null probability of occurring, we want to represent stochastic behaviours. To do so, the model must be enriched using probabilities. More precisely, this is achieved by adding a probability matrix indicating how the transitions, labelled by events, are randomly chosen.

Definition 2.6. *A* probabilistic labelled transition system *(pLTS) is a tuple* $\mathcal{A} = \langle Q, q_0, \Sigma, T, \mathbf{P} \rangle$ *where:*

- $\mathbb{A} = \langle Q, q_0, \Sigma, T \rangle$ *is an LTS;*
- $\mathbf{P}$ *is the transition matrix from* T *to* $\mathbb{Q}_{>0}$ *fulfilling for all* $q \in Q$*:*

$$\sum_{(q,a,q') \in T} \mathbf{P}[q, a, q'] = 1 \ .$$

The LTS $\mathbb{A}$ *is called the* underlying labelled transition system *of* $\mathcal{A}$*.*

Note that since we assume the state space to be at most countable, a pLTS is by definition at most countably branching: in every state q, only countably many transitions are enabled, so that the summation $\sum_{(q,a,q') \in T} \mathbf{P}[q, a, q']$ is well-defined. The definitions introduced for LTS are naturally lifted to pLTS. Given a countable set Z, a distribution on Z is a mapping $\mu : Z \to [0, 1]$ such that $\sum_{z \in Z} \mu(z) = 1$. The support of μ is $\mathsf{Supp}(\mu) = \{z \in Z \mid \mu(z) > 0\}$. If $\mathsf{Supp}(\mu) = \{z\}$ is a single element, μ is a Dirac distribution on z written $\mathbf{1}_z$. We denote by $\mathsf{Dist}(Z)$ the set of distributions on Z. The transition matrix defines in every state q a distribution on the transitions whose support are exactly the transitions enabled in q.

Example 2.5. *Consider the pLTS represented in Figure 2.3. Its underlying LTS is represented in Figure 2.2. The difference is thus that probabilities were added on the transitions so that the sum of the probabilities exiting any state is equal to one. The run* ρ *that was described as being a normal use of the system in Example 2.4 can now be associated with a probability, which is the product of the probability of the events it triggered. Here,* $1 \times 0.29 \times 0.7 = 0.203$*. This result, being low, seems to point out that, in this representation, most consumers do not take exactly one unit of sugar in their coffee or do not even get their coffee.*

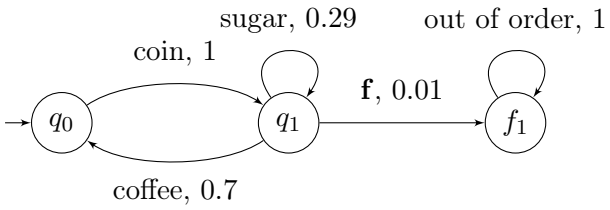


Figure 2.3: A pLTS representing a coffee machine. The probability of a transition is given next to the event labelling it.

We now use the probabilities within the system to formally define the probability measure that are used on runs. This is done using the descriptive set theory presented in Subsection 1.1. The construction of the measure uses the Carathéodory's extension theorem, similarly to what is done in Example 2.3.

Here, the set space X we are interested in is the set of all infinite runs Ω. The open set Σ_1^0 is built from the cylinders: Σ_1^0 is the smallest set containing $\emptyset$, Ω, such that for all finite run ρ, $\mathsf{Cyl}(\rho) \in \Sigma_1^0$, and such that Σ_1^0 is stable by union and finite intersection. The complement of a cylinder is a finite union of cylinders, the complement of an (potentially infinite) union is an (potentially infinite) intersection and the complement of a finite intersection is a finite union. A set F is thus closed if and only if $F = \bigcap_{n\in\mathbb{N}} O_n$ where O_n is a union of cylinders. Therefore an F_σ set F can be written as $F = \bigcup_{m\in\mathbb{N}} \bigcap_{n\in\mathbb{N}} O_{m,n}$ where $O_{m,n}$ is a union of cylinders whose associated paths have length n. Without loss of generality, the sequence of closed sets may be chosen as a non-decreasing sequence. The cylinders are thus used as the basis to build a Borel hierarchy on Ω.

Given a pLTS $\mathcal{A}$, in order to have a probabilistic measure spanning all the sets of infinite runs of the Borel hierarchy generated by the cylinders, we define a pre-measure $\mathbb{P}_\mathcal{A}$ on the ring of sets generated by the open sets by: for all finite run $q_0a_0q_1\dots q_n$,

$$\mathbb{P}_\mathcal{A}(\mathsf{Cyl}(q_0a_0q_1\dots q_n)) = \mathbf{P}[q_0, a_0, q_1]\cdots\mathbf{P}[q_{n-1}, a_{n-1}, q_n]$$

and for all finite sequence of pairwise disjoint cylinders $A_1, \dots, A_n$,

$$\mathbb{P}_\mathcal{A}(\cup_{i=1,\dots,n} A_i) = \sum_{i=1,\dots,n} \mathbb{P}_\mathcal{A}(A_i) \ .$$

According to Carathéodory's extension theorem, this pre-measure can be extended uniquely on the whole σ-algebra generated by this ring. We still write $\mathbb{P}_\mathcal{A}$ for the obtained measure. Moreover, when $\mathcal{A}$ is fixed, we may omit the subscript. To simplify, for ρ a finite run, we sometimes abuse notation and write $\mathbb{P}(\rho)$ for $\mathbb{P}(\mathsf{Cyl}(\rho))$. If R is a (countable) set of finite runs such that no run is a prefix of another one, we write $\mathbb{P}(R)$ for $\sum_{\rho\in R}\mathbb{P}(\rho)$ which is consistent since all intersections of associated cylinders are empty.

1.3 Partial observation

In partial observation problems, one considers an observer of the system (a user, an attacker...) who perfectly knows the model, yet only partially observes its behaviours. In this case, one needs to formalise which information can be observed. This allows to associate with every run of the system, an observation or sequence of observations.

The first question is to determine what part of a run gives an information: events, states or both. In fact, these three options are equivalent in terms of expressibility. Thus, for every problem, one selects the option that is the more efficient at representing the specificities of the problem. For example, when studying diagnosis, we try to detect a faulty action of the system. As this faulty action is an event, observations are put on events. This is the option we follow in most chapters of this thesis, see Chapter 7, page 195, for a model with observations on states.

Events represent actions taken by the system. A formalisation of the observation could be made by distinguishing internal and external actions. This would mean that some actions occur within the system and are thus *unobservable* while others are done

publicly and are thus *observable*. Formally, one partitions the set of events Σ into two disjoint sets Σ_o and Σ_u, the sets of observable and unobservable events, respectively.

Example 2.6. *Consider the pLTS of Figure 2.3. Clearly, introducing a coin, adding sugar to the cup and receiving the coffee are observable actions. The other two are more subject to discussion. Depending on the fault and the sensors within the system,* $\mathbf{f}$ *could be considered observable or unobservable. Indeed, if* $\mathbf{f}$ *stands for the explosion of the machine, it would clearly be observable, however if* $\mathbf{f}$ *is an internal error, detecting it requires the existence of an appropriate sensor within the system. The status of the last event also depends on which fault occurred. For our example, let us assume* $\mathbf{f}$ *is unobservable, but the failure of the system is detected, allowing the machine to publicly send an 'out of order' message. In this case, the run "*q_0 *coin* q_1 *sugar* q_1 $\mathbf{f}$ f_1 *out of order" produces the sequence of observations "coin sugar out of order".*

A more sophisticated method than this partition would be to equip the pLTS with a *mask function*. This mask function associates every event with an observation, taken from an observation alphabet. This function can map an event to ε meaning that the event is unobservable or project multiple events onto the same observation, making them indistinguishable. When using a mask function, Σ_o is the observation alphabet and Σ_u is the set of unobservable events (*i.e.* events which observation is ε).

Example 2.7. *Consider the pLTS of Figure 2.3 again. We use the observation alphabet* $\Sigma_o = \{$*coin, coffee, beep*$\}$ *and the mask function* $\mathcal{P}$ *such that* $\mathcal{P}($*coin*$) =$ *coin,* $\mathcal{P}($*coffee*$) =$ *coffee,* $\mathcal{P}(\mathbf{f}) = \varepsilon$ *and* $\mathcal{P}($*sugar*$) = \mathcal{P}($*out of order*$) =$ *beep. Here, we see that two of the events are indistinguishable as they share the same observation beep. When a beep is produced, a user does not know whether the machine is out of service or if it is adding more sugar. In other words, the infinite runs "*q_0 *coin* $q_1\mathbf{f}f_1($*out of order* $f_1)^\omega$*" and "*q_0 *coin* $q_1($*sugar* $q_1)^\omega$*" share the same observation sequence "coin sugar beep*$^\omega$*".*

Observe that the mask function setting generalises the partition discussed above. Indeed, the partition is mimicked by the mask function which projects every unobservable event to ε and every observable event onto itself. We now define multiple notations using the mask function formalism. Due to the previous remark, these definitions can easily be applied to the partition setting. In the future chapters, we mostly use partitions for simplicity. We state explicitly when we use mask functions.

Given an observation alphabet, a mask function is a mapping $\mathcal{P} : \Sigma \to \Sigma_o$. It is extended to words from Σ^* inductively by: $\mathcal{P}(\varepsilon) = \varepsilon$ and $\mathcal{P}(\sigma a) = \mathcal{P}(\sigma)\mathcal{P}(a)$. We write $|\sigma|_o$ for the *observable length* of σ, that is $|\mathcal{P}(\sigma)|$. The observable length of a run ρ, denoted $|\rho|_o$, is the observable length of its associated sequence. Given a run ρ and its sequence σ_ρ we sometimes use $\mathcal{P}(\rho)$ for $\mathcal{P}(\sigma_\rho)$. When σ is an infinite word over Σ, its projection (resp. observable length) is the limit of the projections (resp. observable length) of its finite prefixes. Given $a \in \Sigma_o$, $|\sigma|_a$ is the number of occurrences of a in σ. As usual the mask function $\mathcal{P}$ is extended to languages: for $L \subseteq \Sigma^* \cup \Sigma^\omega$, $\mathcal{P}(L) = \{\mathcal{P}(\sigma) \mid \sigma \in L\}$.

With respect to the mask function $\mathcal{P}$, a pLTS $\mathcal{A}$ is said *convergent* if there is no infinite sequence of unobservable events from any reachable state: $\mathcal{L}^\omega(\mathcal{A}) \cap \Sigma^*\Sigma_u^\omega = \emptyset$.

When $\mathcal{A}$ is convergent, for every $\sigma \in \mathcal{L}^\omega(\mathcal{A})$, $\mathcal{P}(\sigma) \in \Sigma_o^\omega$. In the rest of the thesis we assume that pLTS are convergent. We refer to a *sequence* for a finite or infinite word over Σ, and to an *observed sequence* for a finite or infinite word over Σ_o. The projection of a sequence onto Σ_o is thus an observed sequence. The prefix of length $n \in \mathbb{N}$ of an observed sequence w is denoted $w_{\leq n}$.

We now define the notion of *signalling runs*. They correspond to the finite runs which last event was observable (*i.e.* finite runs $q_0a_0q_1 \cdots a_{n-1}q_n$ such that $\mathcal{P}(a_{n-1}) \neq \varepsilon$). Signalling runs are precisely the relevant runs w.r.t. partial observation issues since each observable event provides additional information about the execution to an external observer. In the sequel, $\mathsf{SR}(\mathcal{A})$ denotes the set of signalling runs of the pLTS $\mathcal{A}$, and $\mathsf{SR}_n(\mathcal{A})$ the set of signalling runs of observable length n. The pLTS is dropped from the notation when it is clear from context. Since we assume that the pLTS are convergent, for every $n > 0$, SR_n is equipped with a probability distribution defined by assigning measure $\mathbb{P}(\rho)$ to each $\rho \in \mathsf{SR}_n$. Given ρ a finite or infinite run, and $n \leq |\rho|_o$, $\rho_{\downarrow n}$ denotes the unique prefix of ρ that belongs to SR_n. For convenience, the empty run q_0 is defined as the single signalling run of null length. For an observed sequence $w \in \Sigma_o^*$, we define its cylinder $\mathsf{Cyl}(w) = w\Sigma_o^\omega$ and the associated probability $\mathbb{P}(\mathsf{Cyl}(w)) = \mathbb{P}(\{\rho \in \Omega \mid \mathcal{P}(\rho_{\downarrow|w|}) = w\}) = \mathbb{P}(\{\rho \in \mathsf{SR}_{|w|} \mid \mathcal{P}(\rho) = w\})$, often shortened as $\mathbb{P}(w)$.

1.4 Fault and ambiguity

We now give definitions and notations for a partial observation problem, which is particularly of interest to us, diagnosis. Diagnosis focuses on the detection of a special unobservable event called the *fault* $\mathbf{f} \in \Sigma_u$ thanks to the observations received from the system. Let us now classify runs depending on whether they contain a fault or not. A run ρ is *faulty* if its associated sequence σ_ρ contains $\mathbf{f}$, otherwise it is *correct*. For $n \in \mathbb{N}$, we write F_n (resp. C_n) for the set of infinite runs whose signalling prefix of observable length n is faulty (resp. correct). We further define the sets of all finite faulty and correct signalling runs F and C and the sets of infinite faulty and correct runs $\mathsf{F}_\infty = \bigcup_{n\in\mathbb{N}} \mathsf{F}_n$ and $\mathsf{C}_\infty = \bigcup_{n\in\mathbb{N}} \mathsf{C}_n$. A run ρ is a *minimal faulty run* if it is a faulty run and there does not exist a prefix ρ' of ρ that is a faulty run. We write, for all $n \in \mathbb{N}$, minF_n for the set of minimal faulty runs of length n and $\mathsf{minF} = \bigcup_{n\in\mathbb{N}} \mathsf{minF}_n$ for the set of all minimal faulty runs.

Given two states q and q' and an observation $a \in \Sigma_o$, we write $q \Rightarrow^a q'$ if there exists a run $\rho = q_0a_0q_1 \dots q_n$ with $q_0 = q$, $q_n = q'$, $\rho \in \mathsf{SR}_1$ and $\mathcal{P}(\rho) = a$. We also write $q \Rightarrow_f^a q'$ (resp. $q \Rightarrow_c^a q'$) if there exists a faulty (resp. correct) run $\rho = q_0a_0q_1 \dots q_n$ with $q_0 = q$, $q_n = q'$, $\rho \in \mathsf{SR}_1 \cap \mathsf{F}$ (resp. $\rho \in \mathsf{SR}_1 \cap \mathsf{C}$) and $\mathcal{P}(\rho) = a$.

Except explicit mention of the opposite, we assume that the state space Q of $\mathcal{A}$ is partitioned into correct states and faulty states: $Q = Q_f \uplus Q_c$ such that faulty (resp. correct) states, *i.e.* states in Q_f (resp. Q_c), are only reachable by faulty (resp. correct) runs. This can be done without loss of generality. Indeed, considering a pLTS $\mathcal{A} = \langle Q, q_0, \Sigma, T, \mathbf{P}\rangle$, we can build the pLTS $\mathcal{A}' = \langle Q', (q_0, \bot), \Sigma, T', \mathbf{P}'\rangle$ where:

- $Q' = Q \times \{\bot, \top\}$,

- $((q,i),a,(q',j)) \in T'$ if $(q,a,q') \in T$, and either $a \neq \mathbf{f}$ and $i = j$ or $a = \mathbf{f}$ and $j = \top$,
- for $((q,i),a,(q',j)) \in T'$, $\mathbf{P}'((q,i),a,(q',j)) = \mathbf{P}(q,a,q')$.

Denoting $Q_f = Q \times \{\top\}$ and $Q_c = Q \times \{\bot\}$, the pLTS $\mathcal{A}'$ has the same behaviour as $\mathcal{A}$ and verifies the partition mentioned above as a run enters Q_f if and only if its last transition was a fault and can never go back to Q_c.

While the correct or faulty status of the current run may not be known to the observer, the observed sequences carry some information about them. An infinite (resp. finite) observed sequence $w \in \Sigma_o^\omega$ (resp. Σ_o^*) is called *ambiguous* if there exists a correct infinite (resp. signalling) run ρ and a faulty infinite (resp. signalling) run ρ' such that $\mathcal{P}(\rho) = \mathcal{P}(\rho') = w$. Otherwise, it is either *surely faulty*, or *surely correct* depending on whether $\mathcal{P}^{-1}(w) \cap \mathsf{SR} \subseteq \mathsf{F}$ or $\mathcal{P}^{-1}(w) \cap \mathsf{SR} \subseteq \mathsf{C}$. A run is ambiguous, surely correct or surely faulty if its observed sequence is ambiguous, surely correct or surely faulty respectively. We write Sf_∞ (resp. Sc_∞) for the set of infinite surely faulty (resp. correct) runs. In addition Sf_n (resp. Sc_n) is the set of infinite runs whose signalling prefix of observable length n is surely faulty (resp. correct).

Example 2.8. *Consider the pLTS of Figure 2.3 associated with the mask function $\mathcal{P}$ such that $\mathcal{P}(\mathbf{f}) = \varepsilon$, $\mathcal{P}(\textit{out of order}) = \mathcal{P}(\textit{sugar}) = \textit{beep}$ and every other event is projected on itself.*

First observe that this pLTS satisfies the partition between faulty and correct states. Indeed, a run ends in f_1 iff it is faulty.

The observed sequence "coin beep" is ambiguous as it can be generated by the correct run "q_0 coin q_1 sugar q_1" and the faulty signalling run "q_0 coin q_1 $\mathbf{f}$ f_1 out of order f_1". Extending this observed sequence with other observations of beep maintains the ambiguity. Extending it with 'coffee' however makes it surely correct. There does not exist any surely faulty observed sequence in this pLTS.

1.5 Which diagnosis for pLTS?

The goal of diagnosis is the automatic detection of the fault event. This detection is performed by a diagnoser, a function observing the system and giving its verdict. Formally, a diagnoser is a function $D : \Sigma_o^* \to \{?, \top, \bot\}$ assigning to every finite observed sequence a verdict. Informally, when a diagnoser outputs ? it does not provide any information, while $\top$ means that the diagnoser announces a fault and $\bot$ that the diagnoser provides some information about the correctness of the current run. Multiple notions of diagnoser, and thus of diagnosis, can be defined depending on the properties that we require. In logical systems, three main features of the diagnoser are considered: *verdict*, *correctness* and *reactivity*. Verdict specifies the nature of the information the diagnoser provides along the run: it may only be related to detection of faults or may also assert that (some prefix of) the run does not include a fault. Correctness specifies that when the diagnoser outputs a verdict, this verdict holds. Reactivity expresses the regularity at which the diagnoser must provide information about the status of the run.

The aim of this section is to define appropriate verdict, correctness and reactivity requirements for probabilistic systems. We start with informal explanations that also motivate the need for considering different variants of diagnosis. We present these variants as decision problems which are intuitively easier to understand and simpler to use. We make the link with diagnosers in Chapter 3.

In seminal works about probabilistic systems, the verdict is limited to fault detections and the reactivity is usually relaxed by requiring that when a fault occurs, a diagnoser *almost surely* detects it after a finite delay [TT05]. Let us look at the pLTS of Figure 2.4. One cannot detect that the run $q_0\mathbf{f}(f_1a)^\omega$ is faulty due to the correct run $q_0u(q_1a)^\omega$ with same observed sequence a^ω. However with probability 1, a faulty run will produce a 'b' and thus almost all faulty runs are unambiguous, so that faults are almost surely detected. On the other hand, one cannot provide any information about the single correct run $q_0u(q_1a)^\omega$ since its observed sequence is ambiguous as well as any of its prefix. Observe that the notion of ambiguity described here is qualitative: the observation of the correct run is considered ambiguous even though the probability to be faulty, conditioned on the observation, converges to 0.

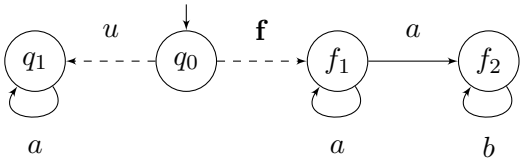


Figure 2.4: Detecting faults but not correct runs. When probabilities are not specified, we assume uniform distributions. Dashed edges are used for unobservable transitions. For observable transitions, the observation given by the mask function labels the edge.

In order to examine which verdict could be provided about correct runs, let us look at the pLTS of Figure 2.5. The sequence a^n is ambiguous. However up to the $n-1^{th}$ observation, all the runs that correspond to this observed sequence were correct which is a useful information for instance to restart later the system from a correct state. Along the (surely correct) observed sequence a^ω, the observer can always deduce that longer and longer prefixes of the run were correct while never being able to assert that the current run is correct.

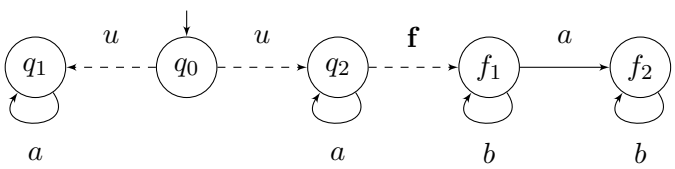


Figure 2.5: Detecting correctness for longer and longer prefixes of correct runs.

The correctness requirement may be specified in different ways. For an exact diagnosis, we ask that a fault can be claimed only when a fault surely happened (as it is the case in non-probabilistic systems). However it may be necessary to weaken the correctness requirement as illustrated by the pLTS of Figure 2.6. Since all observed sequences are ambiguous no exact diagnosis can be provided. However it is clear that when in a long enough observed sequence the ratio between occurrences of 'b' and 'a' is close to 3, the probability that the corresponding run is faulty is close to 1. Let us fix any $\varepsilon > 0$ and only require that the probability for the verdict to be erroneous should be less than ε. Then using the strong law of large numbers, (approximate) fault detection is possible in this pLTS.

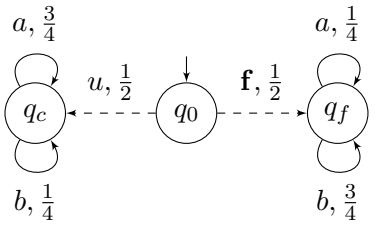


Figure 2.6: When approximate diagnosis is necessary.

To formalise later the correctness of an approximate diagnoser, with every observed sequence $w \in \Sigma_o^*$ we associate a *correctness proportion*

$$\mathsf{CorP}(w) = \frac{\mathbb{P}(\{\rho \in \mathsf{C} \cap \mathsf{SR}_{|w|} \mid \mathcal{P}(\rho) = w\})}{\mathbb{P}(\{\rho \in \mathsf{SR}_{|w|} \mid \mathcal{P}(\rho) = w\})} \ ,$$

which is the conditional probability that a signalling run is correct given that its observed sequence is w.

The standard way to specify reactivity in probabilistic systems for fault detection is to require that whatever the minimal faulty run, almost surely the diagnoser will output its (faulty) verdict. We may also consider *uniform reactivity* which strengthens reactivity by requiring that the (random) delay is independent of the minimal faulty run. More formally, uniform reactivity ensures that given any positive probability threshold $\alpha > 0$ there exists a delay n_α independent of the considered minimal faulty run such that the probability to exceed this detection delay is bounded by α.

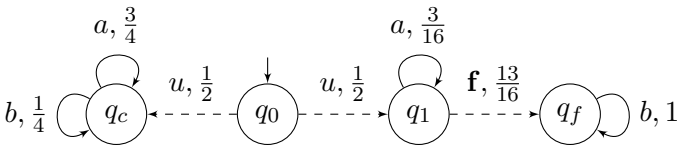


Figure 2.7: When reactivity cannot be uniform.

Let us illustrate these reactivity features with the pLTS of Figure 2.7 for which only approximate diagnosis is possible. Fix some $\varepsilon > 0$ and consider the minimal faulty run $q_0uq_1(aq_1)^m\mathbf{f}q_f$. After a certain number of occurrences of 'b' (say n), the correctness proportion of the observed sequence a^mb^n will be less than ε and thus the diagnoser can output its verdict. However due to the probabilities of an occurrence of 'a' in correct and faulty runs respectively equal to $\frac{3}{4}$ and $\frac{3}{16}$, n must depend on m and so this reactivity cannot be uniform. This can be mathematically seen through the definition of CorP: for $n \geq 1$,

$$\mathsf{CorP}(a^mb^n) = \frac{\frac{3^m}{4^{m+n}}}{\frac{3^m}{4^{m+n}} + (\frac{3}{16})^m \times \frac{15}{16}} = \frac{1}{1 + \frac{15}{16} \times 4^{n-m}}$$

In order to have $\mathsf{CorP}(a^mb^n) \leq 1/2$, one needs $n > m$. Therefore uniform diagnosis is not possible.

In order to formalise the different requirements discussed above, we first define several sets of runs related to ambiguity.

Definition 2.7 (Ambiguous runs)**.** *Let $\mathcal{A}$ be a pLTS, $\varepsilon \geq 0$ and $n \in \mathbb{N}_{>0}$. Then:*

- FAmb_∞ *is the set of infinite faulty ambiguous runs of $\mathcal{A}$;*
- CAmb_∞ *is the set of infinite correct ambiguous runs of $\mathcal{A}$;*
- FAmb_n *is the set of infinite runs of $\mathcal{A}$ whose signalling prefix of observable length n is faulty and ambiguous;*
- CAmb_n *is the set of infinite runs of $\mathcal{A}$ whose signalling prefix of observable length n is correct and ambiguous.*
- $\mathsf{FAmb}_n^\varepsilon$ *is the set of infinite faulty ambiguous runs of $\mathcal{A}$ whose observed sequence of length n, w fulfils:* $\mathsf{CorP}(w) > \varepsilon$.

By definition, for all $n \in \mathbb{N}$, $\mathsf{FAmb}_n^0 = \mathsf{FAmb}_n$. Observe that, for all $n \in \mathbb{N}$, and $\varepsilon \geq 0$, $\mathsf{CAmb}_n, \mathsf{FAmb}_n$ and $\mathsf{FAmb}_n^\varepsilon$ are open sets, thus measurable. However, CAmb_∞ and FAmb_∞ are not Borel sets in the general case (*e.g.* see Chapter 3, Section 3).

We propose five specifications of exact diagnosability for probabilistic systems based on three discriminating criteria: whether the unambiguity requirement holds for faulty runs only or for all runs, whether ambiguity is defined at the level of infinite runs or for longer and longer finite signalling prefixes, and whether the delay before detection of minimal faulty runs is uniform. These notions are summarised in Figure 2.8 except for uniformity postponed to next figure.

Definition 2.8 (Exact diagnosability)**.** *Let $\mathcal{A}$ be a pLTS.*

- $\mathcal{A}$ *is* IF-diagnosable *if* $\mathbb{P}(\mathsf{FAmb}_\infty) = 0$.
- $\mathcal{A}$ *is* IA-diagnosable *if* $\mathbb{P}(\mathsf{FAmb}_\infty \uplus \mathsf{CAmb}_\infty) = 0$.

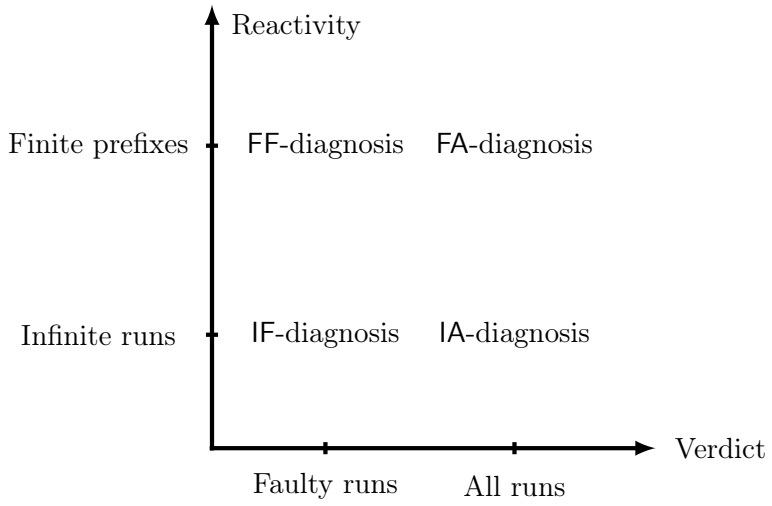


Figure 2.8: Summarising the variants of exact diagnosis.

- $\mathcal{A}$ *is* FF-diagnosable *if* $\limsup_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = 0$.
- $\mathcal{A}$ *is* FA-diagnosable *if* $\limsup_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n \uplus \mathsf{CAmb}_n) = 0$.
- $\mathcal{A}$ *is* uniformly FF-diagnosable *if for all* $\alpha > 0$ *there exists* $n_\alpha \in \mathbb{N}$ *such that for all* $n \geq n_\alpha$ *and all minimal faulty run* $\rho \in \mathsf{minF}$

$$\mathbb{P}(\{\rho' \in \mathsf{FAmb}_{n+|\rho|_o} \mid \rho \preceq \rho'\}) \leq \alpha \cdot \mathbb{P}(\rho) \ .$$

Uniform and/or approximate diagnoses are defined for FF-diagnosis as summarised in Figure 2.9. We chose FF-diagnosis as it corresponds to the classical notion of diagnosis. Moreover there is no clear intuition on what would be the meaning of uniformity and approximation for the other variants. εFF-*diagnosability* allows the diagnoser to claim a fault when the correctness proportion does not exceed ε, and accurate approximate diagnosability denoted by AFF-*diagnosability* corresponds to εFF-diagnosability for arbitrary $\varepsilon > 0$.

Definition 2.9 (Approximate diagnosability)**.** *Let* $\mathcal{A}$ *be a pLTS, and* $\varepsilon \geq 0$.

- $\mathcal{A}$ *is* εFF-diagnosable *if for every minimal faulty run* $\rho \in \mathsf{minF}$ *and all* $\alpha > 0$ *there exists* $n_{\rho,\alpha}$ *such that for all* $n \geq n_{\rho,\alpha}$*:*

$$\mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}^{\varepsilon}_{n+|\rho|_o}) \leq \alpha \cdot \mathbb{P}(\rho).$$

- $\mathcal{A}$ *is* uniformly εFF-diagnosable *if for all* $\alpha > 0$ *there exists* n_α *such that for all minimal faulty run* $\rho \in \mathsf{minF}$ *and all* $n \geq n_\alpha$*:*

$$\mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}^{\varepsilon}_{n+|\rho|_o}) \leq \alpha \cdot \mathbb{P}(\rho).$$

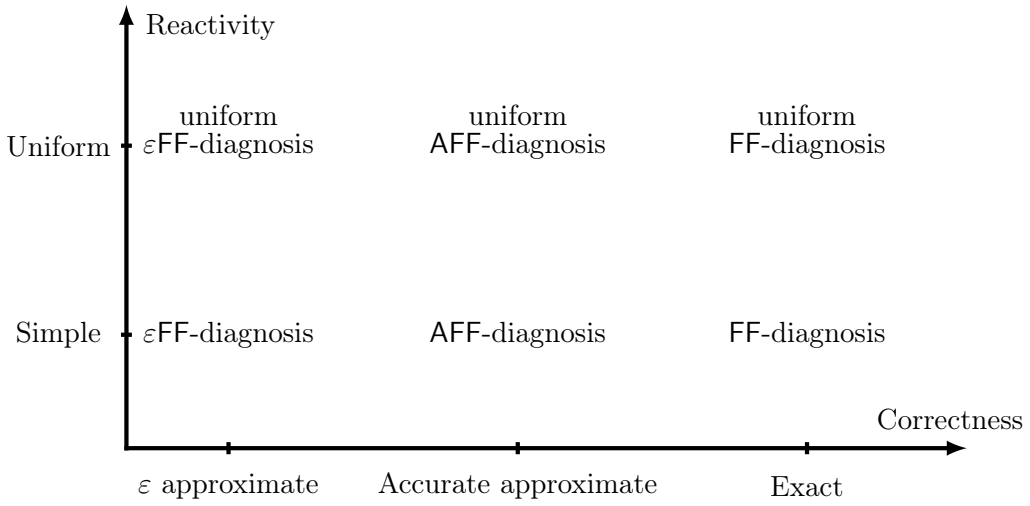


Figure 2.9: Summarising the approximate variants of FF-diagnosis.

- $\mathcal{A}$ *is (resp. uniformly)* AFF-diagnosable *if it is (resp. uniformly)* εFF*-diagnosable for all* $\varepsilon > 0$.

When studying diagnosis, we are interested in the following problems. First, for every notion of diagnosability, we want to determine if it is possible to automatically decide whether a given system is diagnosable. Moreover, in the positive case, we want to establish the exact complexity of the problem. Then, given a diagnosable system, we want to build a diagnoser satisfying the corresponding verdict, correctness and reactivity features. When possible, we represent the diagnoser using a finite-state automaton to express that only a finite memory is necessary. The problems can vary depending on the framework. For example, when the system is controllable, we want to decide if one can control the system in a way ensuring diagnosability.

2 State of the art on diagnosis

Diagnosis of finite LTS. For LTS, diagnosability requires that the occurrence of unobservable faults can be deduced after a finite delay from the sequence of observable events occurring before and after the fault [SSL+95]. Using the definitions we introduced in this chapter, an LTS is diagnosable iff $\bigcup_{n\in\mathbb{N}} \bigcap_{m\geq n} \mathsf{FAmb}_m = \emptyset$. Diagnosability of finite LTS was shown to be decidable in NLOGSPACE [JHCK01, YL02]. The algorithm relies on the twin-plant described page 29. An LTS is not diagnosable iff there exists a reachable cycle in the twin-plant in which each state is a pair composed of a faulty and a correct state. Detecting such a cycle can be done in non-deterministic logarithmic space in the twin-plant, which is quadratic in the size of the original LTS, hence the result.

Surprisingly, while deciding diagnosability is easy, the construction of the diagnoser can require exponential time. This construction is based on the belief construction quickly presented page 29.

Diagnosis of infinite LTS. An LTS with infinitely many states must be represented by a higher level model in order to be studied. This can be done using Petri nets (PN) for example. The semantics of a PN is called the reachability graph and can be represented by an LTS. This LTS is finite iff the PN is bounded. Cabasino et al. studied diagnosis of both bounded and unbounded PN [CGLS12]. In order to solve diagnosability in the unbounded case they first build the verifier net, which is a construction similar to the twin-plant, then construct the coverability graph (a finite abstraction of the reachability graph) of the verifier net and finally analyse the cycles of the coverability graph. This final analysis requires some additional properties of the cycles compared to the analysis in the twin-plant for finite LTS. This algorithm however has a complexity that depends on the size of the coverability graph, which may be Ackermanian in the size of the description of the PN. An algorithm with better complexity was developed in [BHSS18]. It still uses the verifier net, but transforms the diagnosability problem into an LTL formula and uses model-checking results to obtain an EXPSPACE upper bound. Interestingly, this paper also studies opacity, confirming the intuition that opacity and diagnosis are two close problems. See [Bas14] for a presentation of the usual techniques used for fault diagnosis in PN. LTS with infinitely many states can also be represented by pushdown systems. Morvan and Pinchinat showed that diagnosability in the general case is undecidable [MP09]. However, for a large subclass called visibly pushdown automata, diagnosability can be decided in PTIME. This is done by building once again a form of twin-plant, this time making the product of the visibly pushdown automata. Thanks to this product, they can define an appropriate Büchi condition on the twin-plant so that diagnosability can be deduced from the emptiness of the obtained Büchi pushdown automaton (checking the emptiness can be done in PTIME). The restriction to visibly pushdown automata, which we discuss in Chapter 5, is required in order to build the twin-plant.

Diagnosis of stochastic systems. When diagnosability was adapted to stochastic systems by Thorsley and Teneketzis in [TT05], two notions of diagnosability were initially defined: A-diagnosability and AA-diagnosability. In finite pLTS, A-diagnosability corresponds to our uniform FF-diagnosability and AA-diagnosability corresponds to our uniform AFF-diagnosability. For A-diagnosability, they gave a necessary and sufficient condition based on the belief construction: they first build a diagnoser similarly to what was done for logical systems, then test for the recurrence of ambiguous states of the belief (*i.e.* states that can contain both correct and faulty states). The complexity of the algorithm checking this characterisation is not mentioned however. For AA-diagnosability, they only give a sufficient condition, leaving the general problem open. The questions left opened by [TT05] were tackled by Chen and Kumar who gave algorithms with PTIME complexities to answer both diagnosability decision problems [CK13]. Their algorithm for AA-diagnosability is particularly interesting as they

translate the problem into a question of language equivalence (in terms of probability of words) for the original pLTS in a specific initial distribution. Unfortunately, these two algorithms are erroneous (see Chapter 4 for details on these two problems).

When a pLTS is not diagnosable, one interesting question that can be raised is how far from diagnosability it is. If there is only a very little chance that the fault is not detected, the system may be "diagnosable enough". This direction was studied in [ND08] where Nouioua and Dague consider an exact notion of diagnosability and wish to measure the probability of the faulty ambiguous runs. To realise this measure, they make the product of the pLTS with its belief construction (thus a method that we already presented page 29). Then they measure the asymptotic probability to be in each state of this product pLTS. As, thanks to the belief, states contain the relevant information to determine if they were reached by a faulty ambiguous run, they can determine the probability to be faulty and ambiguous at the limit. This approach is continued in [BFG17] where Bazille et al. introduce a notion of k-diagnosability degree, which is defined as the probability to detect a fault at most k steps after it occurs, conditioned to the occurrence of a fault. They measure this degree by (1) building the product of the pLTS with the belief construction and (2) using polynomial time algorithms that compute the sum of the probabilities of the runs that reach a target state set (in this case, the states which belief component show they were reached by faulty ambiguous runs). Using the computation of k-diagnosability degrees for different values of k, they also investigate the average speed of detection of a fault.

Active diagnosis. One can enrich an LTS by allowing a form of control. This is done by introducing non-determinism, that is resolved at every step by a controller. One possibility of control is done through restriction of the enabled events: at every step, the controller selects a set of observable events $\Sigma^\bullet \subseteq \Sigma_o$ and the next transition taken by the LTS is labelled either by an unobservable event or by an observable one which observation belongs to $\Sigma^\bullet$. Some observable events can also be considered uncontrollable and are enabled no matter the choice of the controller. In this framework, the diagnosability of a system depends on the choice of the controller. Finding a controller such that the controlled system is diagnosable is the goal of *active diagnosis*. This problem was introduced in [SLT98]. Sampath, Lafortune and Teneketzis then solve the question by building the most permissive controller through a complicated iterative procedure which complexity is not given (and seems to be doubly exponential). Later, a planning-based approach via a twin-plant construction was proposed in [CP09]. The exact complexities were finally established in [HHMS13, HHMS17], where the active diagnosis problem was shown to be EXPTIME-complete and finite-memory controllers (most permissive and optimal in memory size) are given. This is done by translating the active diagnosis problem into a Büchi game (using a variant of the product with the belief) and then solving the Büchi game, which gives an optimal strategy which can be translated back into a controller. This analysis was extended to controllable stochastic systems [BFH$^+$14]. Here, instead of translating the problem to a Büchi game, Bertrand et al. use partially observable Markov decision processes and show that the problem of stochastic active diagnosis is also EXPTIME-complete. They also study a safe notion of

stochastic active diagnosis where the controller is required to keep a positive probability of infinite correct runs. This second problem is then showed to be undecidable in the general case and EXPTIME-complete when limited to finite-memory strategies.

The control on the system is not necessarily related to the events, it can also be applied on the observations. The observations of a system are given by sensors. In order to detect an event, one needs to have a sensor at the appropriate position and for it to be switched on. In [CT08] and [TT07] the authors investigate in slightly different frameworks, how to limit the number of sensors needed and how to build a controller which chooses at every step which sensor is switched on or off. The main differences between the two papers are twofold: (1) [TT07] considers both logical and stochastic systems while [CT08] only focuses on logical systems and (2) [CT08] establishes that a most permissive finite-state controller can be computed in doubly exponential time, using a game-theoretic approach while [TT07] does not give the exact complexity of their algorithm.

See [ZL13] for a survey on diagnosis mainly describing results for logical systems, but discussing also timed, stochastic and active systems.

Part II

Analysing information in passive systems

Chapter 3

Semantical analysis of diagnosability

As explained in Chapter 2, diagnosers observe the system to determine if its behaviour is correct or not. They are formalised as functions giving a verdict to each sequence of observation produced by a run of the system. There exist many different ways to define a diagnoser, depending on the properties a system designer may want. We identified as the most important features of a diagnoser its verdict, correctness and reactivity.

- The *verdict* determines what information is given by the diagnoser. Consequently, changing the verdict literally means modifying the purpose of the diagnoser. For example, when testing a car, if a component malfunctions, the company needs to detect it. The diagnoser thus only needs to detect the faulty behaviour of the system. However, the verdict of a diagnoser could also require to detect the correct behaviour of a system. For example, when following online a critical system like a power plant, the technicians needs to know that the system is correct. If the system can be faulty, they may need to shut it down in order not to take any risk, which can have an important cost.

- The *correctness* determines if the diagnoser is allowed to make an erroneous claim and, if so, how accurate must its claim be. It is of course better to have a diagnoser that does not make any error, but this restriction is not always realistic. Consider a program simulating dice throws. If the program outputs 4 a high number of times in a row, this may be a correct behaviour as such a throw has a positive probability. However, the longer this streak of 4 continues, the more likely it is caused by a malfunction of the program.

- The *reactivity* determines how quickly and how often the diagnoser must output a verdict. One possibility would be to require that if a fault occurs, after a bounded delay this fault will be detected. This is the reactivity that is often required in non-probabilistic systems represented by an automaton [SSL+95]. This requirement is too strong however for probabilistic systems. Indeed, in probabilistic systems, one can have for example an event which can occur at every step after a fault

with some probability, say 1/2, and which allows the detection of the fault. In expectation, the fault is thus detected after two steps, but no bound can be given on the maximum delay. This is a situation occurring in every system where, after a fault, the system can, with some probability, continue to act normally. When studying probabilistic systems, the reactivity requirement must thus be adapted. This can be done, for example, by requiring that with probability 1 the fault will be detected. This version of reactivity allows to ignore runs that have a zero probability of occurring.

The choice of the verdict, correctness and reactivity notions of a diagnoser thus depends on what information the designer of the system desires and on which guarantees are demanded. The decisions that are taken also affect the complexity of determining the diagnosability and of building the diagnoser. They thus also affect the capacities required from the computer that will carry out the diagnosis. Studying as many relevant notions as possible and clearly establishing their complexity is thus necessary for an appropriate application of diagnosis.

In Chapter 2, multiple notions of diagnosability were defined for probabilistic systems based on the ambiguity of specific sets of runs. This allowed us to establish diagnosability analysis as a decision problem which is easier to use in proofs and more intuitive. However, the end goal of analysing the diagnosability of a system being to build a diagnoser, one could also use the following definition: given notions of verdict, correctness and reactivity, a system is called diagnosable if there exists a diagnoser of the system achieving these features. We will show in Section 1 how to reconcile these two approaches: we will associate a diagnoser with each diagnosability notion previously defined. Having defined these associated diagnosers will allow us to see how the notions of diagnosability translate on an actual run of the system.

Once the notions of diagnoser have been clearly defined, we must determine how the different variants of diagnosability relate to one another. Establishing links between multiple notions is a classical part of the semantical analysis of a problem. Through this analysis, one can establish that two definitions that are syntactically different, are in fact equivalent. In this case, one can choose to use either definition in a proof for example. Implications between two notions are also useful. If a system verifies a stronger property, we will not have to check for the weaker ones. For diagnosability, it allows the system designer to select the strongest available diagnoser without having to check every notion. Or at least one of the strongest notions as some of them may be incomparable. Therefore, establishing the links between the notions of diagnosability defined previously will be the focus of Section 2

The last goal of this chapter, presented in Section 3 will be to give, when possible, a characterisation of the notions of diagnosability. As we wish to have the simplest possible characterisation, an important question is to determine what information is needed to characterise diagnosability. For example, can we restrict ourselves to studying the structure of the system or do probabilities matter? And if they do, in what way? In fact, in the general case, a characterisation relies both on the structure of the system and on its probabilities.

- The structural part of the characterisation is based on descriptive set theory recalled in Chapter 2, Section 1.1. We will define a logic generating sets of runs belonging to a low level of the Borel hierarchy and associate, when possible, a formula with every notion of diagnosability.
- The probabilistic part then consists in measuring the probability of the set of runs identified above. The system is then diagnosable if and only if this measure verifies a qualitative requirement.

Having such a characterisation has multiple advantages. For example, using model-checking techniques, one could use these characterisations to solve diagnosability as we explain in Chapter 5. However this is not necessarily the optimal method as shown in Chapter 4. As another example, each Borel set is associated with a level of the Borel hierarchy. The higher this level, the more complicated the set is. This complexity reflects a complexity to measure the probability of the set but also a complexity of understanding the meaning behind this set. Having a characterisation with a set belonging to a low level of the Borel hierarchy thus makes it easier for the system designer to understand the associated diagnosability notion.

This chapter presents and extends some of the results given in [BHL14, BHL16a, BHL16b].

1 Diagnoser and diagnosability

In this section we focus on the synthesis of diagnosers for the notions of diagnosis defined in Chapter 2. Recall the definition of diagnosers:

Definition 3.1. *A* diagnoser *is a function* $D : \Sigma_o^* \to \{?, \top, \bot\}$ *assigning to every finite observed sequence a verdict.*

Multiple verdicts can be required for the diagnosers. Intuitively, ? does not provide any information, $\top$ claims the occurrence of a fault and $\bot$ provides information about the correctness of the current run.

Diagnosability as defined in Chapter 2 considers infinite behaviours: either by focusing on the ambiguity of infinite runs, or by requiring that the probability of a set of finite runs converge to 0 when their length diverges to infinity. On the contrary, diagnosers are built to react and give an information after a finite number of observations. There is therefore no easy direct link between diagnosability and diagnosers.

Due to its definition, a diagnoser may use infinite memory or more precisely, unbounded memory. While infinite memory is not achievable in real systems, unbounded memory is. It however raises a question on how to implement this memory and how much information has to be kept by the diagnoser. For example, if a diagnoser only needs to remember how many observations occurred, it may rely on a counter which, although unbounded, is easy to represent and to modify. When implementing a diagnoser (which will be done in Chapter 4), it is still natural to limit oneself to finite memory. We therefore define now the notion of finite-memory diagnosers.

Definition 3.2. *A* finite-memory diagnoser *is given by a tuple* $(M, \Sigma_o, m_0, \mathsf{up}, D_{fm})$ *where:*

- M *is a finite set of memory states,*
- $m_0 \in M$ *is the initial memory state,*
- $\mathsf{up} : M \times \Sigma_o \to M$ *is a memory update function,*
- $D_{fm} : M \to \{?, \top, \bot\}$ *is a diagnoser function.*

A finite-memory diagnoser $(M, \Sigma_o, m_0, \mathsf{up}, D_{fm})$ can be seen as a deterministic automaton over Σ_o where the set of states is M, the initial state is m_0 and the transition function is up. Moreover the states of this automaton are labelled by an element of $\{?, \top, \bot\}$ which is given by the function D_{fm}. The update up is extended into a function $\mathsf{up} : M \times \Sigma_o^* \to M$ defined inductively by $\mathsf{up}(m, \varepsilon) = m$ and $\mathsf{up}(m, wa) = \mathsf{up}(\mathsf{up}(m, w), a)$. The *size* of a finite-memory diagnoser is given by its number of memory states. A finite-memory diagnoser is not a diagnoser as defined in Chapter 2 and recalled in Definition 3.1, yet it induces the diagnoser D defined by $D(w) = D_{fm}(\mathsf{up}(m_0, w))$.

Example 3.1. *Consider the finite-memory diagnoser* $(M, \{a, b\}, m_0, \mathsf{up}, D_{fm})$ *(represented in Figure 3.1) where*

- $M = \{m_0, m_b\}$,
- $\mathsf{up}(m_0, a) = m_0$, $\mathsf{up}(m_0, b) = \mathsf{up}(m_b, a) = \mathsf{up}(m_b, b) = m_b$,
- $D_{fm}(m_0) = ?$ *and* $D_{fm}(m_b) = \top$.

It induces a diagnoser D which makes no claim as long as it only observes 'a' and claims a fault as soon as a 'b' is observed. It then commits to this choice and keeps claiming a fault whatever is observed next.

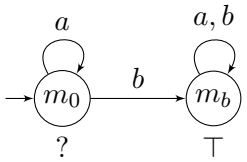


Figure 3.1: The finite-memory diagnoser of Example 3.1. The verdict given by D_{fm} in a memory state is written below the state.

Each following subsection focuses on one notion of diagnosability. They are ordered from the easiest to the most difficult definition of diagnoser. Due to the relations that will be established in Section 2, we won't detail every notion of diagnosability here.

1.1 FF-diagnosers

We will start by defining the FF-diagnosers of a pLTS, which is the diagnoser associated with FF-diagnosability. Recall that FF-diagnosability requires the probability of the set of faulty ambiguous finite runs to converges to 0 (*i.e.* $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = 0$). These diagnosers only provide information about faulty runs, they therefore never raise a $\bot$ verdict. However they need to raise a $\top$ verdict almost surely after a finite delay in a faulty run. We can thus restrict their verdict to the set $\{?, \top\}$. We require from diagnosers associated with exact diagnosability notions that they satisfy an additional property,*commitment*, which means that when it claims a fault it will persistently claim it in the future. This can be done thanks to the permanence of the fault, *i.e.* a faulty run will remain faulty.

Definition 3.3. *An* FF-diagnoser *for a pLTS $\mathcal{A}$ is a function $D : \Sigma_o^* \to \{\top, ?\}$ such that:*

commitment *For every $w \preceq w' \in \Sigma_o^*$, if $D(w) = \top$ then $D(w') = \top$.*

correctness *For every $w \in \Sigma_o^*$, if $D(w) = \top$ then w is surely faulty.*

reactivity *For every $\rho \in \mathsf{minF}$, $\mathbb{P}(\{\rho' \in \Omega \mid \rho \preceq \rho' \wedge D(\mathcal{P}(\rho')) =?\}) = 0$ where for $w \in \Sigma_o^\omega$, $D(w) = \lim_{n\to\infty} D(w_{\leq n})$.*

Let us comment on this definition. The commitment property ensures that if the diagnoser outputs $\top$ at some point it will always output $\top$. The correctness property forbids the diagnoser to claim a fault during the observation of a correct run. This reflects that FF-diagnosability is a notion of *exact* diagnosability. Thus the FF-diagnoser is *exact* too. The limit in the reactivity condition of the above definition is well defined. Indeed, note that if $\top$ is produced, due to the commitment property, the limit is $\top$. Otherwise the diagnoser always outputs ? so that the limit is also a ? verdict.

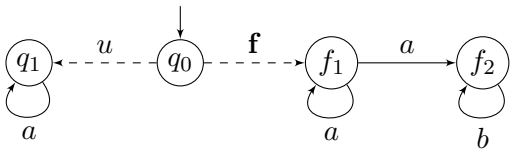


Figure 3.2: An FF-diagnosable pLTS.

Example 3.2. *For the pLTS of Figure 3.2, we define the diagnoser D such that $D(\varepsilon) =$?, for all $n \in \mathbb{N}, D(a^n) =?$ and for every word $w \notin a^*, D(w) = \top$. This diagnoser is in fact the one induced by the finite-memory diagnoser of Example 3.1. Such a diagnoser is indeed an* FF*-diagnoser:*

commitment *Once a 'b' is observed, every subsequent observation is also a 'b', therefore once D produces $\top$, it will keep outputting $\top$.*

correctness *If a $\top$ has been produced, b was observed. The only transition labelled by a 'b' is the self-loop on f_2 which can only be reached by faulty runs. Therefore any observed sequence for which D outputs $\top$ is surely faulty.*

reactivity *Let ρ be a minimal faulty run. It ends either in f_1 or in f_2. If it ends in f_1, then with probability 1 a run extending ρ takes the transition to f_2. In f_2, it will read a 'b' in the next step. In other words, with probability 1, a faulty run will trigger a 'b' and thus $\top$ is raised by the diagnoser. Due to the commitment property, given a faulty run ρ, the probability that the diagnoser outputs ? infinitely often during the observation of a run extending ρ is thus 0.*

Observe also that this pLTS is indeed FF*-diagnosable as for $n \geq 1$, we have*

$$\mathbb{P}(\mathsf{FAmb}_n) = \mathbb{P}(\{\rho \in \mathsf{F}_n \mid \mathcal{P}(\rho_{\downarrow n}) = a^n\}) = \frac{1}{2^n}.$$

As suggested by the above example, there exists an FF-diagnoser if and only if the system is FF-diagnosable. We now prove this formally.

Proposition 3.1. *A pLTS is* FF*-diagnosable if and only if it admits an* FF*-diagnoser.*

This proof is done in the following way. Assuming there exists an FF-diagnoser we study a family of sets of faulty runs FD_n that corresponds to runs where the fault is claimed after n observations. We compute the probability of this set and link this value to the probability of FAmb_n. This then allow us to show that the probability of FAmb_n converges to 0, proving thus the FF-diagnosability of the pLTS. Assuming the pLTS is FF-diagnosable, we present a diagnoser and then show it is an FF-diagnoser by proving the properties one by one.

Proof. Let $\mathcal{A}$ be a pLTS, and assume there exists an FF-diagnoser D for $\mathcal{A}$. For every $n \in \mathbb{N}$, we define $\mathsf{FD}_n = \{\rho \in \mathsf{F}_\infty \mid D(\mathcal{P}(\rho_{\downarrow n})) = \top\}$ the set of faulty runs that are diagnosed faulty after n observed events. We will start by showing that $\lim_{n\to\infty} \mathbb{P}(\mathsf{FD}_n) = \mathbb{P}(\mathsf{F}_\infty)$. As a consequence of the commitment property, the sequence $(\mathsf{FD}_n)_{n\in\mathbb{N}}$ is non-decreasing and for every faulty run $\rho \in \mathsf{F}_\infty$, $D(\mathcal{P}(\rho)) = \lim_{n\to\infty} D(\mathcal{P}(\rho_{\downarrow n})) =?$ is equivalent to $\rho \notin \bigcup_{n\in\mathbb{N}}(\mathsf{FD}_n)$, *i.e.* $\lim_{n\to\infty} \mathsf{FD}_n = \bigcup_{n\in\mathbb{N}}(\mathsf{FD}_n) = \{\rho \in \mathsf{F}_\infty \mid D(\mathcal{P}(\rho)) \neq ?\}$. Since D is reactive, for every minimal faulty run $\rho \in \mathsf{minF}$, we have $\mathbb{P}(\{\rho' \in \Omega \mid \rho \preceq \rho' \wedge D(\mathcal{P}(\rho')) = ?\}) = 0$. As every faulty run is prefixed by an unique minimal faulty run, we have

$$\mathbb{P}(\{\rho' \in \mathsf{F}_\infty \mid D(\mathcal{P}(\rho')) =?\}) = \sum_{\rho\in\mathsf{minF}} \mathbb{P}(\{\rho' \in \Omega \mid \rho \preceq \rho' \wedge D(\mathcal{P}(\rho')) =?\}) = 0.$$

Thus,

$$\begin{aligned}\mathbb{P}(\bigcup_{n\in\mathbb{N}}(\mathsf{FD}_n)) &= \mathbb{P}(\{\rho' \in \mathsf{F}_\infty \mid D(\mathcal{P}(\rho')) \neq ?\}) \\ &= \mathbb{P}(\mathsf{F}_\infty) - \mathbb{P}(\{\rho' \in \mathsf{F}_\infty \mid D(\mathcal{P}(\rho')) =?\}) \\ &= \mathbb{P}(\mathsf{F}_\infty).\end{aligned}$$

Moreover since D is correct, for every $n \in \mathbb{N}$, $\mathsf{FD}_n \subseteq \mathsf{Sf}_n$. Therefore, for every $n \in \mathbb{N}$, $\mathbb{P}(\mathsf{FAmb}_n) = \mathbb{P}(\mathsf{F}_n) - \mathbb{P}(\mathsf{Sf}_n) \leq \mathbb{P}(\mathsf{F}_n) - \mathbb{P}(\mathsf{FD}_n)$ and

$$\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) \leq \lim_{n\to\infty} \mathbb{P}(\mathsf{F}_n) - \mathbb{P}(\mathsf{FD}_n) = 0.$$

This shows that $\mathcal{A}$ is FF-diagnosable.

Assume now that $\mathcal{A}$ is FF-diagnosable. We define the function $D : \Sigma_o^* \to \{\top, ?\}$ by $D(w) = \top$ if and only if w is a surely faulty observed sequence. Let us check that D is an FF-diagnoser. As a surely faulty ambiguous sequence cannot become ambiguous again, D fulfils the commitment property. Moreover, since $D(w) = \top$ iff w is a surely faulty sequence, D is correct. Now, let ρ be a minimal faulty run.

$$\mathbb{P}(\{\rho' \in \Omega \mid \rho \preceq \rho' \wedge D(\mathcal{P}(\rho')) =?\}) = \lim_{n\to\infty} \mathbb{P}(\{\rho' \in \mathsf{FAmb}_{n+|\rho|_o} \mid \rho \preceq \rho'\}) \ .$$

For every $n \in \mathbb{N}$, we have $\{\rho' \in \mathsf{FAmb}_{n+|\rho|_o} \mid \rho \preceq \rho'\} \subseteq \mathsf{FAmb}_{n+|\rho|_o}$ and, as $\mathcal{A}$ is FF-diagnosable, $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = 0$. Therefore $\mathbb{P}(\{\rho' \in \Omega \mid \rho \preceq \rho' \wedge D(\mathcal{P}(\rho')) =?\}) = 0$ and D is reactive. □

The notion of FF-diagnoser we defined is therefore appropriate for FF-diagnosability.

1.2 FA-diagnosers

FA-diagnosability and IA-diagnosability not only consider the diagnosis of faults but also of correct runs. Indeed, recall that they require respectively that the probability of the set of ambiguous finite runs converges to 0 (*i.e.* $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n \cup \mathsf{CAmb}_n) = 0$) and that the probability of infinite ambiguous runs is equal to 0 (*i.e.* $\mathbb{P}(\mathsf{FAmb}_\infty \cup \mathsf{CAmb}_\infty) = 0$). Contrary to FF-diagnosers, FA- and IA-diagnosers have three possible verdicts: $\top$, related to faulty sequences, $\bot$, linked with correctness, and ? when no information can be derived from the observation. We consider the partial order $\prec$ on these values defined by $? \prec \top$ and $? \prec \bot$. This order is natural as ? gives less information than the other verdicts. Although we consider the detection of correct and faulty runs, note that the situation is not symmetric given that the faults are persistent while correct runs may become faulty.

Definition 3.4. *An* FA-diagnoser *for a pLTS $\mathcal{A}$ is a function $D : \Sigma_o^* \to \{\top, \bot, ?\}$ such that:*

commitment *For every $w \preceq w' \in \Sigma_o^*$, if $D(w) = \top$ then $D(w') = \top$.*

correctness *For every $w \in \Sigma_o^*$,*

- *if $D(w) = \top$ then w is surely faulty;*
- *if $D(w) = \bot$ then w is surely correct.*

reactivity $\mathbb{P}(\{\rho \in \Omega \mid D(\mathcal{P}(\rho)) =?\}) = 0$ *where for $w \in \Sigma_o^\omega$, $D(w) = \liminf_{n\to\infty} D(w_{\leq n})$.*

Let us comment on this definition. The commitment property is similar to the one of the notion of FF-diagnoser. In particular, there is no commitment for a $\bot$ verdict. This is natural as a fault could appear later, forcing the diagnoser to change its verdict. The correctness property is also similar to the FF-diagnoser for a $\top$ verdict. For the $\bot$ verdict, it is the dual, the diagnoser cannot output $\bot$ while observing a faulty run. This diagnoser is 'exact'. The limit in the reactivity requires the partial order. While if a $\top$ is claimed, the limit will be $\top$, a $\bot$ verdict can be followed by a ? verdict. Due to the correctness property, the limit is equal to $\bot$ if the observed sequence is, after a finite number of observations, always surely correct.

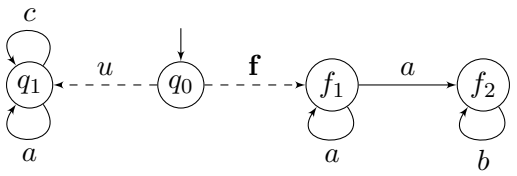


Figure 3.3: An FA-diagnosable pLTS.

Example 3.3. *For the pLTS of Figure 3.3, we define the diagnoser D such that $D(\varepsilon) = \bot$ and given an observed sequence $w \in \Sigma_o^*$, then $D(wb) = \top$, $D(wc) = \bot$ and $D(wa) = D(w)$. Remarking that once a b or a c is observed, the other cannot appear any more, this diagnoser is induced by the finite-memory diagnoser of Figure 3.4. Such a diagnoser is indeed an* FA*-diagnoser:*

commitment *Once a 'b' is observed, every subsequent observation is also a 'b', therefore once D produces $\top$, it will keep outputting $\top$.*

correctness *If a $\top$ has been produced, 'b' was observed previously. The only transition labelled by a 'b' is the self-loop on f_2 which can only be reached by faulty runs. Therefore any observed sequence for which D outputs a $\top$ is surely faulty. Similarly, when D outputs $\bot$ it means that a 'c' was observed previously and such an observation can only be made in q_1 which is a correct state from which no fault can ever be triggered. Thus the observed sequence is surely correct.*

reactivity *The set of infinite observed sequences w for which $D(w) =?$ is restricted to $\{a^\omega\}$. There exist exactly two runs corresponding to this observed sequence $\rho_1 = q_0u(q_1a)^\omega$ and $\rho_2 = q_0\mathbf{f}(f_1a)^\omega$. Moreover, the probability of each of these two runs is 0 as firing 'a' in q_1 or in f_1 only has probability $1/2$. Thus with probability 1, a 'b' or a 'c' will be observed, ensuring the reactivity of D.*

Observe also that this pLTS is indeed FA*-diagnosable as for every $n \geq 1$, we have $\mathsf{FAmb}_n = \{q_0u(q_1a)^nq_1, q_0\mathbf{f}(f_1a)^nf_1, q_0\mathbf{f}(f_1a)^{n-1}f_1af_2\}$. Thus $\mathbb{P}(\mathsf{FAmb}_n) = \frac{3}{2^{n+1}}$.*

We now want to establish the link between the existence of an FA-diagnoser and FA-diagnosability. However, there is no equivalence in the general case. Indeed, let us

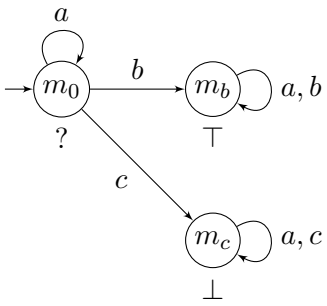


Figure 3.4: The finite-memory diagnoser of Example 3.3.

consider the pLTS of Figure 3.5. The probability of the set of ambiguous runs of length $n \geq 1$ starting by u_1 is the probability to have read a 'b' on the $n-1$'th observation. This is equal to $\frac{1}{n+1}$. Moreover, runs initially starting by u_2 will almost surely trigger a 'c', removing the ambiguity. Thus it is FA-diagnosable. However, a run starting with u_1 will almost surely trigger infinitely many 'b's. Because of the correctness property, we have for every diagnoser D, $\mathbb{P}(\{\rho \in \Omega \mid D(\mathcal{P}(\rho)) =?\}) \geq \mathbb{P}(\{\rho \in \Omega \mid q_0u_1q_1 \preceq \rho\}) = \frac{1}{2}$. Thus, D is not reactive.

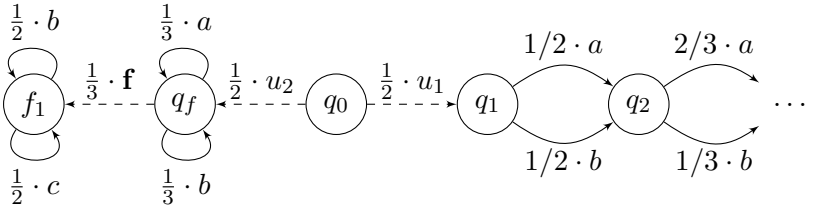


Figure 3.5: An infinite FApLTS which does not accept any FA-diagnoser.

Fortunately, the equivalence holds when restricted to finite pLTS. We postpone the proof of the following proposition to Chapter 4 (more precisely to Proposition 4.14, page 126) which focuses on finite pLTS.

Proposition 3.2. *A finite pLTS $\mathcal{A}$ is* FA*-diagnosable if and only if it admits an* FA*-diagnoser.*

As a conclusion, the definition of FA-diagnosers is similar to the one of FF-diagnosers, but with additional requirements to deal with the correct ambiguous runs. This fits the definition of FA-diagnosability which becomes equivalent to the one of FF-diagnosability when the set of correct ambiguous runs can be neglected. However, due to the complexity created by the fact that being correct is not a permanent status of runs (contrary to being faulty), the link between existence of an FA-diagnoser and FA-diagnosability cannot be established in the general case.

1.3 IA-diagnosers

The problem in defining IA-diagnosers is that IA-diagnosability defines ambiguity for infinite runs while a diagnoser must give its verdict after a finite observation. As a consequence, the information an IA-diagnoser gives after a finite run is weaker than the information an FA-diagnoser has to give.

Definition 3.5. *An* IA-diagnoser *for $\mathcal{A}$ is a function $D : \Sigma_o^* \to \{\top, \bot, ?\}$ such that:*

commitment *For every $w \preceq w' \in \Sigma_o^*$, if $D(w) = \top$ then $D(w') = \top$.*

correctness *For every $w \in \Sigma_o^*$*

- *if $D(w) = \top$, then w is surely faulty;*
- *if $D(w) = \bot$, letting $|D(w)|_\bot = |\{0 < n \leq |w| \mid D(w_{\leq n}) = \bot\}|$, then for every signalling run ρ such that $\mathcal{P}(\rho) = w$, $\rho_{\downarrow |D(w)|_\bot}$ is correct.*

reactivity $\mathbb{P}(\{\rho \in \Omega \mid D(\mathcal{P}(\rho)) = ?\}) = 0$ *where for any observed sequence $w \in \Sigma_o^\omega$, $D(w) = \limsup_{n\to\infty} D(w_{\leq n})$.*

Let us comment on the definition. Commitment is once again focused only on the $\top$ verdict. Correctness is usual for $\top$ but quite different for $\bot$. Indeed, the correctness of FA-diagnosers requires that a $\bot$ verdict means the observed sequence is surely correct. The interpretation of $D(w) = \bot$ for IA-diagnoser is that the diagnoser ensures that any signalling subrun of length $|D(w)|_\bot \leq |w|$ of a signalling run for w is correct. Of course it may deduce this information from the last $|w| - |D(w)|_\bot$ observations. This does not reveal if the current run is correct or not. However, if the diagnoser outputs $\bot$ infinitely often along an observed sequence w, $\lim_{n\to\infty} |D(w_{\leq n})|_\bot = \infty$. Therefore the infinite observed sequence w is surely correct. The reactivity condition uses a limit superior as we only need $\bot$ to be claimed infinitely often but not necessarily without ? in between.

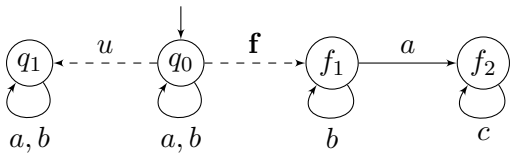


Figure 3.6: An IA-diagnosable pLTS.

Example 3.4. *For the pLTS of Figure 3.6 we define the diagnoser D by: for any observed sequence w if there exists an observed sequence w' such that $w = w'c$ then $D(w) = \top$, if $w \in \{w'aa, w'ab\}$, $D(w) = \bot$ else $D(w) = ?$. This diagnoser is induced by the finite-memory diagnoser represented in Figure 3.7 Such a diagnoser is indeed an* IA*-diagnoser:*

commitment *Once a 'c' is observed, only a 'c' can appear, thus the diagnoser keep the $\top$ verdict.*

correctness *Observing a 'c' can only be made for faulty run, thus D correctly raises a $\top$. For the correct runs the idea is the following. After observing any sequence waa or wab, with $w \in \{a, b\}^*$, the diagnoser knows a posteriori that one step before, that is after the observation of wa, the run was necessarily correct. Indeed, observing the suffix aa is not possible after a fault, yet wba is not surely correct. After a run ρ with such an observation we thus know that $\rho_{\downarrow|\mathcal{P}(\rho)|-1}$ is correct and $|D(w)|_\perp$ is at most equal to $|\mathcal{P}(\rho)| - 2$ as no $\perp$ can be claimed after 0 or 1 observation. Thus D is correct.*

reactivity *With probability 1 a faulty run will trigger a 'c' (raising a $\top$ verdict) and a correct run will trigger infinitely many 'a's (raising infinitely many $\perp$ verdicts).*

Moreover this pLTS is IA*-diagnosable as* $\mathsf{CAmb}_\infty \cup \mathsf{FAmb}_\infty = \mathcal{P}^{-1}(\{a, b\}^* b^\omega)$ *which has probability 0 as in every state there is a positive probability to trigger an action whose observation is different than b.*

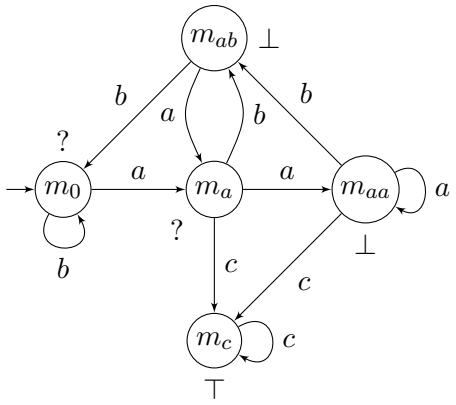


Figure 3.7: The finite-memory diagnoser of Example 3.3.

We now aim at establishing the link between IA-diagnosability and the existence of an IA-diagnoser. Since IA-diagnosability gives an information about infinite runs, we need a way to translate it to finite runs in order to establish the link with the diagnoser. Hence, we first introduce a lemma linking the sets FAmb_n and FAmb_∞ for $n \in \mathbb{N}$. This lemma will be reused in the next section when establishing the link between FF-diagnosability and IF-diagnosability.

Lemma 3.1. *Let $\mathcal{A}$ be a pLTS. Then* $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_\infty \setminus \mathsf{FAmb}_n) = 0$. *Moreover, if $\mathcal{A}$ is finitely branching, then* $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n \setminus \mathsf{FAmb}_\infty) = 0$.

The main difficulty of the following proof is in the second point. There, using the finitely-branching assumption and invoking König's lemma, we show that if the prefixes of a finite run are never surely faulty, then we can build an infinite correct run with the same observed sequence.

Proof. Observe that Ω admits the following partitions $\Omega = \mathsf{FAmb}_\infty \uplus \mathsf{C}_\infty \uplus \mathsf{Sf}_\infty$ and for all $n \in \mathbb{N}$, $\Omega = \mathsf{FAmb}_n \uplus \mathsf{C}_n \uplus \mathsf{Sf}_n$. Thus, for all $n \in \mathbb{N}$,

$$\begin{aligned}\mathsf{FAmb}_\infty \setminus \mathsf{FAmb}_n &= (\mathsf{C}_n \uplus \mathsf{Sf}_n) \cap \mathsf{FAmb}_\infty \\ &= (\mathsf{C}_n \uplus \mathsf{Sf}_n) \setminus (\mathsf{C}_\infty \uplus \mathsf{Sf}_\infty) \subseteq (\mathsf{C}_n \setminus \mathsf{C}_\infty) \uplus (\mathsf{Sf}_n \setminus \mathsf{Sf}_\infty).\end{aligned}$$

Since for all n, $\mathsf{Sf}_n \subseteq \mathsf{Sf}_\infty$, one gets:

$$\mathsf{FAmb}_\infty \setminus \mathsf{FAmb}_n \subseteq \mathsf{C}_n \setminus \mathsf{C}_\infty \ .$$

$\{\mathsf{C}_n\}_{n\in\mathbb{N}}$ is a non-increasing family of sets and we claim that $\mathsf{C}_\infty = \bigcap_{n\in\mathbb{N}} \mathsf{C}_n$. Indeed an infinite run ρ is correct if and only if $\mathbf{f}$ does not occur in it *i.e.* if and only if all its signalling subruns are correct. Thus, $\lim_{n\to\infty} \mathbb{P}(\mathsf{C}_n \setminus \mathsf{C}_\infty) = 0$ which implies $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_\infty \setminus \mathsf{FAmb}_n) = 0$.
For the other direction, using again the two partitions we obtain:

$$\begin{aligned}\mathsf{FAmb}_n \setminus \mathsf{FAmb}_\infty &= (\mathsf{C}_\infty \uplus \mathsf{Sf}_\infty) \cap \mathsf{FAmb}_n \\ &= (\mathsf{C}_\infty \uplus \mathsf{Sf}_\infty) \setminus (\mathsf{C}_n \uplus \mathsf{Sf}_n) \subseteq (\mathsf{C}_\infty \setminus \mathsf{C}_n) \uplus (\mathsf{Sf}_\infty \setminus \mathsf{Sf}_n).\end{aligned}$$

Since for all n, $\mathsf{C}_\infty \subseteq \mathsf{C}_n$, one gets:

$$\mathsf{FAmb}_n \setminus \mathsf{FAmb}_\infty \subseteq \mathsf{Sf}_\infty \setminus \mathsf{Sf}_n$$

Let us show that, under the assumption that $\mathcal{A}$ is finitely branching, $\mathsf{Sf}_\infty \subseteq \bigcup_{n\in\mathbb{N}} \mathsf{Sf}_n$. Let $\rho \notin \bigcup_{n\in\mathbb{N}} \mathsf{Sf}_n$. We build a tree as follows:

- Nodes at level n correspond to the correct signalling runs whose observed sequence is $\mathcal{P}(\rho_{\downarrow n})$;
- The node at level $n+1$ associated with ρ' is a child of the node at level n associated with ρ'' if $\rho'' \preceq \rho'$.

Since $\rho \notin \bigcup_{n\in\mathbb{N}} \mathsf{Sf}_n$, for all $n \in \mathbb{N}$, there exists a correct run with observed sequence $\mathcal{P}(\rho_{\downarrow n})$, so that the above-defined tree is infinite. Since the pLTS is finitely branching and convergent, the tree is also finitely branching. By König's lemma, it must contain an infinite branch, thus there exists an infinite correct run whose observed sequence is $\mathcal{P}(\rho)$. As a consequence ρ is not surely faulty: $\rho \notin \mathsf{Sf}_\infty$. This establishes that $\mathsf{Sf}_\infty \subseteq \bigcup_{n\in\mathbb{N}} \mathsf{Sf}_n$. Thus $\lim_{n\to\infty} \mathbb{P}(\mathsf{Sf}_\infty \setminus \mathsf{Sf}_n) = 0$ implying $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n \setminus \mathsf{FAmb}_\infty) = 0$ which concludes the proof. □

We can now establish the following proposition.

Proposition 3.3. *A finitely-branching pLTS $\mathcal{A}$ is* IA*-diagnosable if and only if it admits an* IA*-diagnoser.*

Proof. Assume first that there exists an IA-diagnoser D for $\mathcal{A}$, and let ρ be an infinite run. By reactivity, almost surely $D(\mathcal{P}(\rho)) \in \{\top, \bot\}$. If $D(\mathcal{P}(\rho)) = \top$ then there exists some n such that $D(\mathcal{P}(\rho_{\downarrow n})) = \top$. By correctness, $\rho_{\downarrow n}$ is surely faulty and thus ρ is surely faulty. If $D(\mathcal{P}(\rho)) = \bot$, we claim that ρ is surely correct. Observe that the diagnoser infinitely often outputs $\bot$, so by correctness, for all n, $\mathcal{P}(\rho_{\downarrow n})$ is surely correct and thus in particular $\rho_{\downarrow n}$ is correct. Assume there exists an infinite faulty run ρ' with $\mathcal{P}(\rho') = \mathcal{P}(\rho)$. There exists a n such that for all $m \geq n$, $\rho'_{\downarrow n}$ is faulty. Thus by correctness there can be no more than n $\bot$ verdicts for $\mathcal{P}(\rho)$ contradicting the fact that $D(\mathcal{P}(\rho)) = \bot$. Thus with probability 1, an infinite run is unambiguous.

Conversely, assume that $\mathcal{A}$ is IA-diagnosable. Given an observed sequence w, we denote by N_w the largest integer such that $\mathsf{Cyl}(\mathcal{P}^{-1}(w)) \cap \mathsf{F}_{N_w} = \emptyset$, *i.e.* the largest integer such that every run with observation w was correct after N_w observations. We define the diagnoser D such that $D(\varepsilon) =?$ and for every observed sequence w and observation $a \in \Sigma_o$, if wa is surely faulty, then $D(wa) = \top$, if $N_{wa} > N(w)$, then $D(wa) = \bot$ else $D(wa) =?$.

- Commitment is direct from the definition of D.
- Correctness is achieved as $\top$ is raised for surely faulty runs and $\bot$ is raised when N_w (for appropriate observed sequence w) increased, thus for all $w \in \Sigma_o^*$, $|D(w)|_\bot \leq N_w$ which implies that for a run $\rho \in \mathcal{P}^{-1}(w), \rho_{\downarrow |D(w)|_\bot}$ is correct.
- Reactivity, however is a bit more complicated, we need the result of Lemma 3.1. Let $\rho \notin \mathsf{FAmb}_\infty \cup \mathsf{CAmb}_\infty$.
  - If ρ is correct, then suppose that there exists n_0 such that for all $n \in \mathbb{N}$, there exists a run $\rho^f \in \mathsf{F}_{n_0}$ with $\mathcal{P}(\rho^f_{\downarrow n}) = \mathcal{P}(\rho_{\downarrow n})$. Then using König's lemma and a construction similar to the one of Lemma 3.1, there exists $\rho_f \in \mathsf{F}_{n_0}$ such that $\mathcal{P}(\rho) = \mathcal{P}(\rho^f)$ which would mean $\rho \in \mathsf{CAmb}_\infty$ and raise a contradiction. Thus, as for each n_0 there exists $n \in \mathbb{N}$ such that $N_{\mathcal{P}(\rho_{\downarrow n})} \geq n_0$, $N_{\mathcal{P}(\rho)} = \lim_{n\to\infty} N_{\mathcal{P}(\rho_{\downarrow n})} = \infty$. As every time this value increases, $\bot$ is produced by D, D outputs infinitely many $\bot$, thus $D(\mathcal{P}(\rho)) = \bot$.
  - If ρ is faulty, according to Lemma 3.1, $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n \setminus \mathsf{FAmb}_\infty) = 0$. Therefore, with probability one, there exists $n_0 \in \mathbb{N}$ such that $\rho \notin \mathsf{FAmb}_n$ for $n \geq n_0$. Let n_1 such that $\rho \in \mathsf{F}_{n_1}$ and $n_2 = \max(n_0, n_1)$, then by definition of D, $D(\mathcal{P}(\rho_{\downarrow n_2})) = D(\mathcal{P}(\rho)) = \top$.

  Therefore D is reactive.

Thus, D is an IA-diagnoser. □

IA-diagnosers are thus appropriately associated with IA-diagnosability. They manage to give information after a finite amount of time about infinite ambiguity. We start to see a complexity hierarchy between the various exact diagnosability notions. FF-diagnosability appears as the simplest notion as its equivalence with the existence of

an FF-diagnoser was established in the general case. Second comes IA-diagnosability for which we needed to restrict our framework to finitely-branching pLTS. Finally, the most difficult notion is FA-diagnosability for which the equivalence only holds for finite pLTS. This hierarchy is however only an intuition for now as it could be the result of an inappropriate choice of diagnosers definitions.

1.4 εFF-diagnosers

We now define the εFF-diagnosers corresponding to εFF-diagnosability for $\varepsilon > 0$. Recall that εFF-diagnosability requires that, after any minimal faulty run ρ, the probability of the set of faulty runs extending ρ whose observed sequence has a correctness proportion (CorP, defined page 46) above ε converges to 0 (*i.e.* $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n^\varepsilon) = 0$)[1]. Given $\varepsilon \geq 0$, εFF-diagnosers are similar to FF-diagnosers in the sense that both only consider faulty runs and thus never output a $\bot$ verdict that would give information about correct runs. However, εFF-diagnoser may make errors: when they give a $\top$ verdict, then the probability that the claim is wrong is at most ε.

Definition 3.6. *Let $\varepsilon \geq 0$. An* εFF-diagnoser *for $\mathcal{A}$ is a function $D : \Sigma_o^* \to \{\top, ?\}$ such that:*

correctness *For every $w \in \Sigma_o^*$, if $D(w) = \top$ then $\mathsf{CorP}(w) \leq \varepsilon$;*

reactivity $\limsup_{n\to\infty} \mathbb{P}(\{\rho \in \mathsf{F} \cap \mathsf{SR}_n \mid D(\mathcal{P}(\rho)) = ?\}) = 0$.

Let us comment on this definition. This diagnoser is no longer exact. Given an observed sequence w, it is allowed to output $\top$, if the probability of error is below ε as shown by the requirement $\mathsf{CorP}(w) \leq \varepsilon$. There is no longer a notion of commitment, allowing the diagnoser to go back from a $\top$ verdict to a ? verdict. This absence of commitment is one of the differences between the definition of εFF-diagnosers and the one of monitors for distinguishability of hidden Markov chains [SZF11, KS16]. See the discussion after Lemma 4.3, page 108, for more details about the links between monitoring and diagnosability. Finally, the reactivity condition as defined here is different from what was done for the other diagnosers. This choice was made in order to be closer to the corresponding diagnosability notion as approximate notions of diagnosis are harder to handle than exact ones.

One could introduce a uniform variant of this definition that would correspond to uniform εFF-diagnosability. However, this definition would not follow the same structure as the other ones as a uniform reactivity would have to be defined on individual runs instead of on the global conditions we used here.

Example 3.5. *Let us observe the pLTS on the left of Figure 3.8. We define the diagnoser D such that given an observed sequence $w \in \Sigma_o^*$, then $D(w) = \top$ iff w contains at least as many 'b' than 'a', else $D(w) = ?$. Clearly, this diagnoser does not satisfy any*

[1] This definition is written slightly differently from the one of Definition 2.9, page 48. Yet, one can quickly see that they are equivalent. However, we cannot easily express uniformity with a similar definition.

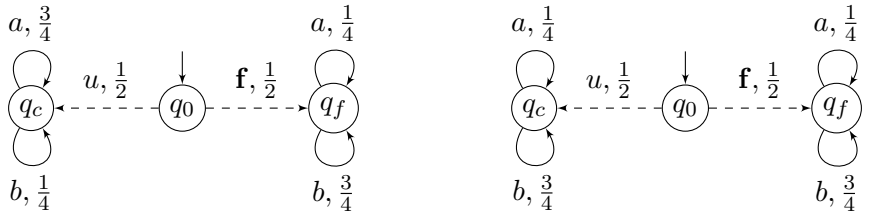


Figure 3.8: Left: an 1/2FF-diagnosable pLTS. Right: a pLTS which is not εFF-diagnosable for any $\varepsilon > 0$.

commitment property, but it is not required for εFF*-diagnosers. Moreover, this diagnoser cannot be translated into a finite-memory diagnoser as we need to keep the difference between the number of occurrences of 'a' and 'b'. We give a visual representation, using an infinite number of states, of this diagnoser in Figure 3.9. Such a diagnoser is a* 1/2FF*-diagnoser:*

correctness *Given an observed sequence* w*, if* $D(w) = \top$*, then* $\mathsf{CorP}(w) = \frac{3^{|w|_a}}{3^{|w|_a}+3^{|w|_b}}$*. As by definition of* D*,* $|w|_a \leq |w|_b$*,* $\mathsf{CorP}(w) \leq 1/2$*, thus* D *is correct.*

reactivity *Let* $\alpha > 0$*, as faulty runs have a probability* $3/4$ *to raise a 'b' at each step, according to the weak law of large number there exists* $n_0 \in \mathbb{N}$ *such that for all* $n \geq n_0$*,* $\mathbb{P}(\{\rho \in \mathsf{F}_n \mid |\mathcal{P}(\rho)|_a > |\mathcal{P}(\rho)|_b\}) < \alpha$*. Let* ρ *be a faulty run such that* $D(\mathcal{P}(\rho)) =?$*. Thus by definition of* D*,* $\rho \in \{\rho' \in \mathsf{F}_n \mid |\mathcal{P}(\rho')|_a > |\mathcal{P}(\rho')|_b\}$*. Therefore, for* $n \geq n_0$*,* $\mathbb{P}(\{\rho \in \mathsf{F} \cap \mathsf{SR}_n \mid D(\mathcal{P}(\rho)) =?\}) \leq \mathbb{P}(\{\rho \in \mathsf{F}_n \mid |w|_a > |w|_b\}) < \alpha$*. Thus* D *is reactive.*

Observe also that this pLTS is indeed 1/2FF*-diagnosable as for* $n \geq 1, \mathbb{P}(\mathsf{FAmb}_n^{1/2}) = \mathbb{P}(\{\rho \in \mathsf{F}_n \mid |\mathcal{P}(\rho_{\downarrow n})|_a > |\mathcal{P}(\rho_{\downarrow n})|_b\})$ *which converges to* 0 *according to the weak law of large numbers.*

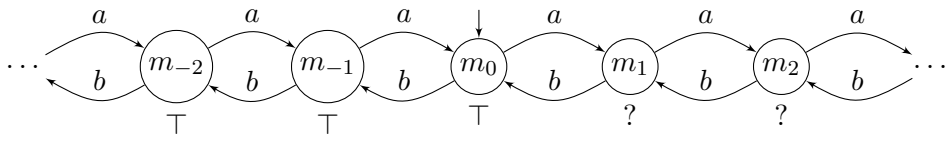


Figure 3.9: An automaton representing the diagnoser of Example 3.5.

The diagnoser used in Example 3.5 requires unbounded memory. In fact, there is no finite-memory 1/2FF-diagnoser for the pLTS on the left of Figure 3.8.

Proposition 3.4. *There exists a* 1/2FF*-diagnosable pLTS that admits no finite-memory* 1/2FF*-diagnoser.*

Proof. Consider the 1/2FF-diagnosable pLTS on the left of Figure 3.8 and assume there exists a 1/2FF-diagnoser with m states. After any sequence a^n for $n \geq 1$, it cannot claim a fault as $\mathsf{CorP}(a^n) = \frac{3^n}{3^n+1} > \frac{1}{2}$. So there exist $1 \leq i < j \leq m+1$ such that the diagnoser is in the same memory state after observing a^i and a^j.
Consider the faulty run $\rho = q_0\mathbf{f}q_f(aq_f)^i$. Due to the reactivity requirement, there must be a run $\rho\rho'$ for which the diagnoser claims a fault. Thus for all n, the diagnoser also claims a fault after $\rho_n = q_0\mathbf{f}q_f(aq_f)^{i+n(j-i)}\rho'$ but $\lim_{n\to\infty} \mathsf{CorP}(\mathcal{P}(\rho_n)) = 1$, which contradicts the correctness requirement. □

We now establish the link between εFF-diagnoser and εFF-diagnosability.

Proposition 3.5. *Let $\varepsilon \geq 0$. A pLTS $\mathcal{A}$ is εFF-diagnosable if and only if it admits an εFF-diagnoser.*

This proof has more computations than the previous ones due to the approximate notion of correctness, however the main ideas are similar: from an εFF-diagnoser we relate $\mathsf{FAmb}^{\varepsilon}_n$ to $\{\rho' \in \mathsf{F}_n \mid D(\mathcal{P}(\rho')) =?\}$ in order to show that the probability of $\mathsf{FAmb}^{\varepsilon}_n$ converges to 0. In the other direction, assuming $\mathcal{A}$ is εFF-diagnosable, we build an εFF-diagnoser that is reactive for the runs that do not belong to $\mathsf{FAmb}^{\varepsilon}_n$ infinitely often.

Proof. Let $\mathcal{A}$ be a pLTS and $\varepsilon \geq 0$. Assume that there exists an εFF-diagnoser D for $\mathcal{A}$. Let ρ be a minimal faulty run and $\alpha > 0$. Since D is reactive, there exists $n_{\rho,\alpha} \in \mathbb{N}$ such that for all $n \geq n_{\rho,\alpha}$,

$$\mathbb{P}(\{\rho' \in \mathsf{F} \cap \mathsf{SR}_n \mid D(\mathcal{P}(\rho')) =?\}) \leq \alpha \cdot \mathbb{P}(\rho) \ .$$

Thus for all $n \geq n_{\rho,\alpha}$:

$$\begin{aligned}\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid D(\mathcal{P}(\rho')) =? \wedge \rho \preceq \rho'\}) &\leq \mathbb{P}(\{\rho' \in \mathsf{F} \cap \mathsf{SR}_{n+|\rho|_o} \mid D(\mathcal{P}(\rho')) =?\}) \\ &\leq \alpha \cdot \mathbb{P}(\rho) \ .\end{aligned}$$

Since D is correct, $\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}^{\varepsilon}_{n+|\rho|_o} \subseteq \mathsf{Cyl}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid D(\mathcal{P}(\rho')) =? \wedge \rho \preceq \rho'\})$. Thus $\mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}^{\varepsilon}_{n+|\rho|_o}) \leq \alpha \cdot \mathbb{P}(\rho)$. This establishes that $\mathcal{A}$ is εFF-diagnosable.
Conversely assume that $\mathcal{A}$ is εFF-diagnosable. Let D be the diagnoser defined by: for all $w \in \Sigma_o^*$, $D(w) = \top$ iff $\mathsf{CorP}(w) \leq \varepsilon$. Such an εFF-diagnoser is correct by definition. Let $\alpha > 0$. Since $(\mathsf{F}_n)_{n\in\mathbb{N}}$ is a non-decreasing sequence converging to F_∞, there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0, \mathbb{P}(\mathsf{F}_n \setminus \mathsf{F}_{n_0}) < \alpha/2$. By εFF-diagnosability of $\mathcal{A}$, for all $\rho \in \bigcup_{k\leq n_0} \mathsf{minF}_k$, there exists n_ρ such that for all $n \geq n_\rho$

$$\mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}^{\varepsilon}_{n+|\rho|_o}) \leq \frac{\alpha}{2} \cdot \mathbb{P}(\rho).$$

Define $n_{max} = \max_{\rho \in \bigcup_{n \leq n_0} \mathsf{minF}_n} {n_\rho}^2$. Then for $n \geq n_0 + n_{max}$ we have

$$\begin{aligned}\mathbb{P}(\mathsf{FAmb}_n^\varepsilon) &\leq \mathbb{P}(\mathsf{FAmb}_n^\varepsilon \cap \mathsf{F}_{n_0}) + \mathbb{P}(\mathsf{FAmb}_n^\varepsilon \setminus \mathsf{F}_{n_0}) \\ &\leq \sum_{\rho \in \bigcup_{k \leq n_0} \mathsf{minF}_k} \mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}_n^\varepsilon) + \mathbb{P}(\mathsf{F}_n \setminus \mathsf{F}_{n_0}) \\ &\leq \sum_{\rho \in \bigcup_{k \leq n_0} \mathsf{minF}_k} \frac{\alpha}{2} \cdot \mathbb{P}(\rho) + \frac{\alpha}{2} \leq \alpha.\end{aligned}$$

So we have established that $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n^\varepsilon) = 0$.
By definition of D, $\mathsf{FAmb}_n^\varepsilon = \{\rho \in \mathsf{F}_n \mid D(\mathcal{P}(\rho)) = ?\}$. Thus D is reactive. □

For every $\varepsilon \geq 0$ we have thus an appropriate notion of εFF-diagnoser associated with εFF-diagnosability. As a pLTS is AFF-diagnosable if it is εFF-diagnosable for every $\varepsilon > 0$, we directly obtain the following corollary.

Corollary 3.1. *A finite pLTS $\mathcal{A}$ is* AFF*-diagnosable if and only if for all $\varepsilon > 0$ it admits an* εFF*-diagnoser.*

In other words, when a pLTS is AFF-diagnosable, the designer can choose the accuracy they want for the diagnoser.

2 Relationships between diagnosability notions

In this section, we establish the links between the multiple notions of diagnosability defined in Section 1.5. We gave in Section 1 diagnosers associated with the various notions of diagnosability. Thus, intuitively requiring a stronger version of one feature (verdict, correctness or reactivity) defines a diagnoser that gives more information and thus which is less likely to exist. For example, the difference between FF-diagnosability and FA-diagnosability is that FA-diagnosability must identify faulty and correct runs, while FF-diagnosability only cares about faulty runs. Thus FA-diagnosability implies FF-diagnosability, which can be formally proven immediately since for all n, $\mathsf{FAmb}_n \subseteq \mathsf{FAmb}_n \uplus \mathsf{CAmb}_n$. The two notions being entirely distinct as FF-diagnosability does not imply FA-diagnosability as shown by the pLTS of Figure 3.10: there is a single ambiguous observed sequence for every $n \in \mathbb{N}$, a^n, this sequence can be observed with a probability $1/2$ of correct runs and by a probability $1/2^n$ of faulty runs, *i.e.* $\forall n \in \mathbb{N}, \mathbb{P}(\mathsf{CAmb}_n) = 1/2$ and $\mathbb{P}(\mathsf{FAmb}_n) = 1/2^n$ therefore it is FF-diagnosable without being FA-diagnosable. Similarly, as $\mathsf{FAmb}_\infty \subseteq \mathsf{FAmb}_\infty \uplus \mathsf{CAmb}_\infty$, IA-diagnosability implies IF-diagnosability The converse is not true however as the pLTS of Figure 3.10[3] is IF-diagnosable while it is not IA-diagnosable.

An interesting case is the link between IF-diagnosability and FF-diagnosability. Intuitively, as for IF-diagnosability we can observe the infinite run before giving a verdict

[2]Note that $\bigcup_{n \leq n_0} \mathsf{minF}_n$ is finite due to $\mathcal{A}$ being finitely branching and convergent.
[3]This pLTS was already displayed in Figure 3.2.

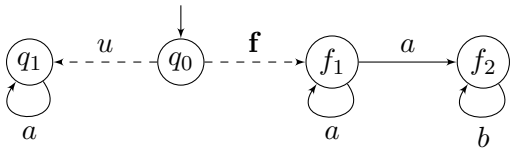


Figure 3.10: An FF/IF-diagnosable pLTS which is not FA/IA-diagnosable.

while FF-diagnosability only allows for a finite observation, FF-diagnosability implies IF-diagnosability. This is indeed true. Using the first direction of the Lemma 3.1 we know that $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_\infty \setminus \mathsf{FAmb}_n) = 0$. Thus if a pLTS is FF-diagnosable, $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = 0$, therefore $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_\infty) \leq \lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_\infty \setminus \mathsf{FAmb}_n) + \lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = 0$. Hence the pLTS is IF-diagnosable. Moreover, the implication is strict. Observe the pLTS of Figure 3.11, it is IF-diagnosable (and even IA-diagnosable in fact) as every correct run ends with an infinity of b which cannot be observed with a faulty run and the only faulty run only triggers the observation a, thus $\mathsf{FAmb}_\infty \uplus \mathsf{CAmb}_\infty = \emptyset$. However, it is not FF-diagnosable as for all $n \in \mathbb{N}, \mathbb{P}(\mathsf{FAmb}_n) = 1/2$. Indeed, for $n \in \mathbb{N}$, there exists a transition from q_0 to a state q_{n1} labelled by an a and with probability $1/2^{n+1}$, the one correct run ρ of observable length$n$ starting by this transition has observation $\mathcal{P}(\rho) = a^n$, thus the only faulty run of observable length n, $q_0\mathbf{f}(f_1a)^n f_1$ is ambiguous. Moreover this faulty run has probability $1/2$. Although we have a strict implication, the pLTS used to prove the strictness has an infinite branching. This is in fact necessary: given a finitely-branching convergent pLTS, IF-diagnosability is equivalent to FF-diagnosability. Given a pLTS $\mathcal{A}$, the implication from IF-diagnosability to FF-diagnosability is shown using once again Lemma 3.1, which, assuming finite branching, says that $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n \setminus \mathsf{FAmb}_\infty) = 0$. Thus if $\mathcal{A}$ is IF-diagnosable, $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = \lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n \setminus \mathsf{FAmb}_\infty) + \mathbb{P}(\mathsf{FAmb}_\infty) = 0$. Thus $\mathcal{A}$ is FF-diagnosable.

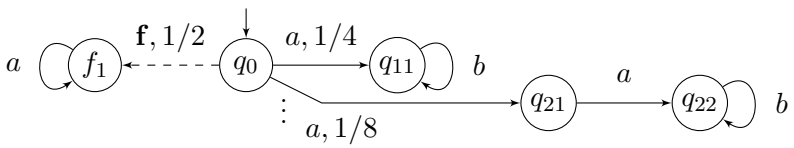


Figure 3.11: An infinitely-branching pLTS that is IA-diagnosable but not FF-diagnosable.

Although the relationship between IA-diagnosability and FA-diagnosability is the same as the one between IF-diagnosability and FF-diagnosability, the same link cannot be established. Indeed, even for finite pLTS, IA-diagnosability does not imply FA-diagnosability. Let us observe the pLTS of Figure 3.12. It is IA-diagnosable, indeed, every infinite correct run have observed sequence a^ω while the observed sequence of every infinite faulty run is of the form $a^n b^\omega$ for $n > 0$, thus $\mathsf{CAmb}_\infty \uplus \mathsf{FAmb}_\infty = \emptyset$.

Consider however the infinite correct run $\rho = q_0 u q_1 (a q_1)^\omega$. It has probability $\frac{1}{2}$, and all its finite signalling subruns are ambiguous since their observed sequence is a^n, for some $n \in \mathbb{N}$ which is the observed sequence of the faulty signalling run $q_0 u (q_2 a)^{n-1} q_2 \mathbf{f} f_1 a f_2$. Thus for all $n \geq 1$, $\mathbb{P}(\mathsf{CAmb}_n) \geq \frac{1}{2}$, so that this pLTS is not FA-diagnosable.

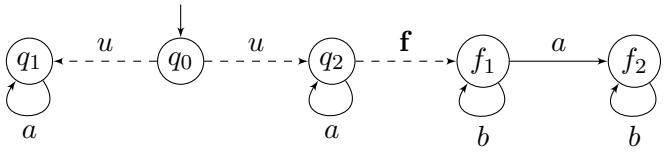
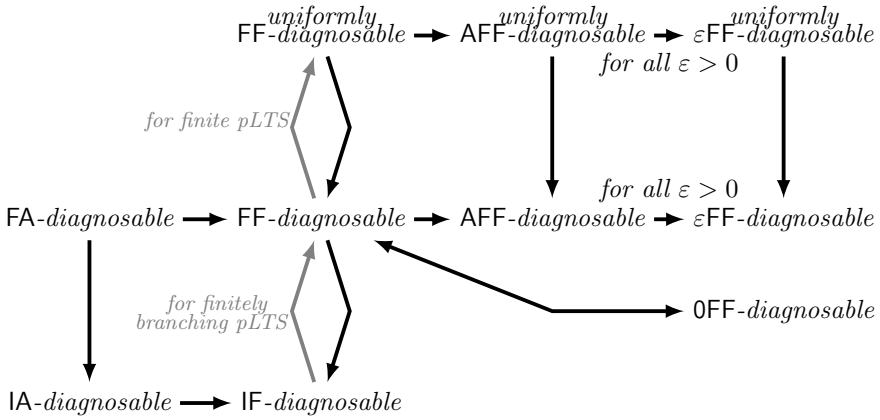


Figure 3.12: An IA-diagnosable pLTS which is not FA-diagnosable.

The next theorem summarises the connections between the different diagnosability notions.

Theorem 3.1. *The diagnosability notions for pLTS are related according to the diagram below, where arrows represent implications. All implications, except the one from* IF*-diagnosability to* FF*-diagnosability and the one from* FF*-diagnosability to uniform* FF*-diagnosability, hold for arbitrary infinite-state pLTS. The latter implications holds for finitely-branching pLTS and finite pLTS respectively. Implications that are not depicted do not hold, already in the case of finite-state pLTS.*



Let us first describe this diagram. Omitting 0FF-diagnosability which is equivalent to FF-diagnosability, the first two columns correspond to exact diagnosis (thus diagnosis where the diagnoser cannot claim a fault if there is a probability that the fault did not occur) while the last two rows correspond to approximate diagnosis. The lower row contains the notions of diagnosis considering infinite runs, the middle row the ones considering finite runs and the upper rows the ones considering finite runs and requiring uniformity on the speed of reactivity of the diagnoser. From any notion of diagnosability, the notion above it, if there is one, has a more restrictive reactivity and the one on its left requires a better correctness or has a verdict extended to more elements.

FF-diagnosability plays a central role in the diagram as the notion of uniformity and approximate diagnosis was built from this notion. This notion was chosen as it is the traditional notion of diagnosis [TT05], moreover there is no clear intuition of what approximation and uniformity means for infinite runs.

Proof. We prove all implications that were not already stated in the beginning of this section. Most are pretty straightforward. For all $\varepsilon > 0$, the implications from AFF- to εFF, uniform AFF- to uniform εFF- and uniform εFF- to εFF- are direct by definitions. The implications from uniform AFF- to AFF-, uniform FF- to FF-, FF- to AFF- and uniform FF- to uniform AFF- comes partially from definitions and from application of other implications (mostly the equivalence between FF- and 0FF- proven below). The most complicated implications are (1) the equivalence between FF- and 0FF-diagnosis which is a careful inspection of sets of runs, taking account the possibly infinite branching, (2) the implication from FA- to IA-diagnosis which is inspired by Lemma 3.1 and (3) the implication from FF- to uniform FF-diagnosability for finite pLTS which requires the characterisation of FF-diagnosability for finite pLTS which will be established later and is thus postponed (See Proposition 4.3, page 95).

FF $\Leftrightarrow$ 0FF.

Let $\mathcal{A}$ be a 0FF-diagnosable pLTS and $\varepsilon > 0$. Since $(\mathsf{F}_n)_{n\in\mathbb{N}}$ is a non-decreasing sequence converging to F_∞, there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0, \mathbb{P}(\mathsf{F}_n \setminus \mathsf{F}_{n_0}) < \frac{\varepsilon}{2}$. By 0FF-diagnosability of $\mathcal{A}$, for all $\rho \in \bigcup_{k\leq n_0} \mathsf{minF}_k$, there exists n_ρ such that for all $n \geq n_\rho$

$$\mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}_{n+|\rho|_o}) \leq \frac{\varepsilon}{4} \cdot \mathbb{P}(\rho).$$

Notice that, because the pLTS may be infinitely branching, the set $\bigcup_{k\leq n_0} \mathsf{minF}_k$ may be infinite. We therefore define n_{max} such that $\mathbb{P}(\{\rho \in \bigcup_{k\leq n_0} \mathsf{minF}_k \mid n_\rho > n_{max}\}) \leq \frac{\varepsilon}{4}$. Thus, only a small portion of runs ρ in $\bigcup_{k\leq n_0} \mathsf{minF}_k$ have $n_\rho > n_{max}$. Then for $n \geq n_0 + n_{max}$ we have

$$\begin{aligned}
\mathbb{P}(\mathsf{FAmb}_n) \leq & \quad \mathbb{P}(\mathsf{FAmb}_n \setminus \mathsf{F}_{n_0}) + \mathbb{P}(\mathsf{FAmb}_n \cap \mathsf{F}_{n_0}) \\
\leq & \quad \mathbb{P}(\mathsf{FAmb}_n \setminus \mathsf{F}_{n_0}) + \mathbb{P}(\{\rho \in \bigcup_{k\leq n_0} \mathsf{minF}_k \mid n_\rho > n_{max}\}) \\
& + \mathbb{P}(\{\rho' \in \mathsf{FAmb}_n \mid \exists \rho \in \bigcup_{k\leq n_0} \mathsf{minF}_k,\ \rho \preceq \rho',\ n_\rho \leq n_{max}\}) \\
\leq & \quad \frac{\varepsilon}{2} + \frac{\varepsilon}{4} + \frac{\varepsilon}{4}\mathbb{P}(\{\rho \in \bigcup_{k\leq n_0} \mathsf{minF}_k \mid n_\rho \leq n_{max}\}) \leq \varepsilon \ .
\end{aligned}$$

Let $\mathcal{A}$ be a FF-diagnosable pLTS. Consider $\rho \in \mathsf{minF}$ and $\alpha > 0$. There exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $\mathbb{P}(\mathsf{FAmb}_n) \leq \alpha \cdot \mathbb{P}(\rho)$. Thus for all $n \geq n_0$:

$$\mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}_{n+|\rho|_o}) \leq \mathbb{P}(\mathsf{FAmb}_{n+|\rho|_o}) \leq \alpha \cdot \mathbb{P}(\rho).$$

FA $\Rightarrow$ IA.

For all $n \in \mathbb{N}$, define $\mathsf{CAmb}_{n,\infty}$ the set of correct ambiguous runs that admit an observationally equivalent run which is faulty before its n^{th} observable event. Observe that the sequence of sets $\{\mathsf{CAmb}_{n,\infty}\}_{n\in\mathbb{N}}$ is non-decreasing and that $\mathsf{CAmb}_\infty = \bigcup_{n\in\mathbb{N}} \mathsf{CAmb}_{n,\infty}$. Moreover, by definition, $\mathsf{CAmb}_{n,\infty} \subseteq \mathsf{CAmb}_n$. Assume that $\limsup_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n \uplus \mathsf{CAmb}_n) = 0$. By Lemma 3.1, $\mathbb{P}(\mathsf{FAmb}_\infty) = 0$. For all $\varepsilon > 0$, there exists $n_1 \in \mathbb{N}$ such that for all $n \geq n_1$, $\mathbb{P}(\mathsf{CAmb}_n) < \varepsilon$ and thus $\mathbb{P}(\mathsf{CAmb}_{n,\infty}) < \varepsilon$. On the other hand, there exists $n_2 \in \mathbb{N}$ such that for all $n \geq n_2$, $\mathbb{P}(\mathsf{CAmb}_\infty) - \mathbb{P}(\mathsf{CAmb}_{n,\infty}) < \varepsilon$. Combining these two inequalities for $n = \max(n_1, n_2)$, one obtains $\mathbb{P}(\mathsf{CAmb}_\infty) < 2\varepsilon$. As ε is arbitrary, $\mathbb{P}(\mathsf{CAmb}_\infty) = 0$.

We now provide counter-examples for the implications that do not hold and which were not already developed at the beginning of the section. The most interesting example is the one establishing the difference between FA-diagnosis to uniform εFF-diagnosis as it requires an infinite pLTS.

uniform AFF $\not\Rightarrow$ IF.

Consider the pLTS depicted on the left of Figure 3.8. All infinite faulty runs are ambiguous, and the probability of faulty runs is $\frac{1}{2}$, thus this pLTS is not IF-diagnosable. Fix some $\varepsilon > 0$ and $0 < \alpha < 1$. There are two minimal faulty runs $\rho_a = q_0\mathbf{f}q_f a q_f$ and $\rho_b = q_0\mathbf{f}q_f b q_f$. Consider first ρ_a and let ρ be the random variable of a signalling run of length n that extends ρ_a. One can express the correctness proportion of ρ in terms of the number of a's in its observed sequence, written $|\rho|_a$:

$$\mathsf{CorP}(\rho) = \frac{(\frac{3}{4})^{|\rho|_a}(\frac{1}{4})^{|\rho|-|\rho|_a}}{(\frac{3}{4})^{|\rho|_a}(\frac{1}{4})^{|\rho|-|\rho|_a} + (\frac{1}{4})^{|\rho|_a}(\frac{3}{4})^{|\rho|-|\rho|_a}}.$$

Simplifying this expression, we obtain: $\mathsf{CorP}(\rho) = \frac{1}{1+3^{|\rho|-2|\rho|_a}}$. Now, by the strong law of large numbers, for any $\eta > 0$, there exists n_η such that for every $n \geq n_\eta$, $\mathbb{P}(|4|\rho|_a - |\rho|| > \eta) < \alpha$. So with probability at least $1-\alpha$, the correctness proportion of ρ is bounded by $\frac{1}{1+3^{\frac{\eta+|\rho|}{2}}}$. For a sufficiently large η, this value is smaller than ε, so that $\mathbb{P}(\mathsf{CorP}(\rho) \leq \varepsilon) \geq 1-\alpha$.

A similar reasoning applies to ρ_b, and one can then take the maximum of the two integers n_η to prove that the pLTS is uniformly AFF-diagnosable.

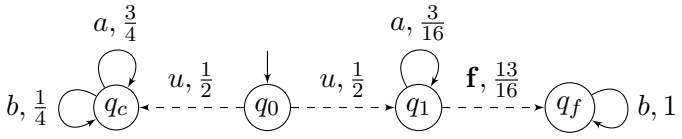


Figure 3.13: An AFF-diagnosable pLTS which is not uniformly εFF-diagnosable for $\varepsilon < 3/4$.

AFF$\not\Rightarrow$ uniform εFF.

Consider the pLTS of Figure 3.13. Fix some $0 < \varepsilon < \frac{3}{4}$, $0 < \alpha < 1$ and n_α. Consider the minimal faulty run $\rho = q_0 u q_1 (a q_1)^{n_\alpha+1} \mathbf{f} q_f b q_f$. Let ρ' be the signalling run of length $2n_\alpha + 2$ such that $\rho \preceq \rho'$. $\mathcal{P}(\rho') = a^{n_\alpha+1} b^{n_\alpha+1}$. Thus, $\mathsf{CorP}(\mathcal{P}(\rho')) \geq \frac{3}{4}$. So

$$\mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}^{\varepsilon}_{2n_\alpha+2}) = \mathbb{P}(\rho) > \alpha \cdot \mathbb{P}(\rho).$$

Thus the pLTS is not uniformly εFF-diagnosable.
Let ρ be a minimal faulty run. Then $\mathcal{P}(\rho) = a^{n_0} b$ for some n_0. For all n, let ρ_n be the single the single signalling run of observable length $|\rho| + n$ that extends ρ. It fulfils $\mathcal{P}(\rho_n) = a^{n_0} b^{n+1}$ and $\mathbb{P}(\rho_n) = \mathbb{P}(\rho)$. The single correct signalling run ρ'_n with $\mathcal{P}(\rho'_n) = \mathcal{P}(\rho_n)$ fulfils $\mathbb{P}(\rho'_n) = \frac{3^{n_0}}{2 \cdot 4^{n_0+n+1}}$. Thus $\lim_{n\to\infty} \mathsf{CorP}(\mathcal{P}(\rho_n)) = 0$. So the pLTS is εFF-diagnosable for all $\varepsilon > 0$ and thus AFF-diagnosable.

FA $\not\Rightarrow$ uniform εFF when considering infinite pLTS.

Let us consider the pLTS of Figure 3.14. It is FA-diagnosable as almost surely a faulty (resp. correct) run contains a b (resp. c) after a finite number of steps that cannot be mimicked by a correct (resp. faulty) run. We claim that it is not uniformly εFF-diagnosable for all ε such that $0 < \varepsilon < \frac{1}{2}$. Note that for all $n \in \mathbb{N}$, $\mathsf{CorP}(a^n) \geq \frac{1}{2}$. Fix some $0 < \alpha < 1$ and $n_\alpha \in \mathbb{N}$. Consider the minimal faulty run $\rho = q_0 u f_1 a f_2 \dots a f_{n_\alpha} \mathbf{f} f'_{n_\alpha}$. The shortest extension of ρ that is not ambiguous (*i.e.* contains a b) contains $n_\alpha + 1$ observable events more than ρ does. Therefore, $\mathbb{P}(\{\rho' \in \mathsf{FAmb}^{\varepsilon}_{n_\alpha+|\rho|_o} \mid \rho \preceq \rho'\}) = \mathbb{P}(\rho) > \alpha \cdot \mathbb{P}(\rho)$.

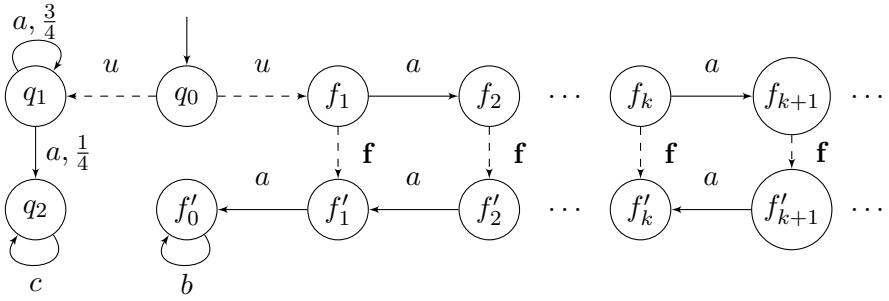


Figure 3.14: An infinite FA-diagnosable pLTS that is not uniformly εFF-diagnosable.

uniform εFF $\not\Rightarrow$ AFF.

Consider the pLTS of Figure 3.15. There is a single signalling minimal faulty run $q_0 \mathbf{f} q_f a q_f$. Any observed sequence of length at least 1 is ambiguous and corresponds with equal probability to a signalling correct or a faulty run. Consequently it is not AFF-diagnosable, yet it is uniformly εFF-diagnosable for $\varepsilon = \frac{1}{2}$.

This concludes the proof. □

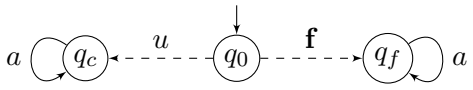


Figure 3.15: A uniform $\frac{1}{2}$FF-diagnosable pLTS, that is not AFF-diagnosable.

3 Characterisation of diagnosability

Our goal in this section is to establish "simple" characterisations of the diagnosability notions for a pLTS. More precisely, we aim at studying whether there exists a Borel set $B \in \mathcal{B}$ that only depends on the underlying LTS such that almost surely a random run belongs to B if and only if the pLTS is diagnosable. Allowing B to depend on the probabilities of the pLTS would give more expressivity, but strongly increases the complexity of the information contained in the set which goes against the goal of a "simple" characterisation. Similarly, as much as possible, we look for a set B belonging to a low level of the Borel hierarchy.

Approximate notions of diagnosability heavily depend on the probabilities of the transitions. This can be seen on the two pLTS of Figure 3.8, page 69 for example: while the one on the left is εFF-diagnosable for every $\varepsilon > 0$, the one on the right, which is obtained simply by swapping two probabilities, is not εFF-diagnosable system for all $\varepsilon > 0$. We thus focus here on FA/FF/IA/IF-diagnosis. For these notions, the characterising Borel sets cannot be obtained directly from the definitions and require some machinery. Indeed, the notions of FF- and FA-diagnosability are expressed by the limit of the probability of a family of open sets and the notions of IF- and IA-diagnosability are expressed by a set which is not *a priori* a Borel set.

We will specify these Borel sets using a logic called pathL. Logics are tools that are efficient at giving specifications for a system and, thanks to model checking, at deciding if these specifications are satisfied by the system. The complexity of model checking a logic depends on the kind of systems considered (finite, probabilistic,...) and on the power of expressivity of the logic. This makes thus yet another argument for requiring simple characterisations of the diagnosability notions.

In this section, we define the logic pathL in Subsection 3.1. Then, in Subsection 3.2 we give, whenever possible, a characterisation of the different diagnosability problems. Finally in Subsection 3.3, we provide impossibility results that justify why characterisations were not given for some of the diagnosability notions.

3.1 The logic pathL

We define the logic pathL in this section. Similarly to Probabilistic Linear Time Logic (PLTL) [CY95], pathL first defines a specification, then specifies a probabilistic condition over this specification. The main difference with PLTL is that pathL is based on the notion of *path formulae* instead of using atomic propositions. A path formula $\mathfrak{p}$ is a predicate over finite prefixes of runs. Before defining their syntax formally, let us first

give some examples of path formulae.

Example 3.6. *Given a finite run $\rho = q_0 a_0 q_1 \dots q_k$, let $\mathfrak{f}$ be defined by $\mathfrak{f}(\rho) = \mathsf{true}$ if $a_i = \mathbf{f}$ for some $i < k$. This path formula characterises the faulty finite runs.*

Let $\mathfrak{U}$ be defined by $\mathfrak{U}(\rho) = \mathsf{true}$ if there exists a correct signalling run ρ' with $\mathcal{P}(\rho) = \mathcal{P}(\rho')$. If this path formula is false*, the current run is surely faulty.*

Let us introduce a more intricate path formula. For $\sigma \in \Sigma_o^*$, we define $\mathsf{firstf}(\sigma)$ as the smallest value such that there exists a faulty run with observation σ such that its prefix of length $\mathsf{firstf}(\sigma)$ is faulty, i.e. $\mathsf{firstf}(\sigma) = \min\{k \mid \exists \rho$ signalling run $\mathcal{P}(\rho) = \sigma \wedge \rho_{\downarrow k}$ is faulty$\}$ with the convention that $\min(\emptyset) = \infty$. Then the path formula $\mathfrak{W}$ is defined by: $\mathfrak{W}(\varepsilon) = \mathsf{false}$ and $\mathfrak{W}(q_0 a_0 \dots q_{n+1}) = \mathsf{true}$ if $\mathsf{firstf}(\mathcal{P}(q_0 a_0 \dots q_{n+1})) = \mathsf{firstf}(\mathcal{P}(q_0 a_0 \dots q_n)) < \infty$. Every time this path formula is false*, we increased the size of the greatest prefix that we are sure is correct.*

As shown by the path formula $\mathfrak{U}$ for example, path formulae are extremely strong as they may depend on the global structure of the system, here by depending on the other existing runs with the same observation. This is far stronger than atomic propositions used in PLTL for example that only depend on the current state of the run.

Formally, a path formula is either generated by a context sensitive grammar or equivalently its acceptance is decided by a linear bounded automaton [Kur64]. In other words, one has to be able to determine the truth of a path formula in linear space.

Example 3.7. *Among the examples of path formulae given in Example 3.6, the most difficult one is $\mathfrak{W}$. Let us show how one can compute the truth of this path formula in linear space. We first define for $\sigma \in \Sigma_o^*$ and $q \in Q$, the restriction of* firstf *to q: $m(\sigma, q) = \min\{k \mid \exists \rho \in \mathsf{SR}, \mathsf{last}(\rho) = q \wedge \mathcal{P}(\rho) = \sigma \wedge \rho_{\downarrow k}$ is faulty$\}$ with the convention that $\min(\emptyset) = \infty$.*

Let ρ be a finite run, σ its observed sequence. If $\sigma = \varepsilon$, $\mathfrak{W}(\rho) =$ false*. Else $\sigma = \sigma_1 \dots \sigma_n$ For every $q \in Q$, we compute the values $m(\sigma, q)$ and $m(\sigma_1 \dots \sigma_{n-1}, q)$ with the following algorithm:*

Algorithm 1 Computing the values of m

Input: pLTS $\mathcal{A}$, $\sigma \in \Sigma_o^*$
Output: $(m(\sigma_{\leq k}, q))_{k \leq |\sigma|, q \in Q}$
for $q \in Q$ **do**
    $m(\varepsilon, q) \leftarrow \infty$
for $i = 1$ *to* n **do**
    for $q, q' \in Q$ **do**
        $m(\sigma_1 \dots \sigma_i, q) \leftarrow \infty$
        if $q \Rightarrow_f^{\sigma_i} q'$ **then**
            $m(\sigma_1 \dots \sigma_i, q') \rightarrow \min(m(\sigma_1 \dots \sigma_{i-1}, q'), i)$
        if $q \Rightarrow^{\sigma_i} q'$ **then**
            $m(\sigma_1 \dots \sigma_i, q') \rightarrow \min(m(\sigma_1 \dots \sigma_{i-1}, q'), m(\sigma_1 \dots \sigma_{i-1}, q))$

This algorithm uses linear space in n. Moreover, we have an equivalence between $\mathfrak{W}(\rho) = \mathsf{true}$ and $\max_{q\in Q}\{m(\sigma_1 \dots \sigma_{n-1}, q)\} < \max_{q\in Q}\{m(\sigma, q)\} < \infty$. Thus the truth of $\mathfrak{W}$ can be decided by a linear bounded automaton. It is therefore a path formula.

We now define the syntax of pathL.

Definition 3.7. *The syntax of a* pathL formula *is:*

$$\phi ::= \mathfrak{p} \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Diamond\phi$$

where $\mathfrak{p}$ is a path formula. In the sequel we use the standard shortcut $\Box\phi \equiv \neg\Diamond\neg\phi$.

A formula is evaluated at some position k of a run $\rho = q_0a_0q_1 \dots$. The prefix $\rho_{\leq k}$ of ρ is defined by $\rho_{\leq k} = q_0a_0q_1 \dots q_k$. The semantics of pathL is inductively defined by:

- $\rho, k \models \alpha$ if $\alpha(\rho_{\leq k})$, note that only the past of ρ is used;
- $\rho, k \models \neg\phi$ if $\rho, k \not\models \phi$;
- $\rho, k \models \phi_1 \wedge \phi_2$ if $\rho, k \models \phi_1$ and $\rho, k \models \phi_2$;
- $\rho, k \models \Diamond\phi$ if there exists $k' \geq k$ such that $\rho, k' \models \phi$.

Finally $\rho \models \phi$ if $\rho, 0 \models \phi$. Due to the presence of path formulae (with no restriction) this language subsumes LTL and more generally any ω-regular specification language, *i.e.* any language that can be recognized by an ω-automaton such as a Rabin automaton.

Proposition 3.6. *The language generated by* pathL *subsumes ω-regular languages.*

Proof. The language of a deterministic Rabin automaton is determined by a finite family of pair of sets (E_i, F_i). It consists of the set of runs ρ for which there exists $i \in \mathbb{N}$ such that ρ visits finitely often the states of E_i and infinitely often the states of F_i. We define the path formulae $\mathfrak{E}_i$ and $\mathfrak{F}_i$ such that $\mathfrak{E}_i(\rho) = \mathsf{true}$ iff $\mathsf{last}(\rho) \in E_i$ and $\mathfrak{F}_i(\rho) = \mathsf{true}$ iff $\mathsf{last}(\rho) \in F_i$. The runs accepted by the Rabin automaton equivalently satisfy the formula $\bigvee_i(\Diamond\Box(\neg\mathfrak{E}_i) \wedge (\Box\Diamond\mathfrak{F}_i))$. The language accepted by the Rabin automaton is thus generated by pathL. □

In order to reason about the probabilistic behaviour of a pLTS, we introduce the notion of qualitative probabilistic formulae:

Definition 3.8. *The syntax of a* qualitative probabilistic formula *of* pathL *is: $\mathbb{P}^{\bowtie p}(\phi)$ with $\bowtie \in \{<, >, =\}$, $p \in \{0, 1\}$ and $\phi \in$ pathL.*

The semantics is obvious: $\mathcal{A} \models \mathbb{P}^{\bowtie p}(\phi)$ if and only if $\mathbb{P}(\{\rho \in \Omega \mid \rho \models \phi\}) \bowtie p$.

The set of Borel sets defined by pathL is closed by complementation since the complement of the set of runs generated by a formula ϕ is generated by the formula $\neg\phi$. Therefore given a pLTS $\mathcal{A}$ and a formula ϕ, $\mathcal{A} \models \mathbb{P}^{=1}(\phi)$ iff $\mathcal{A} \models \mathbb{P}^{=0}(\neg\phi)$ and $\mathcal{A} \models \mathbb{P}^{<1}(\phi)$ iff $\mathcal{A} \models \mathbb{P}^{>0}(\neg\phi)$. The qualitative probabilistic formulae can thus be restricted to $\mathbb{P}^{=0}(\phi)$ and $\mathbb{P}^{>0}(\phi)$.

3.2 Logical characterisation of diagnosability

We now exhibit qualitative probabilistic formulae ψ such that a pLTS $\mathcal{A}$ is diagnosable iff $\mathcal{A} \models \psi$.

FF-diagnosability. We start with FF-diagnosability. This notion seems to be the easiest one as it focuses on faulty runs and its definition already uses a family of Borel sets. Indeed, we can prove a simple characterisation of FF-diagnosability using the path formulae $\mathfrak{f}$ and $\mathfrak{U}$ that we defined in Example 3.6.

Proposition 3.7. *Let $\mathcal{A}$ be a pLTS. Then $\mathcal{A}$ is* FF*-diagnosable iff $\mathcal{A} \models \mathbb{P}^{=0}(\Diamond\Box(\mathfrak{f} \wedge \mathfrak{U}))$.*

Once a run becomes surely faulty, it cannot become ambiguous. Thus, informally, this formula means that a pLTS is FF-diagnosable if the measure of runs that are infinitely often faulty and ambiguous is equal to 0. These runs are the faulty runs for which the fault can never be claimed by the FF-diagnoser. It is thus natural to require their probability to be equal to 0.

Proof. We write $E = \{\rho \in \Omega \mid \Diamond\Box(\mathfrak{f} \wedge \mathfrak{U})\}$ for the set of runs we are interested in. We further define, for every $\rho \in \mathsf{minF}$, $E_\rho = \{\rho' \in \Omega \mid \rho \preceq \rho' \wedge \rho' \models \Box\mathfrak{U}\}$ and for every $n \in \mathbb{N}$, $E^n_\rho = \{\rho' \in \Omega \mid \rho \preceq \rho' \wedge \rho' \models \Box^n\mathfrak{U}\}$ where $\rho \models \Box^n\phi$ if for every $k \leq n$, $\rho, k \models \phi$. As $\rho \in \mathsf{minF}$ and $\rho', n \models \mathfrak{f} \wedge \mathfrak{U}$ implies $\rho'[0,n]$ is faulty and ambiguous, we have for all $n \geq 0$ $E^n_\rho \subseteq \mathsf{FAmb}_{n+|\rho|}$. Observe that $E = \biguplus_{\rho\in\mathsf{minF}} E_\rho$ and that $E_\rho = \cap_{n\in\mathbb{N}} E^n_\rho$. Thus $\mathbb{P}(E) = \sum_{\rho\in\mathsf{minF}} \mathbb{P}(E_\rho)$ and $\lim_{n\to\infty} \mathbb{P}(E^n_\rho) = \mathbb{P}(E_\rho)$.
• Assume first that $\mathbb{P}(E) > 0$. Then, there exists $\rho \in \mathsf{minF}$ such that $\mathbb{P}(E_\rho) > 0$. By definition, for every $n > |\rho|_o$ $\mathbb{P}(\mathsf{FAmb}_n) \geq \mathbb{P}(E_\rho)$. Thus, $\mathcal{A}$ is not FF-diagnosable.
• Assume now that $\mathbb{P}(E) = 0$. So, for every $\rho \in \mathsf{minF}$, $\mathbb{P}(E_\rho) = 0$. Let us pick some $\varepsilon > 0$. Since $\mathsf{F} = \bigcup_{n\in\mathbb{N}} \mathsf{F}_n$, there exists n_0 such that for every $n \geq n_0$, $\mathbb{P}(\mathsf{F} \setminus \mathsf{F}_n) \leq \frac{\varepsilon}{3}$. Let $R = \{\rho \in \mathsf{minF} \mid |\rho|_o < n_0\}$. Pick a finite subset R' of R such that $\sum_{\rho\in R\setminus R'} \mathbb{P}(\rho) \leq \frac{\varepsilon}{3}$. Define $K = |R'|$. Let n_1 be such that for every $n \geq n_1$ and every $\rho \in R'$, $\mathbb{P}(E^n_\rho) \leq \frac{\varepsilon}{3K}$. Observe now that for every $n \geq n_0$, $\mathsf{FAmb}_n \subseteq (\mathsf{F} \setminus \mathsf{F}_n) \cup \biguplus_{\rho\in R\setminus R'} C(\rho) \cup \bigcup_{\rho\in R'} E^n_\rho$. Thus, for every $n \geq n_1$, $\mathbb{P}(\mathsf{FAmb}_n) \leq \frac{\varepsilon}{3} + \frac{\varepsilon}{3} + K\frac{\varepsilon}{3K} = \varepsilon$. Since ε is arbitrary, $\mathcal{A}$ is FF-diagnosable. □

IF-diagnosability. IF-diagnosability focuses on infinite runs and FAmb_∞ is not *per se* a Borel set. Obtaining a characterisation is thus more difficult. Thanks to Theorem 3.1 however, in finitely-branching pLTS the above characterisation also holds for IF-diagnosability.

Corollary 3.2. *Let $\mathcal{A}$ be a finitely-branching pLTS. Then $\mathcal{A}$ is* IF*-diagnosable iff $\mathcal{A} \models \mathbb{P}^{=0}(\Diamond\Box(\mathfrak{f} \wedge \mathfrak{U}))$.*

IA-diagnosability. The assumption of finitely-branching pLTS is also needed in order to characterise IA-diagnosability. In addition to the path formula $\mathfrak{U}$ we also use the path formula $\mathfrak{W}$ defined in Example 3.6, page 78.

Proposition 3.8. *Let $\mathcal{A}$ be a finitely-branching pLTS. Then $\mathcal{A}$ is* IA*-diagnosable iff* $\mathcal{A} \models \mathbb{P}^{=0}(\Diamond\Box(\mathfrak{U} \wedge \mathfrak{W}))$.

Intuitively, as a faulty run cannot become correct, if a run does not satisfy $\mathfrak{U}$ once, then $\rho \models \Diamond\Box(\neg\mathfrak{U})$. Thus, $\rho \not\models \Diamond\Box(\mathfrak{U} \wedge \mathfrak{W})$ if either (1) it does not satisfy $\mathfrak{U}$ at least once, meaning ρ is surely faulty, or (2) does not satisfy $\mathfrak{W}$ infinitely often. In the latter case, the intuition is the following. Infinitely often firstf has increased. If there exists a faulty run ρ_f with $\mathcal{P}(\rho) = \rho_f$, then firstf is bounded, on the prefixes of $\mathcal{P}(\rho)$, by the length at which ρ_f becomes faulty. Thus no faulty run exists with observation $\mathcal{P}(\rho)$ and ρ is surely correct.

Proof. In order to prove formally the adequacy of the formula, it is enough to show that $\rho \in \Omega$ is ambiguous if and only if $\rho \models \Diamond\Box(\mathfrak{U} \wedge \mathfrak{W})$. We focus below on correct runs; the case of faulty runs can be treated in a similar, and even simpler, way.
• Let $\rho \in \mathsf{CAmb}_\infty$. Since ρ is ambiguous, there exists a faulty run ρ' such that $\mathcal{P}(\rho') = \mathcal{P}(\rho)$. Let k_0 be such that $\rho'_{\downarrow k_0}$ is faulty. Thus for all $k \geq k_0$, $\mathsf{firstf}(\mathcal{P}(\rho_{\downarrow k})) \leq k_0$ and in addition it is non-decreasing. So there exists some $k_1 \geq k_0$ such that for all $k \geq k_1$, $\mathsf{firstf}(\mathcal{P}(\rho_{\downarrow k}))$ is constant. We thus obtain $\rho \models \Diamond\Box\mathfrak{W}$. Moreover, since $\rho \models \Box\,\mathfrak{U}$, we conclude that $\rho \models \Diamond\Box(\mathfrak{U} \wedge \mathfrak{W})$.
• Conversely, let ρ be a correct run such that $\rho \models \Diamond\Box(\mathfrak{U} \wedge \mathfrak{W})$. Thus there is a position k_0 such that for all $k \geq k_0$, $\rho, k \models \mathfrak{W}$. In particular, by definition of $\mathfrak{W}$, for all $k \geq k_0$, there is a finite signalling run $\rho'^{(k)}$ such that $\mathcal{P}(\rho'^{(k)}) = \mathcal{P}(\rho_{\downarrow k})$ and $\rho'^{(k)}_{\downarrow k_0}$ is faulty. Consider the tree of these runs $\rho'^{(k)}$ by merging the common prefixes. This tree is finitely branching and infinite. By König's lemma, it must admit an infinite branch, corresponding to a run ρ' with $\mathcal{P}(\rho') = \mathcal{P}(\rho)$ and $\rho'_{\downarrow k_0}$ faulty. We deduce that ρ is ambiguous. □

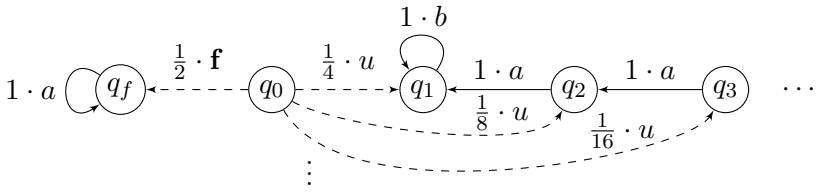


Figure 3.16: An infinitely-branching IA-diagnosable pLTS which does not satisfy $\mathbb{P}^{=0}(\Diamond\Box(\mathfrak{U} \wedge \mathfrak{W}))$.

The pLTS of Figure 3.16 illustrates the need for the finitely-branching assumption in Proposition 3.8. The set of unobservable events is $\{u, \mathbf{f}\}$. Observation b occurs in every infinite correct run, while the observed sequence of the single infinite faulty run is a^ω. This pLTS is thus IA-diagnosable. However, it does not satisfy $\mathbb{P}^{=0}(\Diamond\Box(\mathfrak{U} \wedge \mathfrak{W}))$ since the unique infinite faulty run has probability $\frac{1}{2}$ and satisfies at the same time

$\Box\mathfrak{W}$, by unicity, and $\Box\mathfrak{U}$. Indeed for every $n \in \mathbb{N}$, there is a correct signalling run with observed sequence a^n.

Observe that the sets of runs specified by the characterisations of FF-diagnosability ($\Diamond\Box(\mathfrak{f} \wedge \mathfrak{U})$) and IA-diagnosability ($\Diamond\Box(\mathfrak{U} \wedge \mathfrak{W})$) are F_σ sets, *i.e.* countable unions of closed sets.

3.3 Non-expressivity results

We chose to use a logic where the structural and probabilistic aspects are separated. This brought some advantages, but also weakens the expressibility. As a consequence, approximate notions of diagnosability for which the probabilities are important to define the ambiguity cannot be characterised. We prove this formally by showing that there is no Borel set E and F such that neither $\mathcal{A} \models \mathbb{P}^{=0}(E)$ nor $\mathcal{A} \models \mathbb{P}^{>0}(F)$ characterises 1/2FF-diagnosability.

Proposition 3.9. *There exists a finitely-branching LTS $\mathbb{A}$ such that for every Borel sets E and F of runs, there exists a pLTS $\mathcal{A} = (\mathbb{A}, \mathbf{P})$ such that:*

- *either $\mathcal{A}$ is 1/2FF-diagnosable and $\mathbb{P}_\mathcal{A}(E) > 0$;*
- *or $\mathcal{A}$ is not 1/2FF-diagnosable and $\mathbb{P}_\mathcal{A}(E) = 0$.*

and

- *either $\mathcal{A}$ is 1/2FF-diagnosable and $\mathbb{P}_\mathcal{A}(F) = 0$;*
- *or $\mathcal{A}$ is not 1/2FF-diagnosable and $\mathbb{P}_\mathcal{A}(F) > 0$.*

This proposition is proved by constructing a family of pLTS whose underlying LTS is the same, thus the Borel set we construct is the same for every member of this family. However the probabilities can be appropriately chosen in order for the pLTS to model or not the formula in contradiction with its diagnosability.

Proof. Consider the LTS $\mathbb{A} = \langle Q, q_0, \Sigma, T\rangle$ defined as follows and let the set of unobservable events be $\Sigma_u = \{\mathbf{f}, u\}$:

- $Q = \{q_0, q_f, q_c\}$,
- $\Sigma = \{a, b, \mathbf{f}, u\}$,
- $T = \{(q_0, u, q_c), (q_0, \mathbf{f}, q_f), (q_c, a, q_c), (q_c, b, q_c), (q_f, a, q_f), (q_f, b, q_f)\}$.

We consider a family of pLTS, represented in Figure 3.17 with underlying LTS $\mathbb{A}$. Given a pair of probabilities (p_1, p_2), we define the pLTS $\mathcal{A}_{(p_1,p_2)} = (\mathbb{A}, \mathbf{P}_{(p_1,p_2)})$ in which $\mathbf{P}_{(p_1,p_2)}(q_0, \mathbf{f}, q_f) = \mathbf{P}_{(p_1,p_2)}(q_0, u, q_c) = 1/2$, $\mathbf{P}_{(p_1,p_2)}(q_c, a, q_c) = p_1$, $\mathbf{P}_{(p_1,p_2)}(q_c, b, q_c) = 1 - p_1$, $\mathbf{P}_{(p_1,p_2)}(q_f, a, q_f) = p_2$ and $\mathbf{P}_{(p_1,p_2)}(q_f, b, q_f) = 1 - p_2$.

First note that $\mathcal{A}_{(p_1,p_2)}$ is 1/2FF-diagnosable iff $p_1 \neq p_2$. This can be established similarly to what was done in Example 3.5, page 68. In fact one can show that $\mathcal{A}_{(p_1,p_2)}$ is AFF-diagnosable iff $p_1 \neq p_2$.

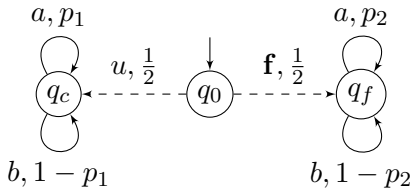


Figure 3.17: A family of pLTS whose underlying LTS has no appropriate characterisation for 1/2FF-diagnosability.

Let E be an arbitrary Borel set over the set of runs of $\mathbb{A}$. If there exists a probability value p such that $\mathcal{A}_{(p,p)} \models \mathbb{P}^{=0}(E)$ we have the first part of the result. Else, let p and p' be two probabilities with $p \neq p'$. As the probabilities of the runs of E in $\mathcal{A}_{(p,p')}$ do not depend simultaneously on p and on p', $\mathbb{P}_{\mathcal{A}_{(p,p')}}(E) + \mathbb{P}_{\mathcal{A}_{(p',p)}}(E) = \mathbb{P}_{\mathcal{A}_{(p,p)}}(E) + \mathbb{P}_{\mathcal{A}_{(p',p')}}(E)$. Moreover, for every probability p'', $\mathcal{A}_{(p'',p'')} \not\models \mathbb{P}^{=0}(E)$, thus either $\mathbb{P}_{\mathcal{A}_{(p,p')}}(E) > 0$ or $\mathbb{P}_{\mathcal{A}_{(p',p)}}(E) > 0$ which concludes the first part of the proof.

Let F be an arbitrary Borel set. If there exists a probability p such that $\mathcal{A}_{(p,p)} \models \mathbb{P}^{>0}(E)$ we have the first part of the result. Else, let p and p' be two probabilities with $p \neq p'$. As before, $\mathbb{P}_{\mathcal{A}_{(p,p')}}(F) + \mathbb{P}_{\mathcal{A}_{(p',p)}}(F) = \mathbb{P}_{\mathcal{A}_{(p,p)}}(F) + \mathbb{P}_{\mathcal{A}_{(p',p')}}(F) = 0$. Thus both $\mathcal{A}_{(p,p')} \models \mathbb{P}^{=0}(F)$ and $\mathcal{A}_{(p',p)} \models \mathbb{P}^{=0}(E)$ which concludes the second part of the proof. □

For exact notions of diagnosability, intuitively, FA-diagnosability would be in between FF-diagnosability and IA-diagnosability in terms of complexity. Surprisingly we showed that FA-diagnosability does not admit such a characterisation: there is no F_σ set E such that a pLTS $\mathcal{A}$ is FA-diagnosable if and only if $\mathcal{A} \models \mathbb{P}^{=0}(E)$.

Proposition 3.10. *There exists a finitely-branching infinite LTS $\mathbb{A}$ such that for every F_σ set E of runs, there exists a pLTS $\mathcal{A} = (\mathbb{A}, \mathbf{P})$ such that:*

- *either $\mathcal{A}$ is* FA*-diagnosable and $\mathbb{P}_{\mathcal{A}}(E) > 0$;*
- *or $\mathcal{A}$ is not* FA*-diagnosable and $\mathbb{P}_{\mathcal{A}}(E) = 0$.*

This proposition is proved in a similar fashion as Proposition 3.9, it is only more involved. The family of pLTS we construct has infinitely many states and infinitely many parametric probabilities. These probabilities can be chosen in order to give a positive value to either the limit of the probability of CAmb_n or to the given F_σ set.

Proof. Consider the LTS $\mathbb{A} = \langle Q, q_0, \Sigma, T \rangle$ defined as follows:

- $Q = \{f_1, q_f\} \cup \{q_i \mid i \in \mathbb{N}\}$;
- $\Sigma = \{a, b, c, u, \mathbf{f}\}$;
- $T = \{(q_0, u, q_f), (q_0, u, q_1), (q_f, a, q_f), (q_f, b, q_f), (q_f, \mathbf{f}, f_1), (f_1, b, f_1), (f_1, c, f_1)\}$ $\cup \{(q_i, a, q_{i+1}), (q_i, b, q_{i+1})\}_{i \geq 1}$;

- $\Sigma_u = \{\mathbf{f}, u\}$.

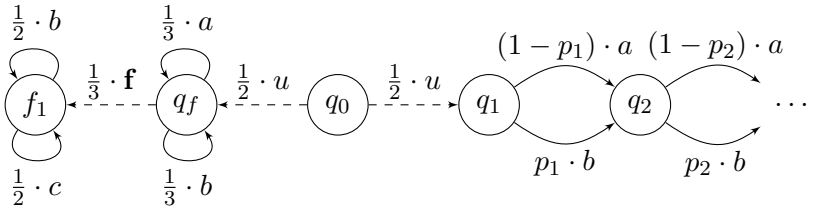


Figure 3.18: A family of pLTS whose underlying LTS has no appropriate characterisation of FA-diagnosability.

We consider a family of pLTS, represented in Figure 3.18, with underlying LTS $\mathbb{A}$. For $\mathbf{p} = (p_n)_{n\geq 1}$ a sequence of probabilities, we define the pLTS $\mathcal{A}_\mathbf{p} = (\mathbb{A}, \mathbf{P}_\mathbf{p})$ in which for every $n \geq 1$ the probability that 'b' occurs from state q_n is $\mathbf{P}_\mathbf{p}(q_n, b, q_{n+1}) = p_n$, the probability that '$a$' occurs from state q_n is $\mathbf{P}_\mathbf{p}(q_n, a, q_{n+1}) = 1 - p_n$ and all other probabilities are independent of $\mathbf{p}$: $\mathbf{P}_\mathbf{p}(q_0, u, q_f) = \mathbf{P}_\mathbf{p}(q_0, u, q_1) = \mathbf{P}_\mathbf{p}(f_1, b, f_1) = \mathbf{P}_\mathbf{p}(f_1, c, f_1) = \frac{1}{2}$, $\mathbf{P}_\mathbf{p}(q_f, a, q_f) = \mathbf{P}_\mathbf{p}(q_f, b, q_f) = \mathbf{P}_\mathbf{p}(q_f, \mathbf{f}, f_1) = \frac{1}{3}$.
Observe that a faulty run almost surely produces a 'c', so that $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = 0$. Moreover, as the ambiguous runs are the ones ending by a 'b', the probability to be ambiguous after n observations on the leftmost part is $\frac{1}{2}\frac{2^{n-1}}{3^n}$ and on the rightmost part is $\frac{1}{2}p_n$ $\mathbb{P}(\mathsf{CAmb}_n) = p_n + \frac{2^{n-1}}{3^n}$. Therefore, $\mathcal{A}_\mathbf{p}$ is FA-diagnosable iff $\lim_{n\to\infty} p_n = 0$.

Let E be an arbitrary F_σ set. We pick some FA-diagnosable $\mathcal{A}_\mathbf{p}$ *i.e.* such that $\lim_{n\to\infty} p_n = 0$ and write $\mathbb{P}_\mathbf{p}$ for the probability measure it induces. If $\mathbb{P}_\mathbf{p}(E) > 0$ we are done. Assume thus that $\mathbb{P}_\mathbf{p}(E) = 0$. In order to define a second pLTS, via $\mathbf{p}'$, consider an infinite increasing sequence $\{n_j\}_{j\leq 1}$ and let for $n \notin \{n_j\}_{j\leq 1}$, $p'_n = p_n$ and for $n \in \{n_j\}_{j\geq 1}$, $p'_n = \frac{1}{2}$. Due to the sub-sequence $p'_{n_j} = \frac{1}{2}$, $\mathcal{A}_{\mathbf{p}'}$ is not FA-diagnosable. The sequence $\{n_j\}_{j\leq 1}$ depends on $\mathbb{P}_\mathbf{p}$ and will be defined after some preliminary observations.
Let $F = \{\rho \mid q_0 u q_1 \preceq \rho\}$. Denoting $\mathbb{P}_{\mathbf{p}'}$ the probability measure of the second pLTS, observe that $\mathbb{P}_{\mathbf{p}'}(E \setminus F) = \mathbb{P}_\mathbf{p}(E \setminus F) = 0$. Using the above discussion, the F_σ set $E \cap F = \bigcup_{m\in\mathbb{N}} \bigcap_{n\in\mathbb{N}} O_{m,n}$ where for all m, n, $O_{m,n}$ is a disjoint union of cylinders $C(\rho)$ with $|\rho| = n$, $O_{m,n+1} \subseteq O_{m,n}$ and $O_{m,n} \subseteq O_{m+1,n}$. Denote $F_m = \bigcap_{n\in\mathbb{N}} O_{m,n}$ For all m, $\lim_{n\to\infty} \mathbb{P}_\mathbf{p}(O_{m,n}) = \mathbb{P}_\mathbf{p}(E \cap F_m) \leq \mathbb{P}_\mathbf{p}(E \cap F) = 0$.
• n_1 is chosen such that for all $n \geq n_1$, $p_n \leq \frac{1}{2}$. Observe now that for all n_j,

$$p'_{n_j} = \frac{1}{2} = \frac{1}{2p_{n_j}} p_{n_j} \text{ and } 1 - p'_{n_j} = \frac{1}{2} \leq 1 - p_{n_j} \leq \frac{1}{2p_{n_j}}(1 - p_{n_j}).$$

By definition of $\mathbf{P}_{\mathbf{p}'}$, since $O_{m,n}$ is a disjoint union of cylinders $C(\rho)$ with $|\rho| = n$, applying inductively the previous inequalities, for all n such that $n_k < n \leq n_{k+1}$ (denoting $n_0 = 0$):

$$\mathbb{P}_{\mathbf{p}'}(O_{m,n}) \leq \frac{\mathbb{P}_\mathbf{p}(O_{m,n})}{2^k \prod_{1\leq j<k} p_{n_j}} \quad . \tag{3.1}$$

• Assume that we have chosen $n_1, \ldots, n_k$. Since $\lim_{n\to\infty} \mathbb{P}_{\mathbf{p}}(O_{k,n}) = 0$, there exists $n_{k+1} > n_k$ such that $\mathbb{P}_{\mathbf{p}}(O_{k,n_{k+1}}) \leq \prod_{1\leq j\leq k} p_{n_j}$. We choose such an index.
Equation (3.1) now implies that for all $m \leq k$,

$$\mathbb{P}_{\mathbf{p}'}(O_{m,n_{k+1}}) \leq \mathbb{P}_{\mathbf{p}'}(O_{k,n_{k+1}}) \leq \frac{1}{2^k}.$$

Thus for all m,

$$\mathbb{P}_{\mathbf{p}'}(F_m) = \lim_{k\to\infty} \mathbb{P}_{\mathbf{p}'}(O_{m,n_{k+1}}) = 0.$$

Since $E \cap F = \bigcup_{m\in\mathbb{N}} F_m$, $\mathbb{P}_{\mathbf{p}'}(E \cap F) = 0$ and so $\mathbb{P}_{\mathbf{p}'}(E) = 0$. □

Beyond Proposition 3.10, we conjecture that the impossibility also holds for arbitrary Borel sets.

Proposition 3.10 only shows the non-existence for characterisations that requires a null probability of the given set. There could thus still exists a characterisation asking for a positive probability. In fact, such a characterisation does not exist. This impossibility is even stronger than the one of Proposition 3.10 as we show that a positive probability characterization cannot exist whatever the Borel set (and not only F_σ).

Proposition 3.11. *There exists a finitely-branching LTS $\mathbb{A}$ such that for every Borel set E of runs, there exists a pLTS $\mathcal{A} = ((\mathbb{A}, \mathbf{P}), \Sigma_o, \mathcal{P})$ such that:*

- *either $\mathcal{A}$ is* FA*-diagnosable and $\mathbb{P}_{\mathcal{A}}(E) = 0$;*
- *or $\mathcal{A}$ is not* FA*-diagnosable and $\mathbb{P}_{\mathcal{A}}(E) > 0$.*

This proof is similar to the previous one, albeit with an even more complex family of pLTS. Indeed, here, instead of having a parametric probability p_i for every $i \in \mathbb{N}$ we have one for every word $w \in \{a, b\}^*$.

Proof. Consider the LTS $\mathbb{A} = \langle Q, q_0, \Sigma, T\rangle$ defined as follows:

- $Q = \{f_1, q_f, q_0\} \cup \{q_w \mid w \in (a + b)^*\}$;
- $\Sigma = \{a, b, c, u, \mathbf{f}\}$;
- $T = \{(q_0, u, q_f), (q_0, u, q_1), (q_f, a, q_f), (q_f, b, q_f), (q_f, \mathbf{f}, f_1), (f_1, b, f_1), (f_1, c, f_1)\}$
  $\cup \{(q_w, a, q_{wa}), (q_w, b, q_{wb})\}_{w\in(a+b)^*}$;
- $\Sigma_u = \{\mathbf{f}, u\}$.

We consider a family of pLTS, represented in Figure 3.19, with underlying LTS $\mathbb{A}$, parametrised by a mapping $\mathbf{p} : (a + b)^* \to (0, 1)$. Let $\mathcal{A}_{\mathbf{p}} = ((\mathbb{A}, \mathbf{P}_{\mathbf{p}}), \Sigma_o, \mathcal{P})$ be the pLTS such that the probability that 'b' occurs from state q_w is $\mathbf{P}(q_w, b, q_{wb}) = \mathbf{p}(w)$, the probability that '$a$' occurs from state q_w is $\mathbf{P}(q_w, b, q_{wa}) = 1 - \mathbf{p}(w)$ and all other probabilities are independent from $\mathbf{p}$: $\mathbf{P}_{\mathbf{p}}(q_0, u, q_f) = \mathbf{P}_{\mathbf{p}}(q_0, u, q_1) = \mathbf{P}_{\mathbf{p}}(f_1, b, f_1) = \mathbf{P}_{\mathbf{p}}(f_1, c, f_1) = \frac{1}{2}$, $\mathbf{P}_{\mathbf{p}}(q_f, a, q_f) = \mathbf{P}_{\mathbf{p}}(q_f, b, q_f) = \mathbf{P}_{\mathbf{p}}(q_f, \mathbf{f}, f_1) = \frac{1}{3}$. In the sequel, for

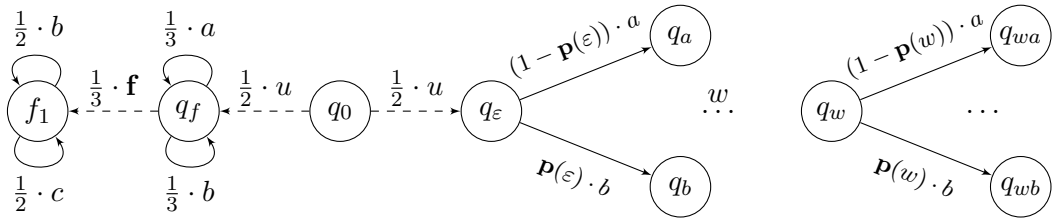


Figure 3.19: Another family of pLTS whose underlying LTS has no appropriate characterisation of FA-diagnosability.

convenience, we also write $\mathbf{p}(w,b)$ for $\mathbf{p}(w)$, and define $\mathbf{p}(w,a) = 1 - \mathbf{p}(w)$, so that $\mathbf{P}(q_w, a, q_{wa}) = \mathbf{p}(w,a)$.
Word w can be decomposed into letters $w = w[1]\dots w[n]$, and we give notations for factors: $w[1,k] = w[1]\dots w[k]$ with the convention that $w[1,0] = \varepsilon$. Finally we define $p_{\mathbf{p}}(w) = \prod_{1\leq k\leq n} \mathbf{p}(w[1,k-1], w[k])$, as the probability to read w from q_ε. Since $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = 0$ and $\mathbb{P}(\mathsf{CAmb}_{n-1}) = \sum_{w||w|=n-1} \mathbf{p}(w,b) + \frac{2^{n-1}}{3^n}$, we deduce that $\mathcal{A}_{\mathbf{p}}$ is FA-diagnosable iff $\lim_{n\to\infty} \sum_{|w|=n-1} \mathbf{p}(w,b) = 0$.

Let E be an arbitrary measurable set. Pick some pLTS $\mathcal{A}_{\mathbf{p}}$ which is FA-diagnosable, *i.e.* with $\lim_{n\to\infty} \sum_{|w|=n-1} \mathbf{p}(w,b) = 0$. If $\mathbb{P}_{\mathbf{p}}(E) = 0$ where $\mathbb{P}_{\mathbf{p}}$ is the probability of this pLTS, we are done. Assume therefore that $\mathbb{P}_{\mathbf{p}}(E) > 0$. Let $F = \{\rho \mid q_0 u q_\varepsilon \sqsubseteq \rho\}$ be the set of runs starting with a u-transition to q_ε. Denoting $\mathbb{P}_{\mathbf{p}'}$ the probability measure of any other pLTS $\mathcal{A}_{\mathbf{p}'}$, observe that $\mathbb{P}_{\mathbf{p}'}(E \setminus F) = \mathbb{P}_{\mathbf{p}}(E \setminus F)$. So, if $\mathbb{P}_{\mathbf{p}}(E \setminus F) > 0$, then by picking any non FA-diagnosable $(\mathbb{A}, \mathbf{P}_{\mathbf{p}'})$, we are done. So assume $\mathbb{P}_{\mathbf{p}}(E \setminus F) = 0$ which implies $\mathbb{P}_{\mathbf{p}}(E \cap F) > 0$. The probability being an inner regular measure (recall Definition 2.4, page 37), there exists a closed set $G \subseteq E \cap F$ with $\mathbb{P}_{\mathbf{p}}(G) > 0$.

If $G = F$ then $\mathbb{P}_{\mathbf{p}'}(G) = \mathbb{P}_{\mathbf{p}}(G) = \frac{1}{2}$. In this case, we can therefore conclude by picking any non FA-diagnosable pLTS $\mathcal{A}_{\mathbf{p}'}$.

Assuming $G \subsetneq F$, since G is closed, there is some cylinder $C(\rho)$ with $\rho = q_0 u q_\varepsilon \dots q_w$ such that $G \cap C(\rho) = \emptyset$. Then we define the pLTS $\mathcal{A}_{\mathbf{p}'}$ as the pLTS $\mathcal{A}_{\mathbf{p}}$ except that for every $w \preceq w'$ and every $x \in \{a,b\}$, $\mathbf{p}'(w',x) = \frac{1}{2}$. Thus for every $n \geq |w|$, $\sum_{|w'|=n} \mathbf{p}'(w',b) \geq \frac{\mathbf{P}_{\mathbf{p}}(\rho)}{2}$. So $\mathcal{A}_{\mathbf{p}'}$ is not FA-diagnosable. On the other hand, $\mathbb{P}_{\mathbf{p}'}(E \cap F) \geq \mathbb{P}_{\mathbf{p}'}(G) = \mathbb{P}_{\mathbf{p}}(G) > 0$. □

With Proposition 3.10 and Proposition 3.11 FA-diagnosability appears to be the most complicated of the exact diagnosability notions. Indeed, it cannot be characterised by any set of at least the first two level of the Borel hierarchy contrary to the other notions. This confirms the intuition that was raised when defining the diagnosers associated with each notion of diagnosability. For approximate notions of diagnosability, the non-expressibility result of Proposition 3.9 clearly shows that studying these notions requires a characterisation that intertwines structural and probabilistic elements.

4 Conclusion

In Section 1, appropriate diagnosers were associated with notions of diagnosability. This gave another point of view on the notions of diagnosability and raised the question of the memory that is necessary for a diagnoser. We showed that for approximate notions of diagnosability, unbounded memory may be necessary even for finite systems. However for exact notions of diagnosability, every example we gave used finite memory. In Chapter 4, we study diagnosability of finite pLTS and establish what memory is needed for each notion of exact diagnosability.

In Section 2, the links between the various diagnosability notions were established. This showed multiple interesting facts. In non-probabilistic systems, diagnosability only focuses on faulty runs as, for finitely-branching systems, if there is an infinite faulty ambiguous run, there also exists an infinite correct ambiguous run. However, when probabilistic models are considered, this symmetry is broken. Caring about both faulty and correct runs make for an entirely new notion. Moreover, probabilities do not affect correct and faulty runs in the same way concerning ambiguity: while IF- and FF-diagnosability are equivalent for finitely-branching systems, IA- and FA-diagnosability are not equivalent even for finite systems. Another interesting point, the uniformity requirement is not as important for exact notions of diagnosis than for approximate ones. Last, while AFF-diagnosability could appear to be close to FF-diagnosability as it forces an arbitrary high accuracy, the two notions are still very different.

In Section 3, characterisations composed of a probabilistic requirement on a set of paths defined by a pathL formula were established, when possible, for the notions of diagnosability. The notions which could not be characterised where the approximate notions of diagnosability (this is due to the sort of characterisation we were looking for) and FA-diagnosability. The latter is quite surprising as the definition of FA-diagnosability seems to be in between the ones of FF- and IA-diagnosability. Using model-checking techniques, one could derive an algorithm to decide a notion characterised by a logical formula. The absence of characterisation for approximate diagnosability and FA-diagnosability does not however mean that there is no algorithm for these problems. Indeed, we show in Chapter 4 that for finite pLTS, one can give a more specific characterisation of the exact diagnosability notions, including the FA-diagnosability. These characterisations have the same descriptive complexity and are used to obtain an algorithm. In Chapter 5, we study diagnosability for infinite-state systems and base our approach on the characterisations proved in the current section.

Chapter 4

Algorithmic analysis of the diagnosability of finite pLTS

In Chapter 3, we studied the notions of diagnosability in a general setting. More precisely, we showed that a diagnoser could be associated with each notion of diagnosability, we established the links between the different diagnosability definitions and gave a logical characterisation of multiple versions of exact diagnosability. With additional assumptions, stronger results can be established: characterisations can be refined and efficient algorithms can be designed. One of the usual restriction is the finiteness of systems. Many real systems can be represented with finitely many states (a vending machine, a Pac-Man game, ...). In fact, most of the systems which interaction with the environment only requires to keep in memory a finite number of events would fit in such a framework. For example, let us consider a server that receives and processes requests. If the server uses a finite memory, then after memorising a fixed number of yet unprocessed requests, the next request might be discarded without being processed due to a stack overflow. Such servers can be represented with finitely many states. Servers able to memorise an unbounded number of requests is dealt with in Chapter 5. When considering finite systems, we immediately get some results as consequences of the ones of Chapter 3. For example, any finite system being finitely branching, according to Theorem 3.1, page 73, IF-diagnosability and FF-diagnosability, two diagnosability notions focused only on faulty runs, are equivalent. As another example, every logical characterisation given in Section 3 of Chapter 3 applies.

The first step is to establish what additional results can be obtained for diagnosability when we only study systems with finitely many states. These systems can easily be represented and have important properties that do not hold in the general case. For example, there is a finite number of probability values in the system since there is a finite number of transitions. This is not true in the general case as can be seen in the example of Figure 3.11, page 72. Using these additional properties, we establish new characterisations of the diagnosability notions in Section 1. These characterisations are not given as probabilistic logical formulae but as conditions on the structure or on the probabilities of the language observed in the system. More precisely, the characterisa-

tions of the exact diagnosability notions are purely structural (*i.e* the probability values do not matter): they rely on a variant of the determinisation of the system. For the approximate diagnosability notions, the probabilities matter, and cannot be removed from the characterisation. The given characterisation relies on the comparisons of the probability of sets of observed sequences that can be observed from pairs of states.

Once the characterisations are established, we apply them to get as many algorithmic and complexity results as possible. One such application, is to provide an algorithm to decide the diagnosability of a system. Indeed, given a "simple" characterisation, one can easily check if it holds on a given system. Checking a characterisation can be done with many different algorithms, some being more efficient than others. The efficiency of an algorithm is described by the complexity class the problem belongs to, in computational complexity theory. When possible we provide algorithms that are optimal with respect to the standard computational classes. For the other diagnosability problems, we establish that the associated decision problem is undecidable. This is done in Section 2.

When a system is diagnosable, there exists a diagnoser for this system. The possible diagnosers use different amount of memory, give their verdict more or less quickly, etc. In Section 3, we show how to automatically construct a diagnoser for systems that are exactly diagnosable. The diagnosers we build have optimal memory and give their verdict as soon as possible. The constructions of the exact diagnosers are based on the characterisations given in Section 1. This strengthens the importance of these characterisations.

This chapter develops and extends some of the results from [BHL14, BHL16a].

1 Characterisations of diagnosability

In this section, we establish characterisations for the different notions of exact diagnosability and for one notion of approximate diagnosability. These characterisations strongly rely on the restriction to finite state systems. Therefore they are easier to express and check, but in general cannot be adapted to more general cases as will be seen in Chapter 5. As a direct consequence of the finite-state restriction and of Theorem 3.1, page 73, FF-diagnosability and IF-diagnosability coincide. So we only consider FF-diagnosability in the rest of this chapter.

For all the exact diagnosability notions, the methodology is similar. We first construct an *ad hoc* deterministic automaton which gathers all the information needed for the diagnosis, by tracking possible correct and faulty executions. Secondly, we build the product of the original pLTS with this deterministic automaton[1]. Diagnosability can then be characterised on the product by graph-based properties.

For approximate diagnosability, we show that the diagnosability notions can be characterised relying on the distance 1 problem for labelled Markov chains (LMC). This problem, recalled in Chapter 2, receives as input two LMC and asks for the existence

[1]Using such a product to enrich the initial model was mentioned as an usual technique page 29. There, the deterministic automaton was called belief automaton.

of an event, that is almost sure in one LMC, and has null measure in the other. It was shown to be decidable in PTIME [CK14].

1.1 Exact diagnosis

The deterministic automata we build are variants of the deterministic Büchi automaton introduced in [HHMS13], that accepts the infinite unambiguous observed sequences. The latter tracks the subsets of possible states reached by signalling runs associated with an observed sequence. It looks like the on-the-fly determinisation of $\mathcal{A}$ viewing unobserved events as silent transitions. However, in view of the forthcoming characterisations, the subsets of correct and faulty states are divided in three sets: U, V and W. The intuitive meaning of these sets is the following one:

- A state q belongs to U, if there is a correct signalling run with the current observed sequence ending in q;
- A state q belongs to $V \cup W$ if there is a faulty signalling run with the current observed sequence ending in q.
- The partition between V and W ensures that for all $q \in V$, $q' \in W$ and ρ a faulty run ending in q, there exists a faulty run ρ' ending in q' with an earlier fault than the fault of ρ. In other words, V and W contain the states reached by faulty runs, while W keeps track of the runs that have been faulty for the longest time.

W corresponds to the set of faulty states for which the ambiguity with the correct states of U has to be resolved (when both are not empty), while V corresponds to a waiting room of states reached by faulty runs that will be examined when the current ambiguity is resolved.

Before giving the definition of the deterministic automaton associated with a pLTS and for sake of readability, we define for two sets of states U and V and an observation $a \in \Sigma_o$ the set of states

$$\begin{aligned}\mathsf{updatefaulty}(U, V, a) = \{q \mid \exists q' \in U, q' \Rightarrow_f^a q\} \\ \cup \{q \mid \exists q' \in V, q' \Rightarrow^a q\}.\end{aligned}$$

This set contains the states reached from U by a faulty signalling run of observation a and the ones reached from V by a signalling run of observation a.

Definition 4.1. *Given a pLTS $\mathcal{A}$, the deterministic automaton associated with $\mathcal{A}$ is $\mathsf{Obs}(\mathcal{A}) = \{S, s_0, \Delta, F\}$ where*

- *$s_0 = (\{q_0\}, \emptyset, \emptyset)$ is the initial state of $\mathsf{Obs}(\mathcal{A})$;*
- *the states and transitions of the deterministic Büchi automaton $\mathsf{Obs}(\mathcal{A})$ are inductively defined by:*

  Given (U, V, W) a state of $\mathsf{Obs}(\mathcal{A})$ and $a \in \Sigma_o$, there exists a state $(U', V', W') \in S$ and a transition $(U, V, W) \xrightarrow{a} (U', V', W')$ in Δ as soon as:

1. $\{q \in U \cup V \cup W \mid q \Rightarrow^a\} \neq \emptyset$,
2. $U' = \{q \mid \exists q' \in U, q' \Rightarrow_c^a q\}$,
3. *If* $W = \emptyset$ *then* $V' = \emptyset$ *and* $W' = \mathsf{updatefaulty}(U, V, a)$,
4. *If* $W \neq \emptyset$ *then* $W' = \mathsf{updatefaulty}(\emptyset, W, a)$ *and* $V' = \mathsf{updatefaulty}(U, V, a) \setminus W'$;

- *the set* F *of accepting states consists of all triples* (U, V, W) *with* $U = \emptyset$ *or* $W = \emptyset$.

When $U = \emptyset$, the current signalling run is surely faulty, since U tracks the possible states after a correct run. When $W = \emptyset$ the current signalling run may be ambiguous (if $V \neq \emptyset$) but the "oldest" possible faulty runs under scrutiny have been discarded. Hence, any infinite observed sequence of $\mathcal{A}$ passing infinitely often through F is not ambiguous (either it is surely faulty, or ambiguities are resolved one after another).

Example 4.1. *Let* $\mathcal{A}$ *be the pLTS of Figure 4.1. We represent the associated deterministic automaton* $\mathsf{Obs}(\mathcal{A})$ *in Figure 4.2, where accepting states for the Büchi condition are doubly framed.*

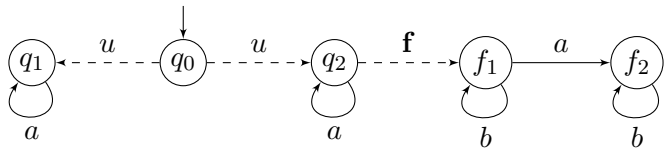


Figure 4.1: An IA and FF-diagnosable pLTS which is not FA-diagnosable.

Let us consider the observed sequence a^ω *which has probability* $1/2$ *in the pLTS. The initial state of* $\mathsf{Obs}(\mathcal{A})$ *is* $(\{q_0\}, \emptyset, \emptyset\}$ *meaning that the initial state of the pLTS is* q_0 *and no fault occurred so far. Then, observing 'a' we reach* $(\{q_1, q_2\}, \emptyset, \{f_2\})$ *meaning that from* q_0*, we can reach* q_1 *and* q_2 *with correct signalling run of observation 'a' and* f_2 *with a faulty signalling run of observation 'a'. As this faulty run is the oldest one, it is added to* W*. The second observed 'a' leads to the state* $(\{q_1, q_2\}, \{f_2\}, \emptyset)$*. The run that ended in* f_2 *previously cannot be extended by observing an 'a', thus the set* W *which tracked this run (as the only oldest faulty run) is empty. However a new fault can be created ending in state* f_2*, this fault being made while* W *was tracking other faults, it is not considered "old" and thus is added to* V*. This state is therefore an accepting state. Observing yet another 'a' leads back to* $(\{q_1, q_2\}, \emptyset, \{f_2\})$*. The run tracked by* V *disappeared, a new fault was created and as* W *was empty it is considered a "old" fault and* f_2 *is put in* W*. As the path of* $\mathsf{Obs}(\mathcal{A})$ *corresponding to the observed sequence* a^ω *alternates between these two states infinitely often, it visits infinitely often a state where* $W = \emptyset$*, thus* a^ω *is accepted.*

For the other observed sequences, a 'b' can only be observed in a faulty state, therefore any observed sequence containing a 'b' is surely faulty. This can be seen in $\mathsf{Obs}(\mathcal{A})$ *as any path containing a 'b' ends in one of the five rightmost states, all of which have an empty set* U *(the first component corresponding to the correct reachable states). As*

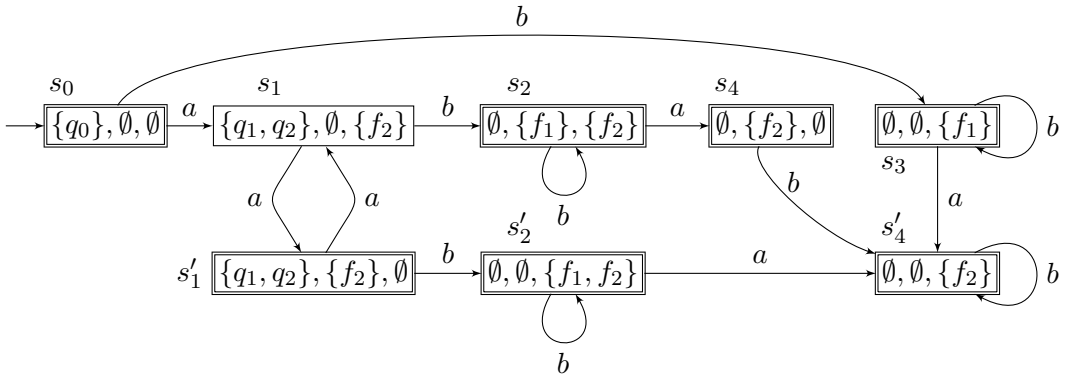


Figure 4.2: The deterministic automaton associated with the pLTS of Figure 4.1. The states that are framed twice are accepting for the Büchi condition.

any state with $U = \emptyset$ is accepting, every infinite observed sequence containing a 'b' is accepted.

The next proposition recalls the main property of this automaton.

Proposition 4.1 ([HHMS13])**.** *Let $\mathcal{A}$ be a finite pLTS. Then the deterministic Büchi automaton $\mathsf{Obs}(\mathcal{A})$ accepts exactly the infinite unambiguous observed sequences of $\mathcal{A}$.*

Example 4.2. *As seen in Example 4.1, every observed sequence of the pLTS of Figure 4.1 is accepted by the associated deterministic automaton. According to Proposition 4.1, this means that there is no infinite ambiguous sequence in this pLTS, thus it is* FF- *and* FA*-diagnosable.*

1.1.1 FF-diagnosability

As explained earlier, for each diagnosability notion, we consider a variant of $\mathsf{Obs}(\mathcal{A})$. For FF-diagnosability, we only need to remove the ambiguity for faulty runs So we can omit the faulty sets of states V and W. We write $\mathsf{FF}(\mathcal{A})$ for the resulting simplified automaton, called FF*-automaton*, obtained from $\mathsf{Obs}(\mathcal{A})$ by only considering the U-component of states.

Example 4.3. *Figure 4.3 illustrates this construction on the pLTS of Figure 4.1. This automaton reflects that once b happens, the current signalling run is surely faulty. Thus the set of possible correct states is empty (state s_2).*

To recover the stochastic behaviour of $\mathcal{A}$ which is not reflected in $\mathsf{FF}(\mathcal{A})$, we now define the pLTS $\mathcal{A}_{\mathsf{FF}} = \mathcal{A} \times \mathsf{FF}(\mathcal{A})$ as the product of $\mathcal{A}$ and $\mathsf{FF}(\mathcal{A})$ synchronised over observed events.

Definition 4.2. *Given a pLTS $\mathcal{A} = \langle Q, q_0, \Sigma, T\rangle$ associated with the* FF*-automaton $\mathsf{FF}(\mathcal{A}) = \{S, s_0, \Delta, F\}$, we define $\mathcal{A}_{\mathsf{FF}} = \langle Q', (q_0, \{q_0\}), \Sigma, T'\rangle$ where:*

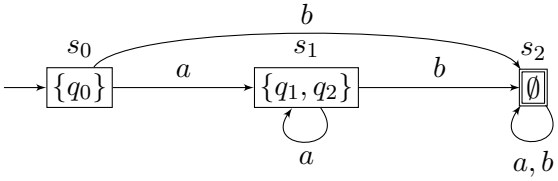


Figure 4.3: The FF-automaton of pLTS of Figure 4.1.

- $Q' = Q \times S$;
- $((q, B), a, (q', B)) \in T'$ *iff* $(q, a, q') \in T$ *and*
  - *either* $a \in \Sigma_u$ *and* $B = B'$,
  - *or* $a \in \Sigma_o$ *and there exists a transition* $B \xrightarrow{a} B'$ *in* Δ.

Since $\mathsf{FF}(\mathcal{A})$ is deterministic and complete, $\mathcal{A}_{\mathsf{FF}}$ is still a pLTS, with the same stochastic behaviour as $\mathcal{A}$. More precisely, there is a bijection between the runs of $\mathcal{A}$ and the runs of $\mathsf{FF}(\mathcal{A})$. A run and its image by the bijection have the same observation and the same probability. In addition, the U-component of a state (q, U) of $\mathcal{A}_{\mathsf{FF}}$ stores the relevant information w.r.t FF-diagnosability of the observed sequence so far.

Example 4.4. *Carrying on with the example pLTS of Figure 4.1, Figure 4.4 shows the resulting product pLTS. Observe that it has two bottom strongly connected components (BSCC), consisting of the absorbing states* (q_1, s_1) *and* (f_2, s_2).

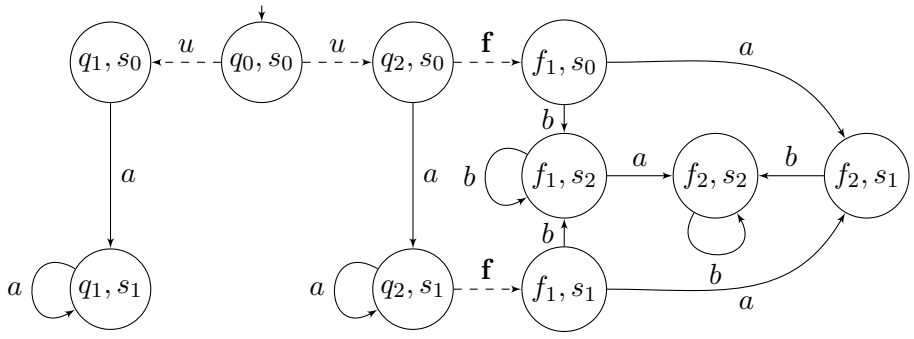


Figure 4.4: The synchronised product of the pLTS of Figure 4.1 and its FF-automaton.

In a finite pLTS almost all runs ends in a BSCC [BK08], and FF-diagnosability is a property of runs expressed as an almost sure event. So, the characterisation of FF-diagnosability can be stated on the BSCC of $\mathcal{A}_{\mathsf{FF}}$.

Proposition 4.2. *Let* $\mathcal{A}$ *be a finite pLTS. Then* $\mathcal{A}$ *is* FF*-diagnosable if and only if* $\mathcal{A}_{\mathsf{FF}}$ *has no BSCC containing a state* (q, U) *with* $q \in Q_f$ *and* $U \neq \emptyset$.

The proof relies on two elements. First, a run will almost surely reach a BSCC of the pLTS. Second, for a faulty run to be ambiguous, the set U contained in the state in which the run ends must not be empty. Therefore if every BSCC corresponds to either correct runs (*i.e.* $q \in Q_c$) or faulty unambiguous runs (*i.e.* $U = \emptyset$), the pLTS is FF-diagnosable.

Proof. Suppose first that there exists a reachable BSCC C of $\mathcal{A}_{\mathsf{FF}}$ and a state $s = (q, U)$ in C such that $q \in Q_f$ and $U \neq \emptyset$. Let ρ be a signalling run leading from the initial state s_0 of $\mathcal{A}_{\mathsf{FF}}$ to s. Now, for every state $s' = (q', U') \in C$, necessarily $q' \in Q_f$ and $U' \neq \emptyset$, because C is strongly connected[2]. So for every signalling run ρ' that extends ρ, writing $s' = (q', U')$ for the state ρ' leads to, there exists a correct signalling run ρ'' such that $\mathcal{P}(\rho'') = \mathcal{P}(\rho')$ and $q_0 \xrightarrow{\rho''} q''$ with $q'' \in U'$. As a consequence the observed sequence $\mathcal{P}(\rho'')$ is ambiguous in $\mathcal{A}_{\mathsf{FF}}$, and for every $n \geq |\rho|_o$, $\mathbb{P}(\mathsf{FAmb}_n) \geq \mathbb{P}(\rho)$. As $\mathcal{A}$ and $\mathcal{A}_{\mathsf{FF}}$ have the same ambiguous observed sequences and the associated runs have the same probabilities, $\mathcal{A}$ is not FF-diagnosable.

Suppose now that for every state $s = (q, U)$ of a BSCC C, either $q \in Q_c$, or $U = \emptyset$. This property is in fact uniform by BSCC: for every BSCC C, either for every state $(q, U) \in C$, $q \in Q_c$, or, for every state $(q, U) \in C$, $U = \emptyset$. This is a straightforward consequence of C being strongly connected. Moreover, if a run ρ reaches a pair (q, U) then $q \in Q_c$ implies $U \neq \emptyset$. Indeed, let ρ' be the greatest signalling run prefix of ρ. ρ' ends in a pair (q', U') where $U' = U$ as $\mathcal{P}(\rho) = \mathcal{P}(\rho')$. Moreover if $q \in Q_c$, then $q' \in Q_c$, therefore $q' \in U$ implying that $U \neq \emptyset$. Therefore in $\mathcal{A}_{\mathsf{FF}}$ the BSCC are partitioned in correct ones, in case all q-components of states in C are correct, and faulty ones, in case all U-components of states in C are empty ensuring unambiguity of faulty runs ending in a BSCC. Since runs almost surely leave the transient states and reach a BSCC, this implies that $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = 0$. □

As a consequence of this characterisation, we establish the equivalence between FF- and uniform FF-diagnosability for finite pLTS, claimed in Theorem 3.1.

Proposition 4.3. *Let $\mathcal{A}$ be a finite pLTS. If $\mathcal{A}$ is FF-diagnosable, then it is uniformly FF-diagnosable.*

In a finite FF-diagnosable pLTS, thanks to the characterisation given in Proposition 4.2, we know that faults can be detected at worst when the run reaches a BSCC. The proof consists in showing and using that the speed at which a BSCC is reached is uniform from any state.

Proof. Let $\mathcal{A}$ be an FF-diagnosable pLTS. Given a run ρ of $\mathcal{A}$, let ρ_{FF} be the corresponding run in $\mathcal{A}_{\mathsf{FF}}$: the states in ρ_{FF} extend the states appearing along ρ by subsets of possible correct states after the corresponding prefix of the observed sequence $\mathcal{P}(\rho)$. Let S_{BSCC} denotes the set of states of $\mathcal{A}_{\mathsf{FF}}$ that belong to a BSCC. Last, for every state (q, U) of $\mathcal{A}_{\mathsf{FF}}$ and every $n \in \mathbb{N}$, denote by $\mathsf{SR}_n^{q,U}$ the set of signalling runs in $\mathcal{A}_{\mathsf{FF}}$ of length n starting at (q, U).

[2]Recall that the set Q_f is absorbing

Let $\alpha > 0$. Our objective is to get n_α such that for every $n \geq n_\alpha$ and every minimal faulty run $\rho \in \mathsf{minF}$:

$$\mathbb{P}_{\mathcal{A}}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathsf{CorP}(\mathcal{P}(\rho)) > 0\}) \leq \alpha \cdot \mathbb{P}(\rho) \ .$$

We first exploit the almost sure convergence towards BSCC in $\mathcal{A}_{\mathsf{FF}}$. For every state (q, U) of $\mathcal{A}_{\mathsf{FF}}$, the measure of runs starting in (q, U) and avoiding all BSCC during n steps tends to 0, when n goes to infinity. Thus, given α, for every reachable (q, U), there exists $n_{q,U} \in \mathbb{N}$ such that for every $n \geq n_{q,U}$,

$$\mathbb{P}_{\mathcal{A}_{\mathsf{FF}}}(\{\rho'_{\mathsf{FF}} \in \mathsf{SR}_n^{q,U} \mid \mathsf{last}(\rho'_{\mathsf{FF}}) \notin S_{\mathsf{BSCC}}\}) \leq \alpha.$$

We define n_α as the maximum of $n_{q,U}$ over all states (q, U).

Now let ρ be a minimal faulty run of $\mathcal{A}$, and define $(q, U) = \mathsf{last}(\rho_{\mathsf{FF}})$. Since $n_\alpha \geq n_{q,U}$, $\mathbb{P}_{\mathcal{A}_{\mathsf{FF}}}(\{\rho'_{\mathsf{FF}} \in \mathsf{SR}_{n_\alpha}^{q,U} \mid \mathsf{last}(\rho'_{\mathsf{FF}}) \notin S_{\mathsf{BSCC}}\}) \leq \alpha$. Therefore, as $\mathcal{A}$ and $\mathcal{A}_{\mathsf{FF}}$ have the same probabilistic behaviour,

$$\mathbb{P}_{\mathcal{A}}(\{\rho' \in \mathsf{SR}_{n_\alpha+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathsf{last}(\rho'_{\mathsf{FF}}) \notin S_{\mathsf{BSCC}}\}) \leq \alpha \cdot \mathbb{P}(\rho).$$

Thanks to the characterisation of Proposition 4.2, all states in BSCC reachable from (q, U) in $\mathcal{A}_{\mathsf{FF}}$ necessarily are of the form $(q', \emptyset)$. Therefore, if a finite run ρ'_{FF} reaches such a BSCC, ρ'_{FF} admits no correct run with same observed sequence, and hence $\mathsf{CorP}(\mathcal{P}(\rho'_{\mathsf{FF}})) = 0$. Equivalently, $\mathsf{CorP}(\mathcal{P}(\rho')) > 0$ implies $\mathsf{last}(\rho'_{\mathsf{FF}}) \notin S_{\mathsf{BSCC}}$. Thus

$$\mathbb{P}_{\mathcal{A}}(\{\rho' \in \mathsf{SR}_{n_\alpha+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathsf{CorP}(\mathcal{P}(\rho')) > 0\}) \leq \alpha \cdot \mathbb{P}(\rho)$$

which shows that $\mathcal{A}$ is uniformly FF-diagnosable. □

1.1.2 FA-diagnosability

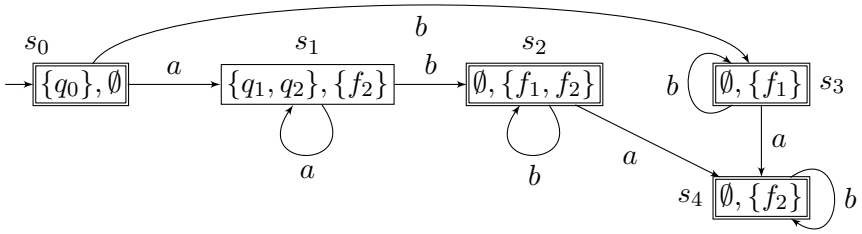


Figure 4.5: The FA-automaton of the pLTS of Figure 4.1.

For FA-diagnosability, we again start from $\mathsf{Obs}(\mathcal{A})$. Here, we need information about the ambiguity of both faulty and correct runs. Yet, we still do not need to keep all the information given by $\mathsf{Obs}(\mathcal{A})$. Indeed, we can gather the V and W components into a unique set, that we again call V. In other words we keep the information on which faulty states could be reached, but not the distinction between "old" and "new" faulty runs. The resulting simplified automaton is denoted by $\mathsf{FA}(\mathcal{A})$.

Example 4.5. *Figure 4.5 illustrates this construction on the pLTS of Figure 4.1. As expected, the* FA*-automaton is a refinement of the* FF*-automaton: the U-component of a state in* $\mathsf{FA}(\mathcal{A})$ *corresponds to a state in* $\mathsf{FF}(\mathcal{A})$*. For instance, state s_2 of Figure 4.3 is split here into s_2, s_3 and s_4.*

As we did for the FF case, we now define the pLTS $\mathcal{A}_{\mathsf{FA}} = \mathcal{A} \times \mathsf{FA}(\mathcal{A})$ as the product of $\mathcal{A}$ and $\mathsf{FA}(\mathcal{A})$ synchronised over observed events (the definition has the same structure as the one of Definition 4.2, only using $\mathsf{FA}(\mathcal{A})$ instead of $\mathsf{FF}(\mathcal{A})$). $\mathcal{A}_{\mathsf{FA}}$ is still a pLTS with same stochastic behaviour as $\mathcal{A}$ augmented with the relevant information of the observed sequence w.r.t FA-diagnosability.

Example 4.6. *Figure 4.6 continues Example 4.5 and shows the synchronised product for the pLTS of Figure 4.1.*

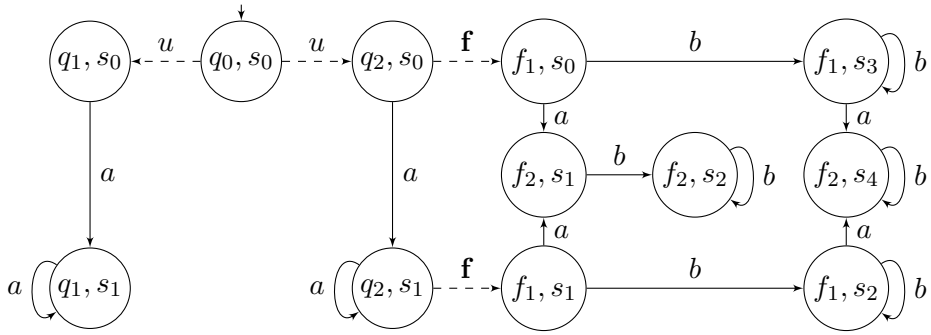


Figure 4.6: The synchronised product of pLTS of Figure 4.1 and its FA-automaton.

Again, FA-diagnosability is characterised through the BSCC of $\mathcal{A}_{\mathsf{FA}}$.

Proposition 4.4. *Let $\mathcal{A}$ be a finite pLTS. $\mathcal{A}$ is* FA*-diagnosable if and only if $\mathcal{A}_{\mathsf{FA}}$ has no BSCC that:*

- *either contains a state (q, U, V) with $q \in Q_f$ and $U \neq \emptyset$;*
- *or contains a state (q, U, V) with $q \in Q_c$ and $V \neq \emptyset$.*

Note that the characterisation of FA-diagnosability is symmetric for correct states and V-component (resp. faulty states and U-component). This reflects the symmetry of the definition of FA-diagnosability.

The main difference between this proof and the one of Proposition 4.2 is that the second item is not uniform inside a BSCC: there may exists a BSCC containing two states (q_1, U_1, V_1) and (q_2, U_2, V_2) with $q_1, q_2 \in Q_c$, $V_1 = \emptyset$ and $V_2 \neq \emptyset$. As a consequence it is harder to show that if a BSCC verifies the second item, then the pLTS is not FA-diagnosable. Instead of having every extension of the run to be correct ambiguous, the extensions only are ambiguous when they visit some particular states. However,

the probability that a run ends in such a state converges towards the steady state distribution of this state which is positive as the state belongs to a BSCC, contradicting the FA-diagnosability.

Proof. To prove the left-to-right implication, we proceed by contraposition. If one assumes the first item holds, the same argument as in the proof of Proposition 4.2 applies. Precisely, suppose that there exists a reachable BSCC C of $\mathcal{A}_{\mathsf{FA}}$ and a state $s = (q, U, V)$ in C such that $q \in Q_f$ and $U \neq \emptyset$. Let ρ be a signalling run leading from the initial state s_0 of $\mathcal{A}_{\mathsf{FA}}$ to s. Now, for every state $s' = (q', U', V') \in C$, necessarily $q' \in Q_f$ and $U' \neq \emptyset$, because C is strongly connected. So for every signalling run ρ' that extends ρ, writing $s' = (q', U', V')$ for the state ρ' leads to, there exists a correct signalling run ρ'' such that $\mathcal{P}(\rho'') = \mathcal{P}(\rho')$ and $q_0 \xrightarrow{\rho''} q''$ with $q'' \in U'$. As a consequence the observed sequence $\mathcal{P}(\rho'')$ is ambiguous, and for every $n \geq |\rho|_o$, $\mathbb{P}(\mathsf{FAmb}_n) \geq \mathbb{P}(\rho)$, so that $\mathcal{A}$ is not FA-diagnosable.
Suppose now that there exists a reachable BSCC C of $\mathcal{A}_{\mathsf{FA}}$ and a state $s = (q, U, V)$ in C such that $q \in Q_c$ and $V \neq \emptyset$. Since the pair (U, V) is unchanged by unobservable transitions, w.l.o.g we assume that s is the successor of some state of C by an observable event and we denote C' the set of such states. Observe that a signalling run that reaches s is ambiguous. Denote $\pi_i(s')$ the probability that a random run of length i ends in a state s'. In a finite DTMC, for every state s' of a reachable BSCC the Cesaro-limit $\pi_\infty(s') = \lim_{n\to\infty} 1/(n+1) \sum_{i=0}^{n} \pi_i(s')$ exists and is greater than 0. For $s' \in C'$ denote by $p_{s',s}$ the probability of an observable transition from s' to s. Then

$$0 < \sum_{s' \in C'} \pi_\infty(s') p_{s',s} \leq \liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} \alpha_i(s)$$

where $\alpha_i(s)$ is the probability that a random signalling run of length i ends in s. α_i differs from π_i by only considering signalling runs. From time 0 to time n, a run can be a signalling run at most $n+1$ times. Thus:

$$\frac{1}{n+1} \sum_{i=0}^{n} \alpha_i(s) \leq \frac{1}{n+1} \sum_{i=0}^{n} \mathbb{P}(\mathsf{CAmb}_i)$$

which implies that

$$0 < \liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} \mathbb{P}(\mathsf{CAmb}_i) \leq \limsup_{n\to\infty} \mathbb{P}(\mathsf{CAmb}_n) \ .$$

In this case also, we conclude that $\mathcal{A}$ is not FA-diagnosable.

The proof of Proposition 4.2 has established that a signalling run reaching a BSCC C where for every state $s = (q, U, V)$, q is faulty and $U = \emptyset$, is surely faulty. Similarly a signalling run that reaches a BSCC where for every state $s = (q, U, V)$, q is correct and $V = \emptyset$, is surely correct. Thus an ambiguous signalling run must only visit transient

states. Since runs almost surely leave the transient states and reach a BSCC, this implies that:

$$\limsup_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) + \mathbb{P}(\mathsf{CAmb}_n) = 0 \ ,$$

and therefore, the pLTS is FA-diagnosable. □

Let us emphasise that although there does not exist a simple logical characterisation of FA-diagnosability, in finite pLTS, it enjoys a characterisation that is similar to the one of FF-diagnosability.

1.1.3 IA-diagnosability

IA-diagnosability is the notion of exact diagnosability for which we need to use $\mathsf{Obs}(\mathcal{A})$ with no simplification. However, to stick to the presentation for the other diagnosability notions, we write here $\mathsf{IA}(\mathcal{A})$ for $\mathsf{Obs}(\mathcal{A})$. As before, to come up with a characterisation, one builds $\mathcal{A}_{\mathsf{IA}} = \mathcal{A} \times \mathsf{IA}(\mathcal{A})$, the product of $\mathcal{A}$ and $\mathsf{IA}(\mathcal{A})$ synchronised over observed events.

Example 4.7. *Figure 4.7 shows the synchronised product corresponding to the pLTS depicted in Figure 4.1. Among the BSCC, all the faulty ones (*i.e. *the ones reached after a faulty event) have $U = \emptyset$, while $\{(q_1, s_1), (q_1, s_1')\}$, the single one that is reached by a correct run, has a state (q_1, s_1') with $W = \emptyset$.*

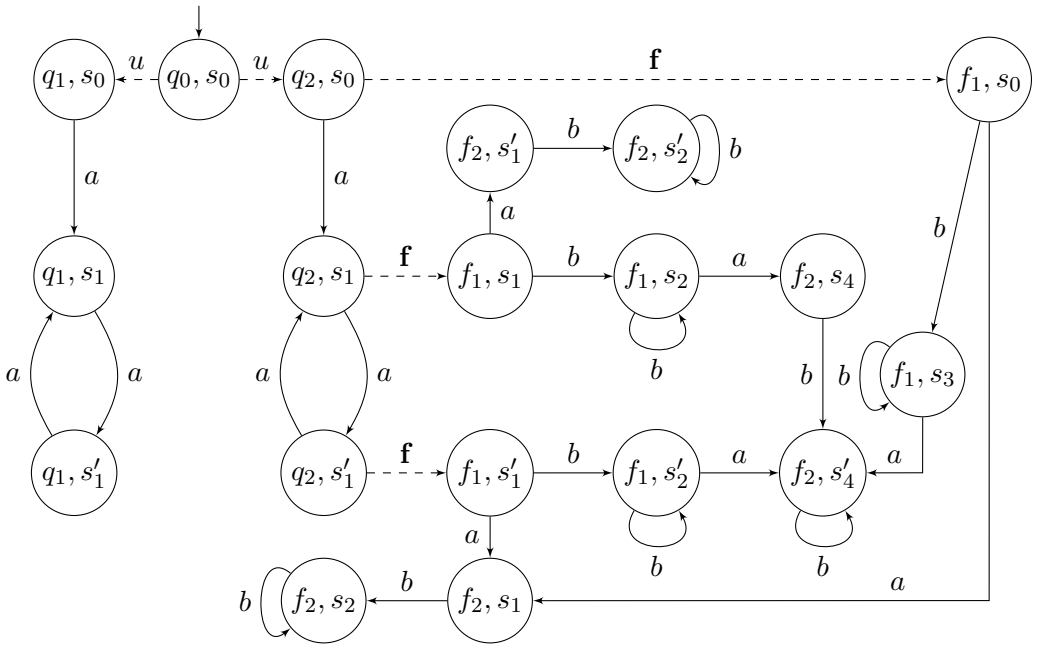


Figure 4.7: The synchronised product of the pLTS of Figure 4.1 and its IA-automaton.

We now establish a characterisation of IA-diagnosability on $\mathcal{A}_{\mathsf{IA}}$.

Proposition 4.5. *Let $\mathcal{A}$ be a finite pLTS. $\mathcal{A}$ is* IA*-diagnosable if and only if $\mathcal{A}_{\mathsf{IA}}$ has no BSCC such that:*

- *either, all its states (q, U, V, W) fulfil $q \in Q_f$ and $U \neq \emptyset$;*
- *or all its states (q, U, V, W) fulfil $q \in Q_c$ and $W \neq \emptyset$.*

Proof. This proof relies on Proposition 4.1. An infinite run is not ambiguous if its observation satisfy the Büchi condition of $\mathsf{Obs}(\mathcal{A})$, therefore $\mathcal{A}_{\mathsf{IA}}$ will be IA-diagnosable if the pLTS almost surely satisfies the Büchi condition. As a run almost surely reaches a BSCC, every BSCC must contain a state that allows the diagnosis.

Assume first that $\mathcal{A}_{\mathsf{IA}}$ has a BSCC with (at least) some state (q, U, V, W) with $q \in Q_f$ and $U \neq \emptyset$. Using Proposition 4.2, $\mathcal{A}$ is not FF-diagnosable and thus not IA-diagnosable either, due to Theorem 3.1. Assume now some BSCC C of $\mathcal{A}_{\mathsf{IA}}$ has all its states (q, U, V, W) with $q \in Q_c$ and $W \neq \emptyset$. In particular none of these states are accepting for the deterministic Büchi automaton $\mathsf{IA}(\mathcal{A})$. Let ρ be a finite signalling run that hits C. By Proposition 4.1, any infinite run ρ' that extends ρ is ambiguous. From $q \in Q_c$ we deduce that $\mathbb{P}(\mathsf{CAmb}_\infty) \geq \mathbb{P}(\rho) > 0$. Therefore $\mathcal{A}$ is not IA-diagnosable.

Assume now $\mathcal{A}_{\mathsf{IA}}$ has no BSCC such that either, all its states (q, U, V, W) fulfil $q \in Q_f$ and $U \neq \emptyset$, or all its states (q, U, V, W) fulfil $q \in Q_c$ and $W \neq \emptyset$. First observe that in case some BSCC of $\mathcal{A}_{\mathsf{IA}}$ contains some state (q, U, V, W) with $q \in Q_f$ and $U \neq \emptyset$, then all its states satisfy the same constraints. Moreover, if some state (q, U, V, W) of a BSCC has $q \in Q_c$, then all states of this BSCC have their first component in Q_c. Therefore, the condition can be reformulated as follows: all BSCC C of $\mathcal{A}_{\mathsf{IA}}$ satisfy:

- either all states (q, U, V, W) of C fulfil $q \in Q_f$ and $U = \emptyset$;
- or all states (q, U, V, W) of C fulfil $q \in Q_c$ and some state (q, U, V, W) of C fulfils $W = \emptyset$.

Whatever the case, all BSCC contain (at least) an accepting state for the Büchi condition of $\mathsf{IA}(\mathcal{A})$. Since all runs almost surely end in a BSCC and visit each of its states infinitely often, using Proposition 4.1, almost all runs of $\mathcal{A}_{\mathsf{IA}}$ are unambiguous. This proves that $\mathcal{A}$ is IA-diagnosable. □

Surprisingly, while in general FA-diagnosability could not be characterised by a logical formula contrary to IA-diagnosability, restricted to finite systems, the characterisation of IA-diagnosability is the more involved one.

1.2 Approximate diagnosis

We now turn to the characterisation of approximate diagnosis and particularly of AFF-diagnosability. The reason why we only consider AFF-diagnosability here will become clear in Subsection 2.2.1 where we show that all other approximate diagnosability notions are undecidable. Our characterisation of AFF-diagnosability relies on the notion of

distance between two Markov chains with labels on the transitions. A *labelled Markov chain* (LMC) is a pLTS where every event is observable: $\Sigma = \Sigma_o$. In order to exploit results of [CK14] on LMC in our context of pLTS, we introduce the mapping $\mathcal{M}$ that computes *in polynomial time* the probabilistic closure of a pLTS w.r.t. unobservable events and produces an LMC. Informally, the probabilities of all paths of $\mathcal{A}$ from state q to state q' with same observed sequence $a \in \Sigma_o$ are gathered to obtain the probability in $\mathcal{M}(\mathcal{A})$ to move from q to q' with label a. The transformation is formally defined below. For sake of simplicity, we denote by $\mathcal{A}_q$, the pLTS $\mathcal{A}$ where the initial state has been substituted by q.

Definition 4.3. *Given a pLTS $\mathcal{A} = \langle Q, q_0, \Sigma, T, \mathbf{P}\rangle$ with $\Sigma = \Sigma_o \uplus \Sigma_u$, the labelled Markov chain $\mathcal{M}(\mathcal{A}) = \langle Q, q_0, \Sigma_o, T', \mathbf{P}'\rangle$ is defined by:*

- $T' = \{(q, a, q') \mid \exists \rho = q \cdots a q' \in \mathsf{SR}_1(\mathcal{A}_q)\}$ *(and so $a \in \Sigma_o$).*
- *for every $(q, a, q') \in T'$, $\mathbf{P}'(q, a, q') = \mathbb{P}(\{\rho \in \mathsf{SR}_1(\mathcal{A}_q) \mid \rho = q \cdots a q'\})$.*

Example 4.8. *The LMC associated with the pLTS of Figure 4.1 is represented in Figure 4.8. The transition from q_0 to q_1 which was unobservable has been replaced by a transition labelled by 'a'. The new transition has probability $1/2$ which is the product of the probability of the replaced unobservable transition (of value $1/2$) and of the transition labelled by 'a' that followed (of value 1). A transition from q_0 to f_2 appeared, it replaces the run $q_0 u q_2 \mathbf{f} f_1 a f_2$ which has probability $1/8$.*

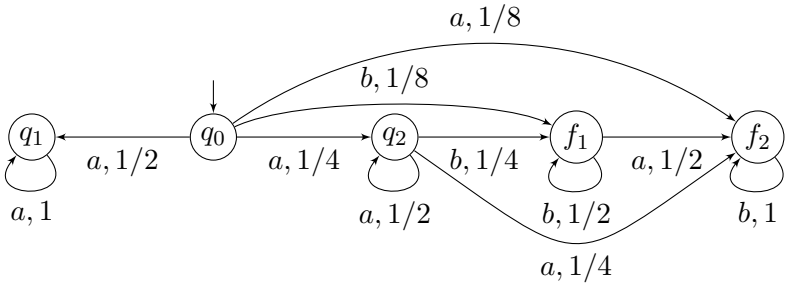


Figure 4.8: The LMC obtained from the pLTS of Figure 4.1.

Let E be a *prospect*[3] of Σ_o^ω (*i.e.* a measurable subset of Σ_o^ω for the standard measure), we denote by $\mathbb{P}^{\mathcal{M}}(E)$ the probability that prospect E occurs in the LMC $\mathcal{M}$. Given two LMC $\mathcal{M}_1$ and $\mathcal{M}_2$, the (probabilistic) distance between $\mathcal{M}_1$ and $\mathcal{M}_2$ generalises the concept of distance for distributions. Given a prospect E, $|\mathbb{P}^{\mathcal{M}_1}(E) - \mathbb{P}^{\mathcal{M}_2}(E)|$ expresses the absolute difference between the probabilities that E occurs in $\mathcal{M}_1$ and in $\mathcal{M}_1$. The distance between $\mathcal{M}_1$ and $\mathcal{M}_2$ is defined as the supremum over the prospects:

[3]The term used in the literature is event. We differ here as we already use event for the letters labelling the transitions as established in Definition 2.5, page 38

Definition 4.4. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two LMC over the same alphabet Σ_o. Then $d(\mathcal{M}_1, \mathcal{M}_2)$ the* distance between $\mathcal{M}_1$ and $\mathcal{M}_2$ *is:*

$$d(\mathcal{M}_1, \mathcal{M}_2) = \sup\{|\mathbb{P}^{\mathcal{M}_1}(E) - \mathbb{P}^{\mathcal{M}_2}(E)| \mid E \text{ prospect of } \Sigma_o^\omega\} \ .$$

Example 4.9. *Consider the LMC of Figure 4.8, called $\mathcal{M}_1$, and the one of Figure 4.9, called $\mathcal{M}_2$. The prospect $E = \{a^\omega\}$ has probability $1/2$ in $\mathcal{M}_1$ and 0 in $\mathcal{M}_2$. Thus $d(\mathcal{M}_1, \mathcal{M}_2) \geq 1/2$. Moreover, $\mathcal{M}_2$ can be obtained from $\mathcal{M}_1$ by deleting the state q_1 (and the associated transitions) and merging q_0 with q_2. As a consequence, for any prospect E such that $a^\omega \notin E$, $\mathbb{P}^{\mathcal{M}_2}(E) = 2.\mathbb{P}^{\mathcal{M}_1}(E)$. We thus have $\mathbb{P}^{\mathcal{M}_1}(\{a^\omega\}) - \mathbb{P}^{\mathcal{M}_2}(\{a^\omega\}) = 1/2$, $\mathbb{P}^{\mathcal{M}_2}(\Sigma_o^\omega \setminus \{a^\omega\}) - \mathbb{P}^{\mathcal{M}_1}(\Sigma_o^\omega \setminus \{a^\omega\}) = 1/2$, and for any other prospect E, the difference is smaller or equal to $1/2$. Therefore $d(\mathcal{M}_1, \mathcal{M}_2) = 1/2$.*

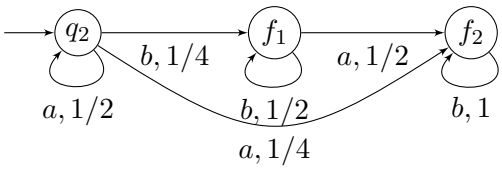


Figure 4.9: An example of LMC.

The *distance* 1 *problem* asks, given two LMC $\mathcal{M}_1$ and $\mathcal{M}_2$, whether $d(\mathcal{M}_1, \mathcal{M}_2) = 1$. The next proposition summarises the results of Chen and Kiefer on LMC, that we use later.

Proposition 4.6 ([CK14])**.**

- *Given two LMC $\mathcal{M}_1, \mathcal{M}_2$, there exists a prospect E such that:*

$$d(\mathcal{M}_1, \mathcal{M}_2) = \mathbb{P}^{\mathcal{M}_1}(E) - \mathbb{P}^{\mathcal{M}_2}(E).$$

- *The distance 1 problem for LMC is decidable in polynomial time.*

The first item of this proposition states that the supremum is reached (and thus is a maximum). In fact, given two LMC $\mathcal{M}_1, \mathcal{M}_2$, the authors show that one prospect reaching the maximum is $E = \{w \in \Sigma_o^\omega \mid \lim_{n \to \infty} \frac{\mathbb{P}^{\mathcal{M}_1}(w_{\leq n})}{\mathbb{P}^{\mathcal{M}_2}(w_{\leq n})} \geq 1\}$.

We now use the notion of distance 1 to characterise AFF-diagnosability. Let us first consider a subclass of pLTS called *initial-fault* pLTS. Informally, an initial-fault pLTS $\mathcal{A}$ consists of two disjoint pLTS $\mathcal{A}^f$ and $\mathcal{A}^c$ and an initial state q_0 with an outgoing unobservable correct transition leading to $\mathcal{A}^c$ and a transition labelled by $\mathbf{f}$ leading to $\mathcal{A}^f$ (see Figure 4.10). Moreover no faulty transitions occur in $\mathcal{A}^c$. In other words, if a fault occurs during a run of an initial-fault pLTS, it does so on the very first transition.

Definition 4.5 (Initial-fault pLTS)**.** *A pLTS $\mathcal{A} = \langle Q, q_0, \Sigma, T, \mathbf{P}\rangle$ is an* initial fault pLTS *if there exist two disjoint pLTS $\mathcal{A}^f = \langle Q_f, q_f, \Sigma, T_f, \mathbf{P}_f\rangle$ and $\mathcal{A}^c = \langle Q_c, q_c, \Sigma \setminus \{\mathbf{f}\}, T_c, \mathbf{P}_c\rangle$ such that:*

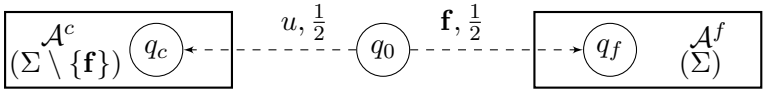


Figure 4.10: Schematic representation of an initial-fault pLTS.

- $Q = \{q_0\} \uplus Q_f \uplus Q_c$;
- $T = T_f \uplus T_c \uplus \{(q_0, u, q_c), (q_0, \mathbf{f}, q_f)\}$ *with* $u \in \Sigma_u$;
- *for every* $t \in T_f$, *we have* $\mathbf{P}(t) = \mathbf{P}_f(t)$ *and for every* $t \in T_c$, *we have* $\mathbf{P}(t) = \mathbf{P}_c(t)$, *and* $\mathbf{P}((q_0, u, q_c)) = \mathbf{P}((q_0, \mathbf{f}, q_f)) = 1/2$.

We denote such a pLTS by $\mathcal{A} = \langle q_0, \mathcal{A}^f, \mathcal{A}^c \rangle$.

Example 4.10. *The pLTS of Figure 4.11 is a simple initial-fault pLTS.*

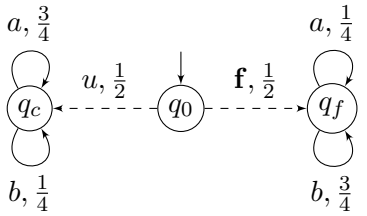


Figure 4.11: A uniformly AFF-diagnosable initial-fault pLTS.

Under the initial-fault restriction, we can get a simple characterisation for AFF-diagnosability as established in the next lemma. The idea of this characterisation is then extended to every pLTS.

Lemma 4.1. *Let* $\mathcal{A} = \langle q_0, \mathcal{A}^f, \mathcal{A}^c \rangle$ *be an initial-fault pLTS. Then* $\mathcal{A}$ *is* AFF*-diagnosable if and only if* $d(\mathcal{M}(\mathcal{A}^f), \mathcal{M}(\mathcal{A}^c)) = 1$.

We write $\mathbb{P}$, $\mathbb{P}_f$ and $\mathbb{P}_c$ for the probability measures of pLTS $\mathcal{A}$, $\mathcal{A}^f$ and $\mathcal{A}^c$. By construction of $\mathcal{M}(\mathcal{A}^f)$ and $\mathcal{M}(\mathcal{A}^c)$, for every observed sequence σ, $\mathbb{P}^{\mathcal{M}(\mathcal{A}^f)}(\sigma) = \mathbb{P}_f(\sigma)$ and similarly $\mathbb{P}^{\mathcal{M}(\mathcal{A}^c)}(\sigma) = \mathbb{P}_c(\sigma)$. In words, the mapping $\mathcal{M}$ leaves unchanged the probability of occurrence of an observed sequence.

AFF-diagnosability is a property of identification (decide whether the run is faulty) of the finite runs that start by a fault with high probability, using the probability $\mathbb{P}$. The distance 1 between $\mathcal{M}(\mathcal{A}^f)$ and $\mathcal{M}(\mathcal{A}^c)$ is also a property of identification (decide which LMC the run that produces the infinite observed sequence belongs to) of infinite observed sequences, using the probabilities $\mathbb{P}_f$ and $\mathbb{P}_c$. As mentioned above, these three probability measures are probabilities of parts of the pLTS $\mathcal{A}$ and are thus related. The proof therefore mostly consists in understanding the links between the two identification

properties: how to translate the set of finite runs identified by AFF-diagnosability into a set of infinite sequences and reciprocally. The intuition to establish the relation is that the prefixes of an infinite faulty run reveals the fault with arbitrarily high accuracy if and only if the associated infinite observed sequence "reveals" that the run belongs to $\mathcal{M}(\mathcal{A}^f)$.

Proof. Let us prove the equivalence, starting with the left-to-right implication.

• Assume $\mathcal{A}$ is AFF-diagnosable. Then, for every $\varepsilon > 0$ and every minimal faulty run ρ:

$$\lim_{n\to\infty} \mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathsf{CorP}(\mathcal{P}(\rho')) > \varepsilon\}) = 0. \tag{4.1}$$

Pick some $0 < \varepsilon < 1$. Applying Equation (4.1) on the minimal faulty run $\rho_f = q_0 \mathbf{f} q_f$ with $|\mathcal{P}(\rho_f)| = 0$, there exists some $n \in \mathbb{N}$ such that:

$$\mathbb{P}(\{\rho \in \mathsf{SR}_n \mid \rho_f \preceq \rho \wedge \mathsf{CorP}(\mathcal{P}(\rho)) > \varepsilon\}) \leq \varepsilon \ .$$

Let $\mathfrak{S}$ be the set of observed sequences of faulty runs with observable length n and correctness proportion not exceeding threshold ε:

$$\mathfrak{S} = \{\sigma \in \Sigma_o^n \mid \exists \rho \in \mathsf{SR}_n, \mathcal{P}(\rho) = \sigma \wedge \rho_f \preceq \rho \wedge \mathsf{CorP}(\sigma) \leq \varepsilon\} \ .$$

We define $E = \mathsf{Cyl}(\mathfrak{S})$ to be the prospect consisting of the infinite suffixes of these sequences. Let us show that $\mathbb{P}_c(E) \leq \varepsilon/(1-\varepsilon)$ and $\mathbb{P}_f(E) \geq 1 - 2\varepsilon$. We have:

$$\mathbb{P}_f(E) = 1 - 2\,\mathbb{P}(\{\rho \in \mathsf{SR}_n \mid \rho_f \preceq \rho \wedge \mathsf{CorP}(\mathcal{P}(\rho)) > \varepsilon\}) \geq 1 - 2\varepsilon$$

where the factor 2 comes from the probability 1/2 in $\mathcal{A}$ to enter $\mathcal{A}^f$ that $\mathbb{P}_f$ does not take into account contrary to $\mathbb{P}$.

Moreover, for every observed sequence $\sigma \in \mathfrak{S}$, $\mathsf{CorP}(\sigma) \leq \varepsilon$. Using the definition of CorP:

$$\mathsf{CorP}(\sigma) = \frac{\mathbb{P}(\{\rho \in \mathsf{C} \cap \mathsf{SR}_n \mid \mathcal{P}(\rho) = \sigma\})}{\mathbb{P}(\{\rho \in \mathsf{SR}_n \mid \mathcal{P}(\rho) = \sigma\})} = \frac{\mathbb{P}_c(\sigma)}{\mathbb{P}_c(\sigma) + \mathbb{P}_f(\sigma)} \leq \varepsilon.$$

Thus, $\mathbb{P}_c(\sigma) \leq \frac{\varepsilon}{1-\varepsilon}\mathbb{P}_f(\sigma)$. Hence:

$$\mathbb{P}_c(E) = \sum_{\sigma\in\mathfrak{S}} \mathbb{P}_c(\sigma) \leq \sum_{\sigma\in\mathfrak{S}} \frac{\varepsilon}{1-\varepsilon}\mathbb{P}_f(\sigma) = \frac{\varepsilon}{1-\varepsilon}\mathbb{P}_f(E) \leq \frac{\varepsilon}{1-\varepsilon}.$$

Therefore $d(\mathcal{M}(\mathcal{A}^c), \mathcal{M}(\mathcal{A}^f)) \geq \mathbb{P}_f(E) - \mathbb{P}_c(E) \geq 1 - \varepsilon(2 + \frac{1}{1-\varepsilon})$. Since ε was arbitrary, taking the limit when ε goes to 0, we obtain the desired result: $d(\mathcal{M}(\mathcal{A}^c), \mathcal{M}(\mathcal{A}^f)) = 1$. Note that we did not exhibit the prospect that reaches the maximum but only a prospect ε-close to it. The proof could be modified to use this maximum prospect, but it makes the proof unnecessarily more complicated.

• Conversely assume that $d(\mathcal{M}(\mathcal{A}^f), \mathcal{M}(\mathcal{A}^c)) = 1$. Thanks to Proposition 4.6, there exists a prospect $E \subseteq \Sigma_o^\omega$ such that $\mathbb{P}_f(E) = 1$ and $\mathbb{P}_c(E) = 0$.

For every $n \in \mathbb{N}$, let $\mathfrak{S}_n$ be the set of prefixes of length n of the observed sequences of E: $\mathfrak{S}_n = \{\sigma \in \Sigma_o^n \mid \exists \sigma' \in E, \sigma \preceq \sigma'\}$. For every $\varepsilon > 0$, we also define $\mathfrak{S}_n^\varepsilon$ as the

subset of $\mathfrak{S}_n$ consisting of sequences whose correctness proportion exceeds threshold ε: $\mathfrak{S}_n^\varepsilon = \{\sigma \in \mathfrak{S}_n \mid \mathsf{CorP}(\sigma) > \varepsilon\}$.

From $\bigcap_{n\in\mathbb{N}} \mathsf{Cyl}(\mathfrak{S}_n) = E$, we derive that $\lim_{n\to\infty} \mathbb{P}_c(\mathfrak{S}_n) = \mathbb{P}_c(E) = 0$. Thus $\lim_{n\to\infty} \mathbb{P}_c(\mathfrak{S}_n^\varepsilon) = 0$.

On the other hand, for every $n \in \mathbb{N}$,

$$\mathbb{P}_c(\mathfrak{S}_n^\varepsilon) = \sum_{\sigma\in\mathfrak{S}_n^\varepsilon} \mathbb{P}_c(\sigma) > \sum_{\sigma\in\mathfrak{S}_n^\varepsilon} \frac{\varepsilon}{1-\varepsilon}\mathbb{P}_f(\sigma) = \frac{\varepsilon}{1-\varepsilon}\mathbb{P}_f(\mathfrak{S}_n^\varepsilon) \ .$$

Since ε is fixed, we have $\mathbb{P}_f(\mathfrak{S}_n^\varepsilon) < \frac{1-\varepsilon}{\varepsilon}\mathbb{P}_c(\mathfrak{S}_n^\varepsilon)$, thus $\lim_{n\to\infty} \mathbb{P}_c(\mathfrak{S}_n^\varepsilon) = 0$ implies that $\lim_{n\to\infty} \mathbb{P}_f(\mathfrak{S}_n^\varepsilon) = 0$.

Let ρ be a minimal faulty run and $\alpha > 0$. There exists $n_\alpha \geq |\rho|_o = 1$ such that for all $n \geq n_\alpha$, $\mathbb{P}_f(\mathfrak{S}_n^\varepsilon) \leq \alpha$. Let $n \geq n_\alpha$, and $\widetilde{\mathfrak{S}}_n$ be the set of observed sequences of length n triggered by a run with prefix ρ and whose correctness proportion exceeds ε:

$$\widetilde{\mathfrak{S}}_n = \{\sigma \in \Sigma_o^n \mid \exists \rho' \in \mathsf{SR}_n, \rho \preceq \rho' \wedge \mathcal{P}(\rho') = \sigma \wedge \mathsf{CorP}(\sigma) > \varepsilon\} \ .$$

Let us prove that $\mathbb{P}(\widetilde{\mathfrak{S}}_n) \leq \alpha$. On the one hand, since $\mathbb{P}_f(\mathfrak{S}_n) \geq \mathbb{P}_f(E) = 1$, $\mathbb{P}_f(\widetilde{\mathfrak{S}}_n \cap (\Sigma_o^n \setminus \mathfrak{S}_n)) = 0$. On the other hand, since $\mathbb{P}_f(\mathfrak{S}_n^\varepsilon) \leq \alpha$, $\mathbb{P}_f(\widetilde{\mathfrak{S}}_n \cap \mathfrak{S}_n) \leq \mathbb{P}_f(\mathfrak{S}_n^\varepsilon) \leq \alpha$. Thus, $\mathbb{P}_f(\widetilde{\mathfrak{S}}_n) = \mathbb{P}_f(\widetilde{\mathfrak{S}}_n \cap \mathfrak{S}_n) + \mathbb{P}_f(\widetilde{\mathfrak{S}}_n \cap (\Sigma_o^n \setminus \mathfrak{S}_n)) \leq \alpha$. Because α was taken arbitrary, we obtain that $\lim_{n\to\infty} \mathbb{P}_f(\widetilde{\mathfrak{S}}_n) = 0$. Observe now that

$$\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho' \wedge \mathsf{CorP}(\mathcal{P}(\rho')) > \varepsilon\}) = \frac{1}{2}\mathbb{P}_f(\widetilde{\mathfrak{S}}_n).$$

Therefore, $\lim_{n\to\infty} \mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho' \wedge \mathsf{CorP}(\mathcal{P}(\rho')) > \varepsilon\}) = 0$. In conclusion $\mathcal{A}$ is AFF-diagnosable. □

This characterisation shows that, for initial-fault pLTS, AFF-diagnosability can be reduced to the distance 1 problem. As one can perform the closure w.r.t. unobservable events and check the distance 1 in polynomial time, AFF-diagnosability for initial-fault pLTS belongs to PTIME.

Example 4.11. *Consider the initial-fault pLTS of Figure 4.11 $\langle q_0, \mathcal{A}^f, \mathcal{A}^c\rangle$ and the prospect $E = \{\sigma \in \Sigma_o^\omega \mid \limsup_{n\in\mathbb{N}} \frac{|\sigma_{\leq n}|_b}{n} \geq \frac{1}{2}\}$. As a 'b' has a probability 3/4 to be observed at each step in $\mathcal{A}^f$ and 1/4 in $\mathcal{A}^c$, $\mathbb{P}_f(E) = 1$ and $\mathbb{P}_c(E) = 0$ where $\mathbb{P}_f$ and $\mathbb{P}_c$ are the probability measures of $\mathcal{A}^f$ and $\mathcal{A}^c$. Therefore this initial-fault pLTS is* AFF*-diagnosable. In fact, as this pLTS has a single minimal faulty run, it is even uniformly* AFF*-diagnosable.*

Remark 4.1. *There exists a single minimal faulty run in every initial-fault pLTS. As a consequence,* AFF*-diagnosability and uniform* AFF*-diagnosability are equivalent for initial-fault pLTS.*

In order to understand why characterising AFF-diagnosability for general pLTS is more involved, consider the pLTS $\mathcal{A}$ presented in Figure 4.12. Recall that $\mathcal{A}$ is AFF-diagnosable as shown in the proof of Theorem 3.1, page 73.

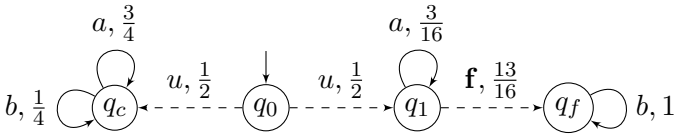


Figure 4.12: An AFF-diagnosable pLTS where the distance 1 characterisation cannot be applied in a simple way.

Let us look at the distance between pairs of a correct and a faulty states of $\mathcal{A}$ that can be reached by runs with the same observed sequence. On the one hand, we have $d(\mathcal{M}(\mathcal{A}_{q_1}), \mathcal{M}(\mathcal{A}_{q_f})) \leq 3/16$ since for any prospect E either (1) $b^\omega \in E$ implying $\mathbb{P}^{\mathcal{M}(\mathcal{A}_{q_f})}(E) = 1$ and $\mathbb{P}^{\mathcal{M}(\mathcal{A}_{q_0})}(E) \geq 13/16$ or (2) $b^\omega \notin E$ implying $\mathbb{P}^{\mathcal{M}(\mathcal{A}_{q_f})}(E) = 0$ and $\mathbb{P}^{\mathcal{M}(\mathcal{A}_{q_0})}(E) \leq 3/16$. On the other hand, $d(\mathcal{M}(\mathcal{A}_{q_c}), \mathcal{M}(\mathcal{A}_{q_f})) = 1$ since $\mathbb{P}^{\mathcal{M}(\mathcal{A}_{q_f})}(b^\omega) = 1$ and $\mathbb{P}^{\mathcal{M}(\mathcal{A}_{q_c})}(b^\omega) = 0$.
Intuitively, the pair (q_1, q_f) is irrelevant, since the correct state q_1 does not belong to a BSCC of the pLTS, while (q_c, q_f) is relevant since q_c belongs to a BSCC triggering a "recurrent" ambiguity. The next theorem characterises AFF-diagnosability for general pLTS, establishing the soundness of this intuition.

Theorem 4.1. *Let $\mathcal{A}$ be a pLTS. Then, $\mathcal{A}$ is AFF-diagnosable if and only if for every correct state q_c belonging to a BSCC and every faulty state q_f reachable by a faulty run ρ_f such that q_c is reachable by a run with same observed sequence, $d(\mathcal{M}(\mathcal{A}_{q_c}), \mathcal{M}(\mathcal{A}_{q_f})) = 1$.*

The proof of Theorem 4.1, due to its complexity and length, is divided into two lemmas, Lemma 4.2 and Lemma 4.3 given below, each of them stating one implication of the equivalence.

Lemma 4.2. *Let $\mathcal{A}$ be a pLTS. If there exists $q_c \in Q_c$ belonging to a BSCC, $q_f \in Q_f$ such that $d(\mathcal{M}(\mathcal{A}_{q_f}), \mathcal{M}(\mathcal{A}_{q_c})) < 1$ and runs $q_0 \stackrel{\rho_c}{\Rightarrow} q_c$ and $q_0 \stackrel{\rho_f}{\Rightarrow} q_f$ such that $\mathcal{P}(\rho_c) = \mathcal{P}(\rho_f)$, then $\mathcal{A}$ is not AFF-diagnosable.*

This lemma is the easiest of the two. It is proved by contraposition. Assume there exist two states in $\mathcal{A}$, $q_c \in Q_c$ belonging to a BSCC and $q_f \in Q_f$ reachable resp. by ρ_c and ρ_f with $\mathcal{P}(\rho_c) = \mathcal{P}(\rho_f)$, and with $d(\mathcal{M}(\mathcal{A}_{q_c}), \mathcal{M}(\mathcal{A}_{q_f})) < 1$. Applying Lemma 4.1 to the initial-fault pLTS $\mathcal{A}' = \langle q'_0, \mathcal{A}_{q_f}, \mathcal{A}_{q_c} \rangle$ where q'_0 is a new state, one deduces that $\mathcal{A}'$ is not AFF–diagnosable. First we relate the probabilities of runs in $\mathcal{A}$ and $\mathcal{A}'$. Then we show that considering the additional faulty runs with same observed sequence as ρ_f does not make $\mathcal{A}$ AFF–diagnosable.

Proof. Let $\mathcal{A}$ be a pLTS, assume there exists $q_c \in Q_c$ belonging to a BSCC, $q_f \in Q_f$ such that $d(\mathcal{M}(\mathcal{A}_{q_f}), \mathcal{M}(\mathcal{A}_{q_c})) < 1$ and runs $q_0 \stackrel{\rho_c}{\Rightarrow} q_c$ and $q_0 \stackrel{\rho_f}{\Rightarrow} q_f$ such that $\mathcal{P}(\rho_c) = \mathcal{P}(\rho_f)$. Let us introduce some notations:

$$\sigma_0 := \mathcal{P}(\rho_f) = \mathcal{P}(\rho_c), \quad p_f := \mathbb{P}_{\mathcal{A}}(\rho_f), \quad p_c := \mathbb{P}_{\mathcal{A}}(\rho_c) \ .$$

Let p_g ($\geq p_f$) be the probability of the faulty runs with observed sequence σ_0:

$$p_g = \mathbb{P}_{\mathcal{A}}(\{\rho \in \mathsf{SR}_{|\sigma|} \mid \mathcal{P}(\rho) = \sigma_0, \text{ and } \rho \text{ is faulty}\}) \ .$$

For all $n \geq |\sigma|$, let $\mathfrak{S}_n$ be the set of observed sequences of length n "extending" ρ_f:

$$\mathfrak{S}_n = \{\sigma \in \Sigma_o^n \mid \exists \rho \in \mathsf{SR}_n, \rho_f \preceq \rho \wedge \mathcal{P}(\rho) = \sigma\} \ .$$

Given $\sigma \in \mathfrak{S}_n$, we refine p_f, p_c and p_g as follows.

- $p_f^\sigma = \mathbb{P}_{\mathcal{A}}(\{\rho \in \mathsf{SR}_n \mid \rho_f \preceq \rho \wedge \mathcal{P}(\rho) = \sigma\})$;
- $p_c^\sigma = \mathbb{P}_{\mathcal{A}}(\{\rho \in \mathsf{SR}_n \mid \rho_c \preceq \rho \wedge \mathcal{P}(\rho) = \sigma\})$;
- $p_g^\sigma = \mathbb{P}_{\mathcal{A}}(\{\rho \in \mathsf{SR}_n \mid \rho \text{ is faulty and } \mathcal{P}(\rho) = \sigma\})$.

We introduce the initial-fault pLTS $\mathcal{A}' = \langle q_0', \mathcal{A}_{q_f}, \mathcal{A}_{q_c}\rangle$ for some new state q_0'. It is well-defined since q_c belongs to a BSCC so that $\mathcal{A}_{q_c}$ does not trigger faults. We write $\mathbb{P}'$ for the probability measure in $\mathcal{A}'$. Since $d(\mathcal{M}(\mathcal{A}_{q_f}), \mathcal{M}(\mathcal{A}_{q_c})) < 1$, due to Lemma 4.1, there exist positive reals $\alpha', \varepsilon' \leq 1$ such that for all $n_0 \in \mathbb{N}$ there exists $n \geq n_0$:

$$\mathbb{P}_{\mathcal{A}'}\{\rho \in \mathsf{SR}_n \mid q_0' \mathbf{f} q_f \preceq \rho \wedge \mathsf{CorP}(\mathcal{P}(\rho)) > \varepsilon\} > \alpha' \ .$$

This entails the following inequality for $\mathcal{A}$:

$$\mathbb{P}(\{\rho \in \mathsf{SR}_n \mid \rho_f \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + \frac{p_c}{p_f} p_f^{\mathcal{P}(\rho)}} > \varepsilon'\}) > 2p_f\alpha' \ .$$

Indeed in $\mathcal{A}'$, the probability of the set of faulty (resp. correct) run with observed sequence $\mathcal{P}(\rho)$ is $\frac{p_f^{\mathcal{P}(\rho)}}{2p_f}$ (resp. $\frac{p_c^{\mathcal{P}(\rho)}}{2p_c}$): the probability in $\mathcal{A}'$ to go in q_f (resp. q_c) initially, $1/2$, times the probability in $\mathcal{A}$ of the runs extending ρ_f (resp. ρ_c) with observation $\mathcal{P}(\rho)$, $p_f^{\mathcal{P}(\rho)}$ (resp. $p_c^{\mathcal{P}(\rho)}$), divided by the probability of ρ_f (resp. ρ_c), p_f (resp. p_c). Finally the $2p_f$ factor of the lower bound takes into account the fact that the probability of reaching q_f in $\mathcal{A}'$ is $1/2$ while the probability of ρ in $\mathcal{A}$ is p_f.
Observe that $\frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + \frac{p_c}{p_f} p_f^{\mathcal{P}(\rho)}} > \varepsilon'$ is equivalent to $\frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_f^{\mathcal{P}(\rho)}} > \frac{\varepsilon' p_c}{\varepsilon' p_c + (1-\varepsilon')p_f}$. So defining $\tilde{\varepsilon} = \frac{\varepsilon' p_c}{\varepsilon' p_c + (1-\varepsilon')p_f} \leq 1$ and $\tilde{\alpha} = 2p_f\alpha' \leq 2$, the previous inequality can be rewritten:

$$\mathbb{P}(\{\rho \in \mathsf{SR}_n \mid \rho_f \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_f^{\mathcal{P}(\rho)}} > \tilde{\varepsilon}\}) > \tilde{\alpha} \ .$$

Let $\mathfrak{S}_n'$ be the subset of observed sequences of $\mathfrak{S}_n$ whose correctness proportion is greater than $\tilde{\varepsilon}$ when only considering extensions of ρ_f, but smaller than $\varepsilon^* = \frac{\tilde{\alpha}\tilde{\varepsilon}}{4}$ when considering all faulty runs:

$$\mathfrak{S}_n' = \{\sigma \in \mathfrak{S}_n \mid \frac{p_c^\sigma}{p_c^\sigma + p_f^\sigma} > \tilde{\varepsilon} \wedge \frac{p_c^\sigma}{p_c^\sigma + p_g^\sigma} \leq \varepsilon^*\}.$$

Let $\sigma \in \mathfrak{S}'_n$, $p_f^\sigma < \frac{1-\tilde{\varepsilon}}{\tilde{\varepsilon}} p_c^\sigma$ and $p_c^\sigma \leq \frac{\varepsilon^*}{1-\varepsilon^*} p_g^\sigma$. Therefore $p_f^\sigma < \frac{(1-\tilde{\varepsilon})\varepsilon^*}{(1-\varepsilon^*)\tilde{\varepsilon}} p_g^\sigma$.
Summing over all sequences of $\mathfrak{S}'_n$: $\sum_{\sigma \in \mathfrak{S}'_n} p_f^\sigma < \frac{(1-\tilde{\varepsilon})\varepsilon^*}{(1-\varepsilon^*)\tilde{\varepsilon}} p_g$.
Since $p_g \leq 1$: $\sum_{\sigma \in \mathfrak{S}'_n} p_f^\sigma \leq \frac{(1-\tilde{\varepsilon})\tilde{\alpha}}{4(1-\frac{\tilde{\alpha}\tilde{\varepsilon}}{4})} \leq \frac{\tilde{\alpha}}{2}$.

Thus,

$$\mathbb{P}(\{\rho \in \mathsf{SR}_n \mid \rho_f \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_g^{\mathcal{P}(\rho)}} > \varepsilon^*\}) \geq$$
$$\mathbb{P}(\{\rho \in \mathsf{SR}_n \mid \rho_f \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_f^{\mathcal{P}(\rho)}} > \tilde{\varepsilon}\}) - \sum_{\sigma' \in \mathfrak{S}'_n} p_f^{\sigma'} > \tilde{\alpha} - \frac{\tilde{\alpha}}{2} = \frac{\tilde{\alpha}}{2}.$$

Observe that given $\sigma \in \mathfrak{S}_n$, $\mathsf{CorP}(\sigma) \geq \frac{p_c^\sigma}{p_c^{\sigma'} + p_g^\sigma}$, since we ignore correct runs ρ with $\mathcal{P}(\rho) = \sigma$ that do no extend ρ_c. So defining $\varepsilon = \varepsilon^*$ and $\alpha = \tilde{\alpha}/2$, for all $n_0 \in \mathbb{N}$ there exists $n \geq n_0$:

$$\mathbb{P}(\{\rho \in \mathsf{SR}_n \mid \rho_f \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_g^{\mathcal{P}(\rho)}} > \varepsilon\}) > \alpha \ .$$

Let ρ_0 be the minimal faulty run such that $\rho_0 \preceq \rho_f$. We observe that $\mathsf{Cyl}(\rho_f) \subseteq \mathsf{Cyl}(\rho_0)$, so that

$$\mathbb{P}(\{\rho \in \mathsf{SR}_n \mid \rho_0 \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_g^{\mathcal{P}(\rho)}} > \varepsilon\}) > \alpha$$

which establishes that $\mathcal{A}$ is not AFF-diagnosable. □

Lemma 4.3. *Let $\mathcal{A}$ be a pLTS. If for all $q_0 \stackrel{\rho_c}{\Rightarrow} q_c$ and $q_0 \stackrel{\rho_f}{\Rightarrow} q_f$ with $\mathcal{P}(\rho_c) = \mathcal{P}(\rho_f)$, $q_f \in Q_f$ and $q_c \in Q_c$ belonging to a BSCC, $d(\mathcal{M}(\mathcal{A}_{q_c}), \mathcal{M}(\mathcal{A}_{q_f})) = 1$, then $\mathcal{A}$ is AFF-diagnosable.*

Let ρ_0 be a minimal faulty run, $\alpha > 0, \varepsilon > 0$, $\sigma_0 = \mathcal{P}(\rho_0)$ and $n_0 = |\sigma_0|$. Before developing the proof, we sketch its structure and illustrate it in Figure 4.13. First, we extend the runs with observed sequence σ_0 by n_b observable events where n_b is chosen in order to get a high probability that the runs end in a BSCC.
Let $\sigma \in \Sigma_o^{n_b}$ be such an observed sequence. We partition the possible runs with observed sequence $\sigma_0\sigma$ into three sets $\mathfrak{R}_\sigma^F$, $\mathfrak{R}_{\sigma'}^C$ and $\mathfrak{R}_\sigma^T$. $\mathfrak{R}_\sigma^F$ is the subset of faulty runs while $\mathfrak{R}_\sigma^C$ (resp. $\mathfrak{R}_\sigma^T$) is the set of correct runs ending (resp. not ending) in a BSCC. At first, we do not take into account the transient runs in $\mathfrak{R}_\sigma^T$. We apply Lemma 4.1 to obtain an integer n_σ such that from $\mathfrak{R}_\sigma^F$ and $\mathfrak{R}_\sigma^C$, we can diagnose with (appropriate) high probability and low correctness proportion after n_σ observations. Among the runs that trigger diagnosable observed sequences, some will exceed the correctness proportion, ε, when taking into account the runs from $\mathfrak{R}_\sigma^T$. Yet we show that the probability of such runs is small when cumulated over all extensions σ leading to the required upper bound α.

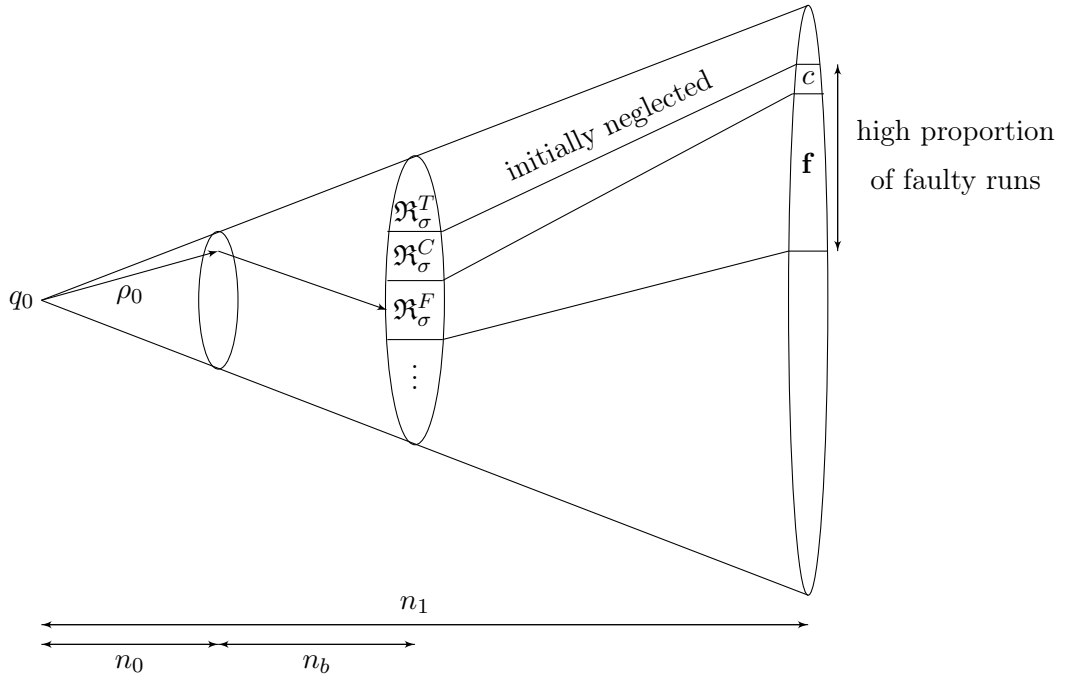


Figure 4.13: Illustration of the proof of Lemma 4.3.

Proof. Let ρ_0 be a minimal faulty run, $\alpha > 0, \varepsilon > 0$, $\sigma_0 = \mathcal{P}(\rho_0)$ and $n_0 = |\sigma_0|$. Since almost surely a random run ends in a BSCC, there exists n_b such that for $\eta = \frac{\alpha\varepsilon}{4}$

$$\mathbb{P}\{\rho \in \mathsf{SR}_{n_0+n_b} \mid \sigma_0 \preceq \mathcal{P}(\rho) \wedge \mathsf{last}(\rho) \text{ does not belong to a BSCC}\} < \eta \ .$$

Let $\mathfrak{S} = \{\sigma \in \Sigma_o^{n_b} \mid \exists \rho \in \mathsf{SR}_{n_0+n_b}\ \rho_0 \preceq \rho \wedge \mathcal{P}(\rho) = \sigma_0\sigma\}$. Pick some $\sigma \in \mathfrak{S}$ and define:

- $\mathfrak{R}_\sigma^F = \{\rho \in \mathsf{SR}_{n_0+n_b} \mid \mathcal{P}(\rho) = \sigma_0\sigma \wedge \mathsf{last}(\rho) \in Q_f\}$;
- $\mathfrak{R}_\sigma^C = \{\rho \in \mathsf{SR}_{n_0+n_b} \mid \mathcal{P}(\rho) = \sigma_0\sigma \wedge \mathsf{last}(\rho) \in Q_c$ and belongs to a BSCC$\}$;
- $\mathfrak{R}_\sigma^T = \{\rho \in \mathsf{SR}_{n_0+n_b} \mid \mathcal{P}(\rho) = \sigma_0\sigma \wedge \mathsf{last}(\rho) \in Q_c$ and does not belong to a BSCC$\}$.

Temporarily, we ignore the runs of $\mathfrak{R}_\sigma^T$. Let $Q_c^\sigma = \{\mathsf{last}(\rho) \mid \rho \in \mathfrak{R}_\sigma^C\}$ and $Q_f^\sigma = \{\mathsf{last}(\rho) \mid \rho \in \mathfrak{R}_\sigma^F\}$. For every pair $(q_f, q_c) \in Q_f^\sigma \times Q_c^\sigma$, consider the initial-fault pLTS $\mathcal{A}' = \langle q_0', \mathcal{A}_{q_f}, \mathcal{A}_{q_c}\rangle$ for some new state q_0', and denote $\mathbb{P}'$ its associated probability measure. Due to Lemma 4.1, for all $\alpha' > 0$, $\varepsilon' > 0$, there exists n_{q_f,q_c} such that for all $n \geq n_{q_f,q_c}$:

$$\mathbb{P}'\{\rho \in \mathsf{SR}_n \mid q_0'\mathbf{f}q_f \preceq \rho \wedge \frac{{p'}_c^{\mathcal{P}(\rho)}}{{p'}_c^{\mathcal{P}(\rho)} + {p'}_f^{\mathcal{P}(\rho)}} > \varepsilon'\} \leq \alpha'$$

where ${p'}_c^{\mathcal{P}(\rho)}$ (resp. ${p'}_f^{\mathcal{P}(\rho)}$) is the probability in $\mathcal{A}'$ of a correct (resp. faulty) run with observed sequence $\mathcal{P}(\rho)$.

Define in $\mathcal{A}$, $p_c^{\mathcal{P}(\rho)}$ (resp. $p_f^{\mathcal{P}(\rho)}$) to be the probability of a correct (resp. faulty) run with observed sequence $\mathcal{P}(\rho)$, $p_f = \min(\mathbb{P}(\rho) \mid \rho \in \mathfrak{R}_\sigma^F)$ and $p_c = \sum_{\rho \in \mathfrak{R}_\sigma^C} \mathbb{P}(\rho)$. By a worst-case reasoning, one gets ${p'}_c^{\mathcal{P}(\rho)} \geq \frac{2}{p_c} p_c^{\sigma_0 \sigma \mathcal{P}(\rho)}$ and ${p'}_f^{\mathcal{P}(\rho)} \leq \frac{2}{p_f} p_f^{\sigma_0 \sigma \mathcal{P}(\rho)}$. Thus for all $n \geq n_0 + n_b + \max(n_{q_f,q_c})$:

$$\mathbb{P}\{\rho \in \mathsf{SR}_n \mid \exists \rho' \in R_\sigma^F \wedge \rho' \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + \frac{p_c}{p_f} p_f^{\mathcal{P}(\rho)}} > \varepsilon'\} \leq 2\alpha'$$

where the factor 2 takes into account the first transition in $\mathcal{A}'$.

Choosing $\varepsilon' = \frac{\varepsilon p_f}{\varepsilon p_f + (2-\varepsilon) p_c}$ and $\alpha' = \frac{\alpha}{4|\mathfrak{S}|}$, after algebraic operations the previous inequality can be rewritten:

$$\mathbb{P}\{\rho \in \mathsf{SR}_n \mid \exists \rho' \in R_\sigma^F \wedge \rho' \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_f^{\mathcal{P}(\rho)}} > \frac{\varepsilon}{2}\} \leq \frac{\alpha}{2|\mathfrak{S}|} \ .$$

Let $n_\sigma = n_0 + n_b + \max(n_{q_f,q_c} \mid (q_f, q_c) \in Q_f^\sigma \times Q_c^\sigma)$ and $n_1 = \max(n_\sigma \mid \sigma \in \mathfrak{S})$ and consider $n \geq n_1$. Ignoring the runs of $\mathfrak{R}_\sigma^T$, one could detect the fault done in ρ_0 with good accuracy and high probability n_1 steps after it occurred.

We now take into account the runs of $\mathfrak{R}_\sigma^T$. Let $\rho \in \{\rho \in \mathsf{SR}_n \mid \exists \rho' \in \mathfrak{R}_\sigma^F \wedge \rho' \preceq \rho\}$. Define $p_t^{\mathcal{P}(\rho)}$ to be the probability of runs (1) with observed sequence $\mathcal{P}(\rho)$ and (2) extending runs of $\mathfrak{R}_\sigma^T$. Since a correct run with observed sequence $\mathcal{P}(\rho)$ must have a prefix in $\mathfrak{R}_\sigma^T$ or in $\mathfrak{R}_\sigma^C$:

$$\mathsf{CorP}(\mathcal{P}(\rho)) \leq \frac{p_c^{\mathcal{P}(\rho)} + p_t^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_t^{\mathcal{P}(\rho)} + p_f^{\mathcal{P}(\rho)}} \ .$$

Consider the following set of runs:

$$\widetilde{\mathfrak{R}}_\sigma^n = \{\rho \in \mathsf{SR}_n \mid \exists \rho' \in \mathfrak{R}_\sigma^F \wedge \rho' \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)} + p_t^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_t^{\mathcal{P}(\rho)} + p_f^{\mathcal{P}(\rho)}} > \varepsilon \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_t^{\mathcal{P}(\rho)} + p_f^{\mathcal{P}(\rho)}} \leq \frac{\varepsilon}{2}\} \ .$$

For $\rho \in \widetilde{\mathfrak{R}}_\sigma^n$, one gets by algebraic operations, $\frac{2 p_t^{\mathcal{P}(\rho)}}{\varepsilon} > p_f^{\mathcal{P}(\rho)}$.

Thus $\mathbb{P}(\widetilde{\mathfrak{R}}_\sigma^n) < \frac{2\mathbb{P}(\mathfrak{R}_\sigma^T)}{\varepsilon}$ and $\sum_{\sigma \in \mathfrak{S}} \mathbb{P}(\widetilde{\mathfrak{R}}_\sigma^n) < \frac{2 \sum_{\sigma \in \mathfrak{S}} \mathbb{P}(\mathfrak{R}_\sigma^T)}{\varepsilon}$.

Due to the choice of n_b, $\sum_{\sigma \in \mathfrak{S}} \mathbb{P}(\mathfrak{R}_\sigma^T) < \eta$, and we derive $\sum_{\sigma \in \mathfrak{S}} \mathbb{P}(\widetilde{\mathfrak{R}}_\sigma^n) < \frac{2\eta}{\varepsilon} = \frac{\alpha}{2}$.

Summarising for all $n \geq n_1$:

$$
\begin{aligned}
&\mathbb{P}\{\rho \in \mathsf{SR}_n \mid \rho_0 \preceq \rho \wedge \mathsf{CorP}(\mathcal{P}(\rho)) > \varepsilon\} \\
=&\sum_{\sigma \in \mathfrak{S}} \mathbb{P}\{\rho \in \mathsf{SR}_n \mid \rho_0 \preceq \rho \wedge \sigma_0\sigma \preceq \mathcal{P}(\rho) \wedge \mathsf{CorP}(\mathcal{P}(\rho)) > \varepsilon\} \\
\leq&\sum_{\sigma \in \mathfrak{S}} \mathbb{P}\{\rho \in \mathsf{SR}_n \mid \exists \rho' \in \mathfrak{R}_\sigma^F \wedge \rho' \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_f^{\mathcal{P}(\rho)}} > \frac{\varepsilon}{2}\} \\
+&\mathbb{P}\{\rho \in \mathsf{SR}_n \mid \exists \rho' \in \mathfrak{R}_\sigma^F \wedge \rho' \preceq \rho \wedge \frac{p_c^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_f^{\mathcal{P}(\rho)}} \leq \frac{\varepsilon}{2} \wedge \frac{p_c^{\mathcal{P}(\rho)} + p_t^{\mathcal{P}(\rho)}}{p_c^{\mathcal{P}(\rho)} + p_t^{\mathcal{P}(\rho)} + p_f^{\mathcal{P}(\rho)}} > \varepsilon\} \\
\leq&|\mathfrak{S}| \frac{\alpha}{2|\mathfrak{S}|} + \frac{\alpha}{2} = \alpha
\end{aligned}
$$

which establishes the AFF-diagnosability of $\mathcal{A}$. □

As an alternative to the proof of Theorem 4.1, one could mimic the approach by Kiefer and Sistla [KS16] for monitorability. The idea would be, from a pLTS $\mathcal{A}$ to derive two hidden Markov chains, say $\mathcal{H}_c$ and $\mathcal{H}_f$ representing respectively the observed sequences for correct and faulty runs of $\mathcal{A}$. However, to establish that distinguishability of $\mathcal{H}_c$ and $\mathcal{H}_f$ corresponds to AFF-diagnosability essentially relies on the same arguments we used in the above proof (and so this alternative approach would not simplify it). The difficulty lies in that the properties one conditions by to obtain $\mathcal{H}_c$ and $\mathcal{H}_f$, namely always correct or eventually faulty, anticipate on the future behaviour of the system; in contrast, the correctness proportion appearing in the definition of AFF-diagnosability only reasons about the possible behaviours up to the last observation.

2 Verification of the diagnosability

We now study the decidability of the different diagnosability notions for finite pLTS and in the positive case provide the complexity. The characterisations given in Section 1 play an important role in this study. Indeed, when a simple characterisation exists, the diagnosability problem is decidable (an algorithm consists in checking this characterisation). Conversely, when we did not exhibit a characterisation, we show that the problem is undecidable.

2.1 Decidability results and upper bounds

We start by showing how to check the characterisations defined in Section 1, therefore providing upper bounds to some of the diagnosability problems. We first consider exact diagnosability notions, and establish that they can all be solved in PSPACE. In all cases, to obtain the PSPACE upper-bound, we avoid building explicitly the exponential size product pLTS (that is used in the characterisations) and only explore it on-the-fly.

The three results have similar proofs. As a consequence we first develop the case of FF-diagnosability then we simultaneously deal with both FA and IA-diagnosability.

Proposition 4.7. *The* FF*-diagnosability problem is decidable in* PSPACE*.*

The proof consists in designing a PSPACE algorithm to check the characterisation given in Proposition 4.2. This algorithm exploits Savitch's Theorem [Sav70] which establishes that PSPACE = NPSPACE. This theorem allow us to use non-determinism in our decision procedure.

Proof. To obtain a PSPACE algorithm, we cannot build explicitly the product pLTS $\mathcal{A}_{\mathsf{FF}}$, which is exponential in the size of $\mathcal{A}$. Given two states s, s' of $\mathcal{A}_{\mathsf{FF}}$, one can check in polynomial space in the size of $\mathcal{A}$ whether s' can be reached from s. Indeed, reachability of a state is known to be in non-deterministic logarithmic space in the size of the system, thus here NPSPACE, which is equal to PSPACE thanks to Savitch's Theorem. Using this procedure, we can check whether a state s is not in a BSCC by guessing another state s' such that s' is reachable from s but s is not reachable from s'. Here again we apply Savitch's Theorem.
Thus the procedure that decides whether $\mathcal{A}$ is not FF-diagnosable consists in guessing a state $s = (q, U)$ with $q \in Q_f$ and $U \neq \emptyset$, checking that it is reachable from s_0 and whether s belongs to a BSCC. □

We state below similar results for FA and IA-diagnosability problems.

Proposition 4.8. *The* FA- *and* IA*-diagnosability problems are decidable in* PSPACE*.*

The two proofs are similar to the proof of Proposition 4.7: we design a decision procedure which uses non-determinism to check the characterisation given in the previous section and the non-determinism is removed using Savitch's Theorem.

Proof. We first check the characterisation of FA-diagnosability given in Proposition 4.4 without explicitly building the product pLTS $\mathcal{A}_{\mathsf{FA}}$. First given a state (q, U, V) of $\mathcal{A}_{\mathsf{FA}}$ we can check in polynomial space whether it belongs to a BSCC (as in the proof of Proposition 4.7). We can also check in polynomial space whether some state (q', U', V') with $U' = \emptyset$ or $V' = \emptyset$ can be reached from (q, U, V) by guessing such a state and then checking the reachability condition. Combining the two, this provides a polynomial space algorithm to check whether (q, U, V) belongs to a BSCC in which no state (q', U', V') fulfils $U' \neq \emptyset$ and $V' \neq \emptyset$. Thus the procedure that decides whether $\mathcal{A}$ is not FA-diagnosable consists in guessing a state $s = (q, U, V)$, checking that it is reachable from s_0 and belongs to a BSCC where all states (q', U', V') fulfil $U' \neq \emptyset$ and $V' \neq \emptyset$.

We use the characterisation of IA-diagnosability given in Proposition 4.5 without building explicitly the product pLTS $\mathcal{A}_{\mathsf{IA}}$. First, given a state (q, U, V, W) of $\mathcal{A}_{\mathsf{IA}}$, we can check in polynomial space that it belongs to a BSCC (as in the proof of Proposition 4.7). We can also check in polynomial space whether it is coreachable from a state (q', U', V', W') that fulfils $U' = \emptyset$ or $W' = \emptyset$ by guessing such a state. Combining the two procedures, we can check in polynomial space whether (q, U, V, W) belongs to a BSCC where all states (q', U', V', W') of the BSCC fulfil $U' \neq \emptyset$ and $W' \neq \emptyset$. Thus the procedure that decides whether $\mathcal{A}$ is not IA-diagnosable consists in guessing a state $s = (q, U, V, W)$, checking that it is reachable from the initial state s_0 and belongs to a BSCC where all states (q', U', V', W') fulfil $U' \neq \emptyset$ and $W' \neq \emptyset$. □

For approximate diagnosability, we focus on AFF-diagnosability and establish a complexity upper-bound, relying on the characterisation from the previous section.

Theorem 4.2. *The* AFF*-diagnosability problem is decidable in* PTIME *for pLTS.*

Proof. The decidability and complexity results rely on the characterisation of AFF-diagnosability showed in Theorem 4.1. Reachability of a pair of states with the same observed sequence is decidable in NLOGSPACE by an appropriate "self-synchronised product" of the pLTS that we detail below. Since there are at most a quadratic number of pairs to check, and given that the distance 1 problem can be decided in polynomial time due to [CK14] (as recalled in Proposition 4.6), the decidability and PTIME upper-bound follow.

We now define the appropriate "self-synchronised product" of the pLTS mentioned above. Given a pLTS $\mathcal{A} = \langle Q, q_0, \Sigma, T, \mathbf{P}\rangle$ we build the product LTS $\mathcal{A} \otimes \mathcal{A} = \langle Q \times Q, \{q_0, q_0\}, \Sigma, T'\rangle$ where $((q_1, q_2), a, (q_1', q_2')) \in T'$ if

- if $a \in \Sigma_u$ then either $(q_1, a, q_1') \in T$ and $q_2' = q_2$ or $(q_2, a, q_2') \in T$ and $q_1' = q_1$;
- else $(a \in \Sigma_o)$, $(q_1, a, q_1') \in T$ and $(q_2, a, q_2') \in T$.

A pair of state (q, q') is reachable in $\mathcal{A} \otimes \mathcal{A}$ from (q_0, q_0) if and only if there exist two runs ρ and ρ' of $\mathcal{A}$ such that $\mathsf{last}(\rho) = q$, $\mathsf{last}(\rho') = q'$ and $\mathcal{P}(\rho) = \mathcal{P}(\rho')$. Therefore, $\mathcal{A}$ is AFF-diagnosable if for any pair of states reachable in $\mathcal{A} \otimes \mathcal{A}$, (q, q'), with q correct, belonging to a BSCC of $\mathcal{A}$, and q' faulty, then $d(\mathcal{M}(\mathcal{A}_q), \mathcal{M}(\mathcal{A}_{q'})) = 1$. All tests can be checked in PTIME. □

We thus have decidability of every notions of exact diagnosability and of one notion of approximate diagnosability, AFF-diagnosability. Surprisingly, AFF-diagnosability, which definition seems more complicated as it depends on the exact probabilistic values of the transitions contrary to the definitions of the exact notions of diagnosability, has a lower upper bound.

2.2 Hardness of Diagnosability

We gave upper bounds on the complexity of diagnosability in Subsection 2.1. We now provide tight lower bounds: on the one hand we establish undecidability of the approximate diagnosability notions that were not characterised, and on the other hand we provide a PSPACE lower bound for the exact diagnosis.

2.2.1 Undecidability results

After having previously proved that AFF-diagnosability can be solved in polynomial time, we now establish that all other specifications of approximate diagnosability are undecidable. This result could be expected for εFF-diagnosability and uniform εFF-diagnosability since it is often the case for problems mixing probabilities, partial ob-

servation and quantitative requirement (here represented by ε)[4]. On the contrary, the undecidability of the *uniform* AFF-diagnosability problem is at first sight surprising since it is a slight variation of the AFF-diagnosability problem. In fact the reduction for the latter problem is more intricate than the one for the εFF- and uniform εFF-diagnosability. We reduce the emptiness problem for probabilistic automata [Paz71] to both problems. Let us first details this problem. A probabilistic automaton is an automaton enhanced with probabilities on the transitions so that given a state and a letter, the sum of the probabilities of the transition exiting each state and labelled by the given letter is 1.

Example 4.12. *Figure 4.14 represents a probabilistic automaton which initial state is q_0 and set of accepting states is $\{q_2\}$. The sum of the probability of the transitions exiting q_0 is 2 as there is a transition labelled by a 'a' and a transition labelled by a 'b'. The word bab has probability $1/2$ to end in q_2 and $1/2$ to end in q_1.*

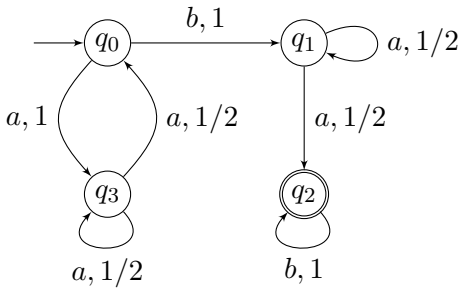


Figure 4.14: An example of probabilistic automaton.

Definition 4.6. *A* probabilistic automaton *is a tuple $A = \langle Q, \delta_0, \Sigma, (\mathbf{P}_a)_{a\in\Sigma}, F\rangle$ where*

- *Q is a set of states with $F \subseteq Q$, the set of final states;*
- *$\delta_0 \in \mathsf{Dist}(Q)$ is the initial distribution;*
- *Σ is an alphabet;*
- *For every $a \in \Sigma$, $\mathbf{P}_a$ is a stochastic $Q \times Q$ matrix.*

When $\mathbf{P}_a[q, q'] > 0$, there is a transition from q to q' labelled by a and $\mathbf{P}_a[q, q']$. Given a word $w = a_1 \dots a_n \in \Sigma^*$, the acceptance probability of w, $\mathbf{P}_A(w)$ is defined by $\mathbf{P}_A(w) = \sum_{q_0\in Q} \delta_0(q_0) \sum_{q\in F} \mathbf{P}_w[q_0, q]$ where $\mathbf{P}_w = \mathbf{P}_{a_1} \cdots \mathbf{P}_{a_n}$. Given a rational threshold $0 < \varepsilon < 1$, the language $\mathcal{L}_{A,\varepsilon}$ is defined by $\mathcal{L}_{A,\varepsilon} = \{w \in \Sigma^* \mid \mathbf{P}_A(w) > \varepsilon\}$. For a probabilistic automaton A and a threshold ε, the emptiness problem asks whether

[4]one of the most famous example is the undecidability of the emptiness problem for probabilistic automaton [Paz71] that we detail below, see also [MHC03] for some examples of undecidable problems for partially observable Markov decision process, a formalism which also contains a form of control of the system

$\mathcal{L}_{A,\varepsilon} = \emptyset$. This problem is undecidable even for a fixed $0 < \varepsilon < 1$ [Paz71]. The problem is also undecidable for the language defined by $\{w \in \Sigma^* \mid \mathbf{P}_A(w) \geq \varepsilon\}$, *i.e.* when the inequality is not strict. One can also restrict oneself to automata such that every word w satisfies $1/4 \leq \mathbf{P}_A(w) \leq 3/4$. This can be ensured by a simple construction. Given a probabilistic automaton A with initial distribution δ_0 and a threshold $0 < \lambda < 1$, we build the probabilistic automaton A' that contains the states of A and two additional states q_a and q_r. The new initial distribution goes in q_a and q_r with probability $1/4$ and with probability $1/2$ it uses the initial distribution of A. In the last case, the behaviour is the one of A, from q_r and q_a everything can be observed with a self-loop and q_a is a final state. For every word w, we have

$$\mathbf{P}_{A'}(w) = \frac{\mathbf{P}_A(w)}{2} + \frac{1}{4}.$$

This is the sum of the probability to be accepted after starting initially in the A component of A' plus the probability to go in q_a. As a consequence, every word is accepted in A' with probability between $1/4$ and $3/4$ and a word is above the λ threshold in A iff it is above $1/4 + \lambda/2$ in A'. Thus the emptiness is also undecidable with this restriction. Note that this assumption relies on the use of an initial distribution δ_0 instead of an initial state q_0. Indeed, with an initial state, the word ε would have probability 0 or 1. When this assumption is not needed, we use an initial state instead of an initial distribution. Another important undecidable problem that is used in a latter chapter is the value 1 problem. It asks for a probabilistic automaton A if for all $\varepsilon > 0$ $\mathcal{L}_{A,1-\varepsilon} \neq \emptyset$. In other words, does there exists words of arbitrarily high probability? This problem is known to be undecidable [GO10].

Theorem 4.3. *For any rational $0 < \varepsilon < 1$, the εFF-diagnosability and uniform εFF-diagnosability problems are undecidable for pLTS.*

We make a reduction from the emptiness problem of probabilistic automata. From a given probabilistic automaton $A = \langle Q, \delta_0, \Sigma, (\mathbf{P}_a)_{a\in\Sigma}, F\rangle$, we build a pLTS which produces with probability 1 runs whose observed sequence belongs to $\Sigma^*\sharp^\omega$ (where $\sharp \notin \Sigma$) and for all $n \geq 2$, the correctness proportion CorP of $w\sharp^n$, with $w \in \Sigma^*$, satisfies $\mathsf{CorP}(w\sharp^n) = \mathbf{P}_A(w)$. In other words, if a word w is accepted with probability greater than ε, then the ambiguity of the word $w\sharp^2$ is greater than ε and every faulty run with this observation will remain ambiguous.

Proof. Let $A = \langle Q, \delta_0, \Sigma, (\mathbf{P}_a)_{a\in\Sigma}, F\rangle$ be a probabilistic automaton. W.l.o.g we assume $1/4 \leq \mathbf{P}_A(w) \leq 3/4$ for every $w \in \Sigma^*$. Define the pLTS $\mathcal{A} = \langle Q', q_0, \Sigma', T', \mathbf{P}'\rangle$ as follows:

- $\Sigma' = \Sigma \uplus \{\sharp, \mathbf{f}, u\}$, $\Sigma'_u = \{\mathbf{f}, u\}$;
- $Q' = Q \cup \{q_0, q_c^\sharp, q_f^\sharp, f^\sharp\}$;
- $T' = \{(q_0, u, q) \mid q \in Q, \delta_0(q) > 0\} \cup \{(q, a, q) \mid q, q' \in Q, a \in \Sigma, \mathbf{P}_a[q, q'] > 0\} \cup \{(q, \sharp, q_c^\sharp \mid q \in F\} \cup \{(q, \sharp, q_f^\sharp \mid q \in Q \setminus F\} \cup \{q_c^\sharp, \sharp, q_c^\sharp\} \cup \{q_f^\sharp, \mathbf{f}, f^\sharp\} \cup \{f^\sharp, \sharp, f^\sharp\}$

- $\mathbf{P}'$ is defined by:
  - For all $q \in Q$ such that $\delta_0(q) > 0$, $\mathbf{P}'(q_0, u, q) = \delta_0(q)$;
  - For all $q \in Q$ and $a \in \Sigma$, $\mathbf{P}'(q, a, q') = \frac{\mathbf{P}_a[q,q']}{1+|\Sigma|}$;
  - For all $q \in F$, $\mathbf{P}'(q, \sharp, q_c^\sharp) = \frac{1}{1+|\Sigma|}$;
  - For all $q \in Q \setminus F$, $\mathbf{P}'(q, \sharp, q_f^\sharp) = \frac{1}{1+|\Sigma|}$;
  - $\mathbf{P}'(q_f^\sharp, \mathbf{f}, f^\sharp) = \mathbf{P}'(f^\sharp, \sharp, f^\sharp) = \mathbf{P}'(q_c^\sharp, \sharp, q_c^\sharp) = 1$.

This reduction is illustrated in Figure 4.15. In each state, the sum of the probabilities of the exiting transitions correctly sum to 1. For instance, let $q \in F$ and $a \in \Sigma$, then $\sum_{q' \in Q} \mathbf{P}_a[q, q'] = 1$, thus:

$$\sum_{(q,a,q') \in T'} \mathbf{P}'(q, a, q') = \sum_{a \in \Sigma} \sum_{q' \in Q} \frac{\mathbf{P}_a[q, q']}{1 + |\Sigma|} + \mathbf{P}'(q, \sharp, q_c^\sharp) = \frac{|\Sigma|}{1 + |\Sigma|} + \frac{1}{1 + |\Sigma|} = 1.$$

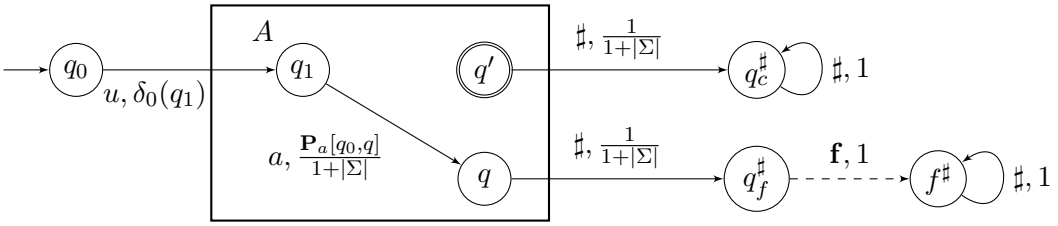


Figure 4.15: From probabilistic automata to pLTS.

We claim that the following three assertions are equivalent:

1. $\mathcal{A}$ is εFF-diagnosable;
2. $\mathcal{A}$ is uniformly εFF-diagnosable;
3. $\mathcal{L}_{A,\varepsilon} = \emptyset$.

Given that uniform εFF-diagnosability entails εFF-diagnosability, we only show that item 1 implies item 3, and item 3 implies item 2. The first implication is proved by contraposition.

1 implies 3 Assume that there exists a word $w \in \Sigma^*$ such that $\mathbf{P}_A(w) > \varepsilon$. Consider the set of signalling correct runs with observed sequence $w\sharp^{n+2}$. By construction, its probability is $\frac{\mathbf{P}_A(w)}{(1+|\Sigma|)^{|w|+1}}$. Similarly, the set of signalling faulty runs with observed sequence $w\sharp^{n+2}$ has probability $\frac{1-\mathbf{P}_A(w)}{(1+|\Sigma|)^{|w|+1}}$. Thus $\mathsf{CorP}(w\sharp^{n+2}) = \mathbf{P}_A(w) > \varepsilon$. By assumption on A, $\mathbf{P}_A(w) \leq 3/4 < 1$, so that the set of faulty runs with

observed sequence $w\sharp^{n+2}$ is non-empty. Pick ρ a minimal faulty run with observed sequence $w\sharp\sharp$. Using the above probability values, for every $n \geq 0$:

$$\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathsf{CorP}(\mathcal{P}(\rho')) > \varepsilon\}) = \mathbb{P}(\rho) \ .$$

Thus $\mathcal{A}$ is not εFF-diagnosable.

3 implies 2 Assume that for every word $w \in \Sigma^*$, $\mathbf{P}r_A(w) \leq \varepsilon$. Let ρ be a minimal faulty run of $\mathcal{A}$. By construction, its observed sequence is of the form $w\sharp^2$ with $w \in \Sigma^*$. Using the same reasoning as above, for every $\rho \preceq \rho'$: $\mathsf{CorP}(\mathcal{P}(\rho')) = \mathbf{P}r_A(w)$, and thus $\mathsf{CorP}(\mathcal{P}(\rho')) \leq \varepsilon$. Therefore, for any $\alpha > 0$, choosing $n_\alpha = 0$, one gets:

$$\mathbb{P}(\{\rho' \in \mathsf{SR}_{n_\alpha+|\rho|} \mid \rho \preceq \rho' \wedge \mathsf{CorP}(\mathcal{P}(\rho')) > \varepsilon\}) = 0 \ .$$

So $\mathcal{A}$ is uniformly εFF-diagnosable.

This completes the proof that εFF-diagnosability and uniform εFF-diagnosability are undecidable. □

Uniform AFF-diagnosability is also shown to be undecidable by a reduction from the emptiness problem for probabilistic automata.

Theorem 4.4. *The uniform* AFF*-diagnosability problem is undecidable for pLTS.*

As this reduction is more involved, we start by a developed sketch of proof and then give the full proof. We proceed by a reduction from the emptiness problem for probabilistic automata where w.l.o.g. one assumes that the acceptance probability of any word lies between 1/4 and 3/4. Given such a probabilistic automaton one builds a pLTS as follows.

- With probability 1/2 one enters one of the two copies of the automaton whose probabilities are modified in a similar way as in the previous proof.
- In a non-final (resp. final) state of the first (resp. second) copy, one may exit the copy of the automaton by taking a transition labelled by $\flat$ (resp. $\mathbf{f}$) and enter a terminating state. In a final state (resp. non-final) state of the first (resp. second) copy, one may "restart" the copy of the automaton by taking a transition labelled by $\sharp$ which lead to the initial state of the copy.
- The terminating state of the first copy iteratively outputs with probability 1/2 $\sharp$ or $\flat$ while the terminating block of the second copy endlessly outputs $\flat$.

Due to the behaviour of the terminating blocks, the correctness proportion of a faulty run goes to 0 as its length increases. Thus the pLTS is AFF-diagnosable. The element that will depend on the probabilistic automaton is the uniformity of the convergence.

Observe that the language of the observed sequences of minimal faulty runs extended by one transition is $(\Sigma^*\sharp)^*\Sigma^*\flat$.

Assume there exists a word w with acceptance probability strictly greater than $1/2$. Then in $\mathcal{A}$, the correctness proportion of $(w\sharp)^n\flat$ fulfils: $\lim_{n\to\infty} \mathsf{CorP}((w\sharp)^n\flat) = 1$. Due to this property (and the behaviours of the terminating blocks), the pLTS is not uniformly AFF-diagnosable. If no such word exists, then for any $w = w_1\sharp w_2\sharp \dots w_k\flat$, $\mathsf{CorP}(w) \leq 3/4$. Due to this property (and the behaviours of the terminating blocks), the pLTS is uniformly AFF-diagnosable.

We now develop the full proof.

Proof. Let $A = \langle Q, \delta_0, \Sigma, (\mathbf{P}_a)_{a\in\Sigma}, F\rangle$ be a probabilistic automaton such that the acceptance probability of any word lies between $1/4$ and $3/4$. Define the pLTS $\mathcal{A} = \langle Q', q'_0, \Sigma', T', \mathbf{P}'\rangle$ as follows.

- $\Sigma' = \Sigma \uplus \{\sharp, \flat, u, \mathbf{f}\}$, $\Sigma'_{uo} = \{u, \mathbf{f}\}$;
- $Q' = \{q^u, q^f \mid q \in Q\} \cup \{q'_0, b^u, b^f\}$;
- $$\begin{aligned} T' = {} & \{(q'_0, u, q^u) \mid \delta_0(q) > 0\} \cup \{(q'_0, u, q^f) \mid \delta_0(q) > 0\} \cup \{(q^u, \sharp, q_0^u) \mid q \in F, \delta_0(q) > 0\} \\ & \cup \{(q^f, \sharp, q^f) \mid q \in Q \setminus F, \delta_0(q) > 0\} \cup \{(q^u, \flat, b^u) \mid q \in Q \setminus F\} \\ & \cup \{b^u, \sharp, b^u\} \cup \{b^u, \flat, b^u\} \cup \{b^f, \flat, b^f\} \cup \{(q^f, \mathbf{f}, b^f) \mid q \in F\} \\ & \cup \{(q^u, a, q'^u), (q^f, a, q'^f) \mid q, q' \in Q, a \in \Sigma, \mathbf{P}_a[q, q'] > 0\} \end{aligned}$$
- $\mathbf{P}'$ is defined by:
  - For all $q \in Q$ with $\delta_0(q) > 0$), $\mathbf{P}'(q'_0, u, q^u) = \mathbf{P}'(q'_0, u, q^f) = \frac{\delta_0(q)}{2}$;
  - For all $(q^u, a, q'^u) \in T'$, $\mathbf{P}'(q^u, a, q'^u) = \frac{\mathbf{P}_a[q,q']}{1+|\Sigma|}$;
  - For all $(q^f, a, q'^f) \in T'$, $\mathbf{P}'(q^f, a, q'^f) = \frac{\mathbf{P}_a[q,q']}{1+|\Sigma|}$;
  - For all $(q^u, \sharp, q_0^u) \in T'$, $\mathbf{P}'(q^u, \sharp, q_0^u) = \frac{\delta_0(q_0)}{1+|\Sigma|}$;
  - For all $(q^f, \sharp, q_0^f) \in T'$, $\mathbf{P}'(q^f, \sharp, q_0^f) = \frac{\delta_0(q_0)}{1+|\Sigma|}$;
  - For all $(q^u, \flat, b^u) \in T'$, $\mathbf{P}'(q^u, \flat, b^u) = \frac{1}{1+|\Sigma|}$;
  - For all $(q^f, \mathbf{f}, b^f) \in T'$, $\mathbf{P}'(q^f, \mathbf{f}, b^f) = \frac{1}{1+|\Sigma|}$;
  - $\mathbf{P}'(b^u, \flat, b^u) = \mathbf{P}'(b^u, \sharp, b^u) = \frac{1}{2}$;
  - $\mathbf{P}'(b^f, \flat, b^f) = 1$.

This reduction is illustrated in Figure 4.16. In each state, the sum of the probabilities of the exiting transitions correctly sum to 1. For instance, let $q \in Q$,

$$\sum_{(q^f,a,q')\in T'} \mathbf{P}'(q^f, a, q') = \sum_{a\in\Sigma} \sum_{q'\in Q} \frac{\mathbf{P}_a[q, q']}{1+|\Sigma|} + \frac{1}{1+|\Sigma|} = \frac{|\Sigma|}{1+|\Sigma|} + \frac{1}{1+|\Sigma|} = 1.$$

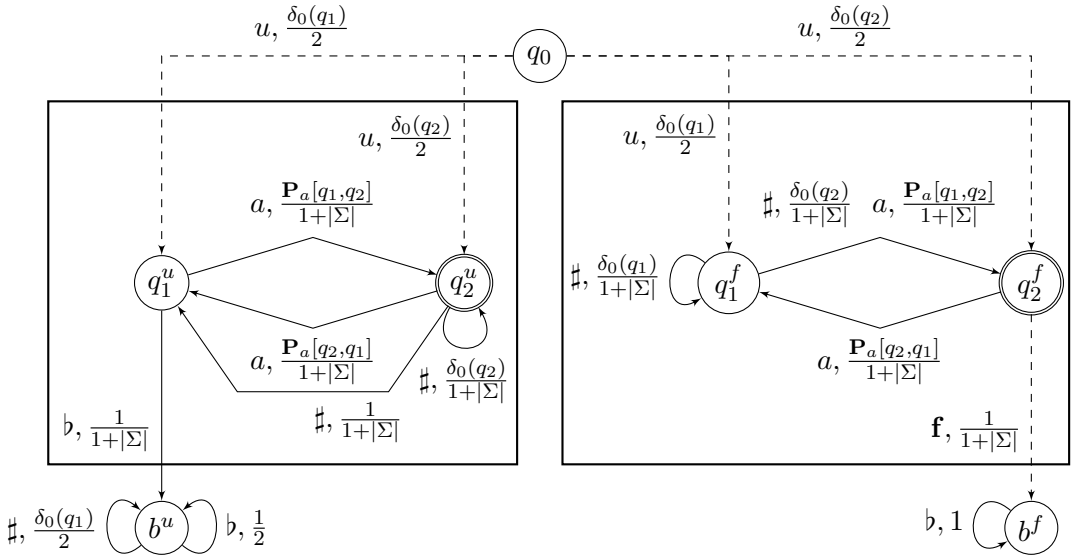


Figure 4.16: From probabilistic automata to pLTS: rectangles surround the two copies of the state space of the probabilistic automaton.

We claim that $\mathcal{A}$ is uniformly AFF-diagnosable if and only if $\mathcal{L}_{A,1/2} = \emptyset$.
Observe first that for all $q \in Q$, $\mathcal{L}^\omega(\mathcal{A}_{q^f}) \subseteq \mathcal{L}^\omega(\mathcal{A}_{q^u})$ so that all faulty runs are ambiguous.
• Assume that there exists a word $w \in \Sigma^*$ such that $\mathbf{P}_A(w) > 1/2$. We prove that $\mathcal{A}$ is not uniformly $\frac{1}{2}\mathsf{FF}$-diagnosable. So we pick arbitrary $0 < \alpha < 1$ and n_α.
Consider the observed sequence $\sigma_n = (w\sharp)^n\flat$ for some n to be fixed later. As every word is accepted with positive probability by A, it is ambiguous. Let

$$\gamma_n = \frac{\mathbb{P}(\{\rho' \in \mathsf{C} \mid \mathcal{P}(\rho') = \sigma_n\})}{\mathbb{P}(\{\rho' \in \mathsf{F} \mid \mathcal{P}(\rho') = \sigma_n\})} \ .$$

Since $\mathbf{P}_A(w) > 1/2$, γ_n fulfils $\lim_{n\to\infty} \gamma_n = \infty$.
Let ρ_n be a minimal faulty run with $\mathcal{P}(\rho_n) = \sigma_n$. Let ρ be a signalling run extending ρ_n with $|\rho|_o = |\rho_n|_o + n_\alpha$. Then $\mathcal{P}(\rho) = \sigma_n\flat^{n_\alpha}$. By a straightforward examination of $\mathcal{A}$ one gets:

$$\frac{\mathbb{P}(\{\rho' \in \mathsf{C} \mid \mathcal{P}(\rho') = \sigma_n\flat^{n_\alpha}\})}{\mathbb{P}(\{\rho' \in \mathsf{F} \mid \mathcal{P}(\rho') = \sigma_n\flat^{n_\alpha}\})} = \frac{\gamma_n 2^{-n_\alpha}}{1 + \gamma_n 2^{-n_\alpha}} \ .$$

Choosing n such that $\gamma_n 2^{-n_\alpha} > 1$, one gets: $\mathsf{CorP}(\rho) > 1/2$. So:

$$\mathbb{P}(\{\rho \in \mathsf{SR}_{n_\alpha+|\rho_n|_o} \mid \rho_n \preceq \rho \wedge \mathsf{CorP}(\mathcal{P}(\rho)) > \frac{1}{2}\}) = \mathbb{P}(\rho) > \alpha\mathbb{P}(\rho) \ .$$

Thus $\mathcal{A}$ is not uniformly $\frac{1}{2}\mathsf{FF}$-diagnosable.

• Conversely assume that for every word $w \in \Sigma^*$, $\mathbf{P}_A(w) \leq 1/2$. Combining this assumption with the hypothesis that $\mathbf{P}_A(w) \geq 1/4$, one deduces that for every observed sequence $\sigma \in (\Sigma \cup \{\sharp\})^*\flat$, $\mathsf{CorP}(\sigma) \leq 3/4$. On the other hand, for every minimal faulty run ρ, $\mathcal{P}(\rho) \in (\Sigma \cup \{\sharp\})^*\flat$.
Pick any positive ε, α and consider an arbitrary minimal faulty run ρ. The observed sequence σ' of a faulty run ρ' that extends ρ fulfils $\sigma' = \mathcal{P}(\rho)\flat^n$ for some n. After a new occurrence of $\flat$ the fraction between the probability of correct runs with observed sequence $\sigma'\flat$ over the probability of faulty runs with observed sequence $\sigma'\flat$ is halved. Thus choosing n_α such that $\frac{3 \cdot 2^{-n_\alpha}}{1+3 \cdot 2^{-n_\alpha}} \leq \varepsilon$, for all $n \geq n_\alpha$:

$$\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathsf{CorP}(\mathcal{P}(\rho)) \leq \varepsilon) = \mathbb{P}(\rho) \geq (1-\alpha)\mathbb{P}(\rho) \ .$$

Thus $\mathcal{A}$ is uniformly εFF-diagnosable and since ε was chosen arbitrarily, $\mathcal{A}$ is uniformly AFF-diagnosable. □

The undecidability of (uniform) εFF- and of uniform AFF-diagnosability are due to different reasons. For (uniform) εFF-diagnosability, it is mainly caused by the use of a quantitative requirement (shown through the use of ε). For the uniform AFF-diagnosability, the problem arises as the detection speed of the fault strongly depends on the behaviour of the system before the occurrence of the fault. Limiting the behaviour of the system before the occurrence of a fault can raise decidability results (as in initial-fault pLTS where uniform AFF-diagnosability is diagnosable as it is equivalent to AFF-diagnosability).

Thanks to Proposition 3.5, page 70, and the definition of AFF-diagnosability, we know that a system is AFF-diagnosable iff for all $\varepsilon > 0$ there exists an εFF-diagnoser of the system. Limiting oneself to finite memory can sometimes bring better decidability or complexity results (in [BFH$^+$14] for example). It is therefore natural to question if such a restriction would make uniform AFF-diagnosability decidable. There is several ways to add the finite-memory restriction to AFF-diagnosability:

1. a system is AFF-diagnosable with finite memory if for all $\varepsilon > 0$ there exists a finite-memory εFF-diagnosers;

2. a system is AFF-diagnosable with finite memory if there exists $\lambda > 0$, such that for all $0 < \varepsilon < \lambda$ there exists a finite-memory εFF-diagnosers;

3. a system is AFF-diagnosable with finite memory if there exists a sequence $(\varepsilon_n)_{n\in\mathbb{N}}$ such that $\forall n, \varepsilon_n > 0$, $\varepsilon_n \xrightarrow{n\to\infty} 0$ and for all $n \in \mathbb{N}$ there exists a finite-memory ε_n-diagnosers;

These three notions are however equivalent since if D is a finite-memory εFF-diagnosers for some $\varepsilon > 0$, then D is a finite-memory ε'-diagnoser for all $\varepsilon' > \varepsilon$. Surprisingly, this restriction complexifies the problem as stated in the following proposition.

Proposition 4.9. *The* AFF*-diagnosability with finite memory problem is undecidable for pLTS.*

The proof is obtained thanks to the reduction made in the proof of Theorem 4.4. Indeed, in this particular case, we can show that the uniform property is equivalent to the finite-memory restriction.

Proof. Given a probabilistic automaton A, let $\mathcal{A}$ be the pLTS built in the reduction of the proof of Theorem 4.4. We know that $\mathcal{A}$ is uniformly AFF-diagnosable if and only if $\mathcal{L}_{A,1/2} = \emptyset$. We show that $\mathcal{A}$ is uniformly AFF-diagnosable iff for every $\varepsilon > 0$ there exists a finite-memory $\varepsilon\mathsf{FF}$-diagnoser of $\mathcal{A}$, which establishes the undecidability.

Assume that $\mathcal{A}$ is uniformly AFF-diagnosable. Let $\varepsilon > 0$. By definition of uniform AFF-diagnosability, there exists $n_0 \in \mathbb{N}$ such that for all minimal faulty run $\rho \in \mathsf{minF}$ and all $n \geq n_0$, $\mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}^{\varepsilon}_{n+|\rho|_o}) \leq \frac{\mathbb{P}(\rho)}{2}$. As $\mathsf{Cyl}(\rho)$ is a single infinite run, this means that $\mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}^{\varepsilon}_{n+|\rho|_o}) = 0$. Let D be the diagnoser that counts how many $\flat$ were observed and outputs $\top$ iff this number is above n_0 and no $\sharp$ was observed after the first $\flat$. This diagnoser can be represented with finite memory as it only needs to count up to a fixed value n_0. Moreover it is an $\varepsilon\mathsf{FF}$-diagnoser:

correctness. if $D(w) = \top$ for some $w \in \Sigma_o \cup \{\flat, \sharp\}^*$, then there exists $w' \in \Sigma_o \cup \{\sharp\}^*$ such that $w = w'\flat^{n'}$ with $n' \geq n_0$. As in A, every word is accepted with positive probability, there exists a minimal faulty run ρ with observation w'. Let ρ_f be the infinite run of $\mathsf{Cyl}(\rho)$. By uniformity, $\rho_f \notin \mathsf{FAmb}^{\varepsilon}_{n'+|\rho|_o}$. Moreover, the prefix of length $n' + |\rho|_o$ of ρ_f has observation w. Thus $\mathsf{CorP}(w) \leq \varepsilon$.

reactivity. The n observations following a fault are $\flat$, and no $\sharp$ can be observed after the first $\flat$ in a faulty run. Thus, for all $m \geq n$, $\mathbb{P}(\rho' \in \mathsf{F} \cap \mathsf{SR}_m \mid D(\mathcal{P}(\rho)) =?) = 0$ which implies reactivity.

Conversely, assume that $\mathcal{A}$ is not uniformly AFF-diagnosable. There thus exists $\varepsilon > 0$ such that $\mathcal{A}$ is not uniformly $\varepsilon\mathsf{FF}$-diagnosable. Suppose there exists a finite-memory $\varepsilon\mathsf{FF}$-diagnoser with m memory states. As $\mathcal{A}$ is not uniformly $\varepsilon\mathsf{FF}$-diagnosable, there exists a minimal faulty run ρ such that for all $n \leq m+1$, $\mathbb{P}(\mathsf{Cyl}(\rho) \cap \mathsf{FAmb}^{\varepsilon}_{n+|\rho|_o}) > 0$. Since there exists only one infinite run ρ_f extending ρ, this means $\rho_f \in \mathsf{FAmb}^{\varepsilon}_{n+|\rho|_o}$. Consider the observation $\mathcal{P}(\rho)\flat^{m+1}$. The last $m+1$ memory states visited in the finite-memory diagnoser while reading this observation are denoted $s_1, \dots s_{m+1}$. None of these memory state can claim a fault by correctness of the diagnoser and as $\rho_f \in \mathsf{FAmb}^{\varepsilon}_{n+|\rho|_o}$ for $n \leq m+1$. Moreover, as the finite-memory diagnoser has m states, there exists $i, j \leq m+1$ such that $s_i = s_j$. there thus exists a cycle labelled by a number of $\flat$ in the finite-memory diagnoser. By determinism of the diagnoser, it means that for all $n \in \mathbb{N}$, $D(\mathcal{P}(\rho)\flat^{m+1}) =?$ which contradict the reactivity requirement. There thus does not exists a finite-memory $\varepsilon\mathsf{FF}$-diagnoser. □

2.2.2 PSPACE-hardness of exact diagnosability

In order to establish a lower bound for the complexity of exact diagnosability, we introduce a variant of language universality.

Definition 4.7. *A language $\mathcal{L}$ over an alphabet Σ is said* eventually universal *if there exists a word $v \in \Sigma^*$ such that $v^{-1}\mathcal{L} = \Sigma^*$.*

Several variants of the universality problem were shown to be PSPACE-complete in [RSX12] but, to the best of our knowledge, eventual universality has not been considered.

Because of our diagnosis framework, we focus on live non-deterministic finite automata (NFA). Similarly to pLTS, an NFA is *live* if from every state there is at least one outgoing transition. The language of an NFA A, denoted $\mathcal{L}(A)$, is defined as the set of finite words that are accepted by A.

Proposition 4.10. *Let A be a live NFA where all states are terminal. Then deciding whether $\mathcal{L}(A)$ is eventually universal is* PSPACE*-hard.*

Proof. We reduce the universality problem for NFA, which is known to be PSPACE-complete [MS72] to the eventual universality problem. Let $A = (Q, q_0, \Sigma, T, F)$ be an NFA. Starting from A, one builds in polynomial time the NFA $A' = (Q', q_0, \Sigma', T', Q')$ where $\Sigma' = \Sigma \uplus \{\sharp\}$, $Q' = Q \uplus \{s\}$, and

$$T' = T \cup \{(q, \sharp, q_0) \mid q \in F\} \cup \{(s, a, s) \mid a \in \Sigma\} \cup \{(q, a, s) \mid a \in \Sigma,\ q \not\to_A\}$$

with $q \not\to_A$ meaning that there is no transition exiting q in A. The additional state s and the associated transitions are added to ensure that A' is live, they do not alter the accepted language.

• Assume that $\mathcal{L}(A) = \Sigma^*$. Any word w over the alphabet Σ' can be decomposed into $w = w_1\sharp w_2\sharp \dots \sharp w_n$ with $w_i \in \Sigma^*$. For each factor w_i, since A is universal, there exists a run ρ_i on w_i ending in some terminal state $q_i \in F$. Therefore w is accepted in A' by the run $\rho_1\sharp\rho_2\sharp \dots \sharp\rho_n$. Hence A' is universal, and thus eventually universal: $\varepsilon^{-1}\mathcal{L}(A') = \Sigma'^*$.

• Conversely assume that A' is eventually universal and let $v \in \Sigma'^*$ be such that $v^{-1}\mathcal{L}(A') = \Sigma'^*$. Given $w \in \Sigma^*$, we consider the word $w' = v\sharp w\sharp$. Since A' is eventually universal with witness v, $w' \in \mathcal{L}(A')$ and there exists an accepting run that can be decomposed as: $\rho\sharp\rho'\sharp q_0$. As a $\sharp$ can only be read in a final state and leads to q_0, the run ρ', which corresponds to the word w, has q_0 as initial state, ends in a final state of A, and by construction of A' only uses transitions of A. So ρ' is a run of A that accepts w. Therefore $w \in \mathcal{L}(A)$, and A is universal. □

Now that we established that universal eventuality is PSPACE-hard, we can use it to establish a complexity lower bound for the different exact diagnosability problems.

Proposition 4.11. *The* FF*-diagnosability,* FA*-diagnosability and* IA*-diagnosability problems are* PSPACE*-hard.*

Proof. The proof is by reduction from the eventual universality problem. Let A be a live NFA over Σ, in which all states are final. One builds in polynomial time the initial-fault pLTS $\mathcal{A} = \langle q'_0, \mathcal{A}^f, \mathcal{A}^c \rangle$ depicted in Figure 4.17 where $\Sigma_o = \Sigma \uplus \{\sharp\}$, $\Sigma_u = \{u, \mathbf{f}\}$ and all transitions outgoing a state have the same probability. $\mathcal{A}^f$ consists of a single state on

which any letter of Σ can be read with a self loop. $\mathcal{A}^c$ is a copy of A to which we add a new state $q_\sharp$ to which one can access by a transition labelled by $\sharp$ from any state of the copy of A.

We establish the following two implications (note that they do not use the same diagnosability notion):

- $\mathcal{A}$ is not FA-diagnosable implies A is eventually universal;
- A is eventually universal implies $\mathcal{A}$ is not FF-diagnosable.

Since FA-diagnosability implies IA-diagnosability, which implies IF-diagnosability which is equivalent to FF-diagnosability according to Theorem 3.1, page 73, this proves that all three notions are at least as hard as eventual universality.

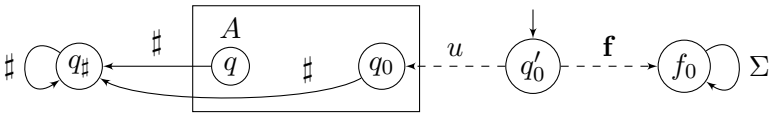


Figure 4.17: A reduction for PSPACE-hardness of IF-, FA- and IA-diagnosability.

• Assume that $\mathcal{A}$ is not FA-diagnosable. By Proposition 4.4, either $\mathcal{A}_{\mathsf{FA}}$ contains a reachable BSCC $\mathcal{C}$ with some state $s = (q, U, V) \in \mathcal{C}$ such that $q \in Q_f$ and $U \neq \emptyset$ or $\mathcal{A}_{\mathsf{FA}}$ contains a reachable BSCC $\mathcal{C}$ with some state $s = (q, U, V) \in \mathcal{C}$ such that $q \in Q_c$ and $V \neq \emptyset$. The latter case is excluded since the only correct state belonging to a BSCC of $\mathcal{A}_{\mathsf{FA}}$ contains $q_\sharp$ as first component and $q_\sharp$ is only reachable by a transition labelled by $\sharp$. As this observation cannot occur in a faulty run, $q = q_\sharp$ implies $V = \emptyset$. Consider the former case: obviously $q = f_0$. Since $\mathcal{C}$ is a BSCC and f_0 is a sink state in $\mathcal{A}$, for every state $s' = (q', U', V') \in \mathcal{C}$, one has $q' = f_0$ and $U' \neq \emptyset$. Since in f_0 all events of Σ are enabled, this implies that for all $w \in \Sigma^*$, there is a correct run ρ_1 in $\mathcal{A}$ starting from some state of $q \in U$ with observed sequence w. Consider an observed sequence $v \in \Sigma^*$ labelling a run in $\mathcal{A}_{\mathsf{FA}}$ from the initial state to s. Then there is correct run in $\mathcal{A}$ from q'_0 to q with observed sequence v. So the run $\rho = \rho_0\rho_1$ has vw as observed sequence. Since $\rho = q'_0 u \rho'$ with ρ' a run of A starting from q_0, $vw \in \mathcal{L}(A)$. This holds for any word w, thus $v^{-1}\mathcal{L}(A) = \Sigma^*$ and A is eventually universal.

• Assume that there exists a word $v \in \Sigma^*$ such that $v^{-1}\mathcal{L}(A) = \Sigma^*$. Of course, any word extending v is also a witness that A is eventually universal. Let $v' \in \Sigma^*$ be such that some faulty run with observed sequence vv' ends in a BSCC $\mathcal{C}$ of $\mathcal{A}_{\mathsf{FF}}$. Since $(vv')^{-1}\mathcal{L}(A) = \Sigma^*$, all states of $\mathcal{C}$ are of the form (f_0, U) with $U \neq \emptyset$. Therefore, by Proposition 4.2, $\mathcal{A}$ is not FF-diagnosable. □

Since the lower bounds matches the upper bounds, the different notions of exact diagnosability are PSPACE-complete for finite pLTS.

3 Diagnoser construction

When a system is shown to be diagnosable, the next step is to build a diagnoser. A diagnoser that works in every case would keep track of every run compatible with the observed sequence and give a verdict depending on the nature of this set of runs. This diagnoser, by nature, uses unbounded memory. For implementation purpose, we are rather interested in finite-memory diagnosers as defined in Section 1 of Chapter 3. We explain how to automatically build a finite-memory diagnoser for a diagnosable system. This is not possible for every notion of diagnosability however. Indeed, we showed in Proposition 3.4, page 69, that $\varepsilon\mathsf{FF}$-diagnosers may need infinite memory. Therefore we do not develop approximate diagnosability here, and focus on exact diagnosability.

3.1 FF-diagnoser

We start with FF-diagnosers. These diagnosers only provide information about faulty runs. In the sequel we fix $\mathcal{A}$ a finite pLTS.

Proposition 4.12. *If $\mathcal{A}$ is an FF-diagnosable pLTS with n correct states, one can build an FF-diagnoser for $\mathcal{A}$ with at most 2^n memory states.*

Proof. The idea of this proof and of all the following proofs for constructing a finite-memory diagnoser is to use the characterisation given in Section 1. For example, in order to construct the FF-diagnoser, we first build the FF-automaton of $\mathcal{A}$. Then we define the finite-memory diagnoser on its structure. Finally we show that the constructed diagnoser is indeed an FF-diagnoser thanks to the FF-diagnosability of the system.

For an FF-diagnosable pLTS $\mathcal{A}$ with $\mathsf{FF}(\mathcal{A}) = (S, s_0, \Delta, F)$, its deterministic and complete FF-automaton, we define the finite memory diagnoser $(S, \Sigma_o, \mathsf{up}, s_0, D_{fm})$ with $\mathsf{up}(s, a) = s'$ if $(s, a, s') \in \Delta$ and $D_{fm}(U) = \top$ iff $U = \emptyset$. Let us show that the induced diagnoser D is indeed an FF-diagnoser, and that it has at most 2^n memory states, where n is the number of correct states of $\mathcal{A}$.

commitment When U is empty, it remains empty forever which implies commitment.

correctness When D outputs the verdict $\top$, $\mathsf{FF}(\mathcal{A})$ is in the state associated with $\emptyset$. As U contains the set of correct states reachable with the current observed sequence, the observed sequence is surely faulty.

reactivity If an infinite faulty run ρ is such that $D(\mathcal{P}(\rho)) =?$ then, by construction of $\mathsf{FF}(\mathcal{A})$ and definition of D, for every length $n \in \mathbb{N}$, there exists a finite correct signalling run $\rho_n \in \mathsf{SR}_n$ such that $\mathcal{P}(\rho_n) = \mathcal{P}(\rho_{\downarrow n})$. Using König's lemma, since $\mathcal{A}$ is finitely branching, one can extract an infinite correct run ρ_∞ such that $\mathcal{P}(\rho_\infty) = \mathcal{P}(\rho)$, so that $\rho \in \mathsf{FAmb}_\infty$. Moreover $\mathbb{P}(\mathsf{FAmb}_\infty) = 0$ as $\mathcal{A}$ is FF-diagnosable. Putting everything together, for every minimal faulty run ρ, $\mathbb{P}(\{\rho' \in \Omega \mid \rho \preceq \rho' \wedge D(\mathcal{P}(\rho')) =?\}) = 0$.

size The memory states are states of $\mathsf{FF}(\mathcal{A})$, which are themselves subsets of correct states of $\mathcal{A}$. Therefore, D uses at most 2^n memory states, with $n = |Q_c|$.

□

We now show that the size order of the previous FF-diagnoser is optimal.

Proposition 4.13. *There is a family* $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ *of* FF*-diagnosable pLTS such that* $\mathcal{A}_n$ *has* $2n+2$ *correct states and it admits no* FF*-diagnoser with less than* 2^n *states.*

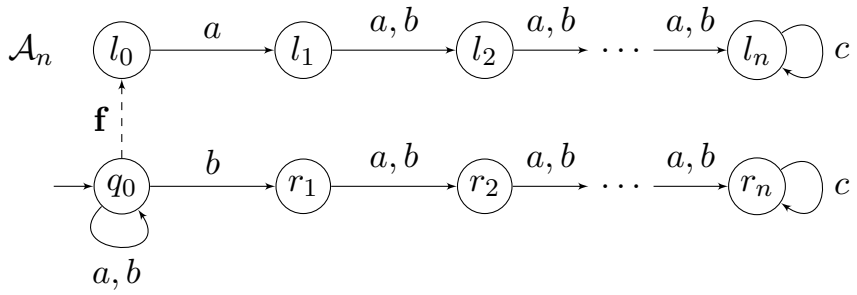


Figure 4.18: An FF-diagnosable pLTS requiring an FF-diagnoser with exponential size.

Proof. This proof is inspired by a similar result of lower bounds on controllers established in [HHMS13]. Consider the example of Figure 4.18 where $\Sigma_o = \{a, b, c\}$ and the initial state is q_0. Consider a finite faulty run including a c event. Its observed sequence belongs to $\mathcal{L} = \{a, b\}^* a \{a, b\}^{n-1} c^+$. Since any finite correct run has an observed sequence belonging to $\mathcal{L}' = \{a, b\}^* \cup \{a, b\}^* b \{a, b\}^{n-1} c^+$ and $\mathcal{L} \cap \mathcal{L}' = \emptyset$, $\mathsf{FAmb}_n \uplus \mathsf{CAmb}_n \subseteq \{\rho \mid \mathcal{P}(\rho) \in \{a, b\}^n\}$. Since $\lim_{n\to\infty} \mathbb{P}(\{\rho \mid \mathcal{P}(\rho) \in \{a, b\}^n\}) = 0$, the pLTS is FA-diagnosable and so IA-diagnosable and FF-diagnosable.

Intuitively, when a c is observed, any FF-diagnoser must have remembered the observable event that happened n steps earlier to know if the run is faulty or not. Thus, it must remember the last n observed events, in case a c event occurs.

More formally, assume there exists a diagnoser $D = (M, \Sigma, m_0, \mathsf{up}, D_{fm})$ with less than 2^n memory states. Then there exist two distinct words $w_1 \in \{a, b\}^n$ and $w_2 \in \{a, b\}^n$ leading to the same memory state: $\mathsf{up}(m_0, w_1) = \mathsf{up}(m_0, w_2)$. The words w_1 and w_2 differ at least from one letter say $w_1[i] = b$ and $w_2[i] = a$. Consider for $k \geq 1$, the signalling correct run $\rho_{1,k}$ corresponding to observed sequence $w_1 a^{i-1} c^k$ whose sequence of visited states is $q_0^i r_1 \dots r_n^{k+1}$ and the signalling faulty run $\rho_{2,k}$ corresponding to observed sequence $w_2 a^{i-1} c^k$ whose sequence of visited states is $q_0^i l_0 l_1 \dots l_n^{k+1}$. They also lead to the same memory state. By correctness, $D(w_1 a^{i-1} c^k) =?$. Thus for all suffixes ρ of $\rho_{2,1}$, $D(\rho) =?$ contradicting the reactivity of D. □

3.2 FA-diagnoser

We now turn to FA-diagnosability which not only considers the diagnosis of faults but also of correct runs. We build the FA-diagnoser from the FA-automaton similarly to what was done in Proposition 4.12.

Proposition 4.14. *A pLTS $\mathcal{A}$ is* FA*-diagnosable if it admits an* FA*-diagnoser. Moreover, if $\mathcal{A}$ is a* FA*-diagnosable pLTS with n states, one can build an* FA*-diagnoser with at most 2^n memory states.*

Proof. Let $\mathcal{A}$ be a pLTS. Assume first that there exists an FA-diagnoser D for $\mathcal{A}$. For every $n \in \mathbb{N}$, we define $\mathsf{FD}_n = \{\rho \in \Omega \mid D(\mathcal{P}(\rho_{\downarrow n})) = \top\}$ the set of runs that are diagnosed faulty after n observed events, and symmetrically $\mathsf{CD}_n = \{\rho \in \Omega \mid \forall m \geq n, D(\mathcal{P}(\rho_{\downarrow m})) = \bot\}$ the set of runs that are persistently diagnosed correct after n observed events. The sequences $(\mathsf{CD}_n)_{n\in\mathbb{N}}$ and $(\mathsf{FD}_n)_{n\in\mathbb{N}}$ are non-decreasing. As $? \prec \bot$ and $? \prec \top$, for every run $\rho \in \Omega$, $D_{\inf}(\mathcal{P}(\rho)) =?$ is equivalent to $\rho \notin \bigcup_n(\mathsf{FD}_n \cup \mathsf{CD}_n)$. Thus we have $\bigcup_{n\in\mathbb{N}}(\mathsf{FD}_n \cup \mathsf{CD}_n) = \{\rho \in \Omega \mid D_{\inf}(\mathcal{P}(\rho)) \neq ?\}$. Since D is reactive, $\mathbb{P}(\{\rho \in \Omega \mid D_{\inf}(\mathcal{P}(\rho)) \neq ?\}) = 1$. Moreover, since D is correct, for every $n \in \mathbb{N}$, $\mathsf{FD}_n \subseteq \mathsf{Sf}_n$ and $\mathsf{CD}_n \subseteq \mathsf{Sc}_n$. Thus for every $n \in \mathbb{N}$, $\mathbb{P}(\mathsf{FAmb}_n \cup \mathsf{CAmb}_n) = 1 - \mathbb{P}(\mathsf{Sf}_n \cup \mathsf{Sc}_n) \leq 1 - \mathbb{P}(\mathsf{FD}_n \cup \mathsf{CD}_n)$ and $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n \cup \mathsf{CAmb}_n) \leq 1 - \liminf_{n\to\infty} \mathbb{P}(\{\rho \in \mathsf{SR}_n \mid D(\mathcal{P}(\rho)) \neq ?\}) = 0$. This shows that $\mathcal{A}$ is FA-diagnosable.

Assume now that $\mathcal{A}$ is FA-diagnosable and has n states. From $\mathsf{FA}(\mathcal{A}) = (S, s_0, \Delta, F)$ the FA-automaton of $\mathcal{A}$, we define $D = (S, \Sigma_o, s_0, \mathsf{up}, D_{fm})$ the finite-memory diagnoser where $\mathsf{up}(s, a) = s'$ if $(s, a, s') \in \Delta$, $D_{fm}((U, V)) = \top$ iff $U = \emptyset$ and $D_{fm}((U, V)) = \bot$ iff $V = \emptyset$. Let us check that D is an FA-diagnoser, and that its size is at most 2^n if n denotes the number of states of $\mathcal{A}$.

commitment When U is empty, it remains empty forever which implies commitment.

correctness Let $w \in \Sigma_o^*$ be an observed sequence. If (U, V) is the state in $\mathsf{FA}(\mathcal{A})$ reached after reading w, then recall that U (resp. V) is the set of states in $\mathcal{A}$ that can be reached by correct (resp. faulty) signalling runs labelled by w. By construction, if $D(w) = \top$ then w is surely faulty, and if $D(w) = \bot$ then w is surely correct.

reactivity Let ρ be a signalling run such that $D(\mathcal{P}(\rho)) =?$. Due to the characterisation of Proposition 4.4, the strictly connected component of $\mathcal{A}_{\mathsf{FA}}$ that ρ has reached cannot be a BSCC. So given some $n \in \mathbb{N}$,

$$\mathbb{P}(\{\rho \in \Omega \mid \exists m \geq n\ D(\mathcal{P}(\rho_{\downarrow m}) =?\}) \leq \mathbb{P}(\{\rho \in \Omega \mid \rho_{\downarrow n} \text{ does not reach a BSCC}\}).$$

Thus

$$\begin{aligned}\mathbb{P}(\{\rho \in \Omega \mid D_{\inf}(\mathcal{P}(\rho)) =?\}) &= \lim_{n\to\infty} \mathbb{P}(\{\rho \in \Omega \mid \exists m \geq n\ D(\mathcal{P}(\rho_{\downarrow m}) =?\}) \\ &\leq \limsup_{n\to\infty} \mathbb{P}(\{\rho \in \Omega \mid \rho_{\downarrow n} \text{ does not reach a BSCC}\}) \\ &= 0\ .\end{aligned}$$

size D has at most 2^n memory states because every state of $\mathsf{FA}(\mathcal{A})$ consists of a pair (U, V) with $U \subseteq Q_c$ and $V \subseteq Q_f$.

□

As the pLTS of Figure 4.18 is FA-diagnosable, and since any FA-diagnoser is also an FF-diagnoser, using Proposition 4.13 we obtain the following lower bound for the size of FA-diagnosers.

Proposition 4.15. *There is a family* $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ *of* FA*-diagnosable pLTS such that* $\mathcal{A}_n$ *has* $2n+2$ *states and it admits no* FA*-diagnoser with less than* 2^n *memory states.*

3.3 IA-diagnoser

We end with IA-diagnosability and build an IA-diagnoser similarly as to what was done in Proposition 4.12.

Proposition 4.16. *If* $\mathcal{A}$ *is a* IA*-diagnosable pLTS with* n_c *correct states and* n_f *faulty states, one can build an* IA*-diagnoser with at most* $2^{n_c}3^{n_f}$ *states.*

Proof. Let $\mathcal{A}$ be an IA-diagnosable pLTS. From $\mathsf{IA}(\mathcal{A}) = (S, s_0, \Delta, F)$ the IA-automaton of $\mathcal{A}$, we define $D = (S, \Sigma_o, s_0, \mathsf{up}, D_{fm})$ the finite-memory diagnoser where $\mathsf{up}(s,a) = s'$ if $(s,a,s') \in \Delta$, $D_{fm}((U,V,W)) = \top$ iff $U = \emptyset$ and $D_{fm}((U,V,W)) = \bot$ iff $W = \emptyset$. Let us prove that D is indeed an IA-diagnoser for $\mathcal{A}$.

commitment. When U is empty, it remains empty forever which implies commitment.

correctness. For any word $w \in \Sigma_o^*$, we denote by (U_w, V_w, W_w) the state in $\mathsf{IA}(\mathcal{A})$ reached after reading w. For any word w, if $U_w = \emptyset$, by construction of $\mathsf{IA}(\mathcal{A})$, w is surely faulty. Assume now that $W_w = \emptyset$ and $U_w \neq \emptyset$. Let w' be the longest proper prefix of w such that $W_{w'} = \emptyset$. Let ρ be any signalling run with $\mathcal{P}(\rho) = w$. Assume that $\rho_{\downarrow|w'|}$ is faulty. Thus the states visited by $\rho_{\downarrow n}$ for $|w'| < n \leq |w|$ were always in $W_{\rho_{\downarrow n}}$. Since $W_w = \emptyset$, this is not possible and so $\rho_{\downarrow|w'|}$ is correct. Thus every time a state with $W = \emptyset$ is visited, the length of the greatest prefix, for which all signalling subruns corresponding to this prefix are correct, is increased. This establishes correctness.

reactivity. Let ρ be an infinite run such that $D_{sup}(\mathcal{P}(\rho)) =?$. Due to the characterisation of Proposition 4.5, either (1) the strongly connected component of $\mathcal{A}_{\mathsf{IA}}$ that ρ infinitely often visits is not a BSCC or (2) ρ does not visit infinitely often all the states of this strongly connected component. The probability of such runs is null which establishes the reactivity.

size D has at most $2^{n_c}3^{n_f}$ memory states because every state of $\mathsf{IA}(\mathcal{A})$ consists of a triple (U,V,W) with $U \subseteq Q_c$ and $V \cup W \subseteq Q_f$. Moreover, one does not keep in V the states that are tracked in W, ensuring $V \cap W = \emptyset$. □

The following lower bound can be derived from the proof of Proposition 4.13, since the pLTS of Figure 4.18 is IA-diagnosable, and because any IA-diagnoser is also an FF-diagnoser.

Proposition 4.17. *There is a family* $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ *of* IA*-diagnosable pLTS such that* $\mathcal{A}_n$ *has* $2n+2$ *states and it admits no* IA*-diagnoser with less than* 2^n *memory states.*

The construction of an exact diagnoser can thus require exponential time, which is one class above the verification of exact diagnosability. This exponential time is only necessary if we want to build the full diagnoser though. Another possibility would be to update the state of the diagnoser on-the-fly during a run. One would only need to keep the current memory state (which has linear space), but the update process would take a polynomial time, instead of the constant time obtained when using a fully-built diagnoser.

4 Conclusion

In Section 1, we gave characterisations of the different notions of exact diagnosability and of one notion of approximate diagnosability for finite systems. The characterisations of the notions of exact diagnosability are of the same descriptive complexity, something that is impossible in the general case as shown in Section 3 of Chapter 3, page 77. The characterisation of the notion of approximate diagnosability has two important differences compared to the ones of exact diagnosability: (1) it depends on the exact probabilities of the system and (2) it only requires a comparison between pairs of states.

In Section 2, we gave matching upper and lower bounds for the various diagnosability problems for finite systems. These results heavily relied on the characterisations given in Section 1. Thus, as the characterisations of every notion of exact diagnosability have the same descriptive complexity, it is not surprising that they end up having the same complexity. The results on the approximate diagnosability notions are more surprising. AFF-diagnosability, the notion for which a characterisation was obtained, is decidable in polynomial time, a better complexity than what is needed for the exact diagnosability notions. This gain in complexity is obtained as, contrary to exact diagnosability where one needs to follow sets of states, for AFF-diagnosability, only pairs of states have to be compared. The comparison however depends on the exact values of the probabilities of the system, which brings a different kind of difficulty to the analysis. While this difficulty could be solved for AFF-diagnosability, it is the main reason of the undecidability of all the other notions of approximate diagnosability: (uniform) εFF- and uniform AFF-diagnosability. On this point, uniform AFF-diagnosability is equivalent to the notion of AA-diagnosability introduced in [TT05] which decidability was left open (only necessary conditions were given). Our undecidability result thus answers negatively to this question.

In Section 3, we gave automatic methods to construct finite-memory diagnosers for systems that are exactly diagnosable. We also showed that the sizes of the generated diagnosers are asymptotically optimal. For approximate notions of diagnosability, there does not necessarily exist finite-memory diagnosers and deciding the existence of such a diagnoser is undecidable. When such a diagnoser does not exist, one may need unbounded memory. Depending on the form of the unbounded memory required, it can be more or less manageable (for example a counter can easily be implemented).

Chapter 5

Algorithmic analysis of the diagnosability of infinite pLTS

In Chapter 4, we showed how to solve diagnosability for systems that can be represented by a pLTS with finitely many states.While this encompasses many kinds of systems, this is far from being exhaustive. Often, in order to satisfy its specification, a system will require unbounded memory: for example, when the system receives and records information or requests from the environment. Observe that an infinite number of states does not mean an infinite memory *per se*, but only an unbounded one. Stacks and queues are instances of such dynamic data structures.

While allowing an infinite amount of states increases the expressive power of the model, it increases the difficulty of its study. First, the studied systems must possess a finite representation. This can be done by assuming only a finite part of the system needs to be studied (*e.g.* infinite systems with finite attractors [BBS06]) or by using a higher level model whose semantics is an infinite-state system (Petri nets [Mur89, Dia09, CGS14], well structured transition systems [FS01], pushdown automata [AM04, KEM06, MP09, HS10, EY12]). We study such a formalism in this chapter.

Diagnosability has already been studied for infinite-state non-stochastic systems: represented by pushdown automata [MP09] or by Petri nets [CGLS12, BHSS18]. Partially observable visibly pushdown automata is a subclass of partially observable pushdown automata for which diagnosability was studied in [MP09]; for such models, diagnosability is decidable (using the determinisation procedure of [AM04]). With a restriction on the unobservable subnet akin to our convergence assumption, [CGLS12] gives a decidable characterisation of a non-stochastic notion of diagnosability for partially observable Petri nets. However the algorithm is non-primitive recursive. The authors of [BHSS18] extend this work by considering different classes of Petri nets and reducing the complexity (EXPSPACE for the general case). However to the best of our knowledge diagnosis of probabilistic infinite-state systems has not yet been studied.

So we extend these works by considering the probabilistic variants of diagnosability. In Section 1, we study the stochastic diagnosability of partially observable probabilistic pushdown automata (POpPDA). As diagnosability is already undecidable for

non-stochastic systems [MP09], the decidability of the stochastic variants is unlikely. However the notations introduced here will be used in Section 2 where we consider a restriction of POpPDA called partially observable probabilistic visibly pushdown automata (POpVPA). This restricted class has many advantages. It naturally benefits from the numerous results which are known for POpPDA [BEKK13], especially the ones on model-checking algorithms [KEM06, EY12]. Moreover, the authors of [AM04] gave an algorithm for the determinisation of a POpVPA. PopVPA generates an infinite-state pLTS and the efficient characterisations given in Chapter 4 strongly rely on the finiteness of the models. They thus cannot be used any more. So, we use the characterisations from the Section 3 of Chapter 3 based on the pathL logic. However the model-checking algorithm cannot be directly applied to the formulae of the pathL logic. Some tricky machinery will be required to "encode" the path formulae of the pathL logic in the system in order to use the results of [EY12]. Finally, in Section 3, we study the diagnosability of partially observable stochastic Petri nets (POSPN), we mimic the restriction used on POpPDA in Section 2, and discuss the case of partially observable stochastic visible Petri nets (VSPN).

This chapter develops and extends some of the results from [BHL16b, LGS18].

1 Diagnosability of probabilistic pushdown automata

In this section we study infinite-state pLTS generated by probabilistic pushdown automata (pPDA). First, we define pPDA and the infinite-state pLTS generated by a pPDA. The pPDA model being very expressive, we show that diagnosability of pPDA is undecidable.

1.1 Probabilistic pushdown automata

A pPDA randomly generates infinite behaviours using a stack. This stack contains letters of a stack alphabet with the most recently added letter put at the top. Transitions of the pPDA can be conditioned by the top of the stack. Moreover, a transition can push a new element onto the stack, pop one element off it or modify the top of the stack. Let us first see an example of an infinite state pLTS, that we will be able to represent by a pPDA.

Example 5.1. *The pLTS of Figure 5.1 represents a server that accepts jobs (event* in*) until it randomly decides to serve the jobs (event* serve*). When a job is done the result is delivered (event* out*). When all jobs are done, the server waits for a new batch of jobs. However randomly, the server may trigger a fault (event* **f***) and then abort all remaining jobs (event* abort*). Afterwards, the server is reset (event* reset*).*

The infinite number of states in this system comes from the unbounded number of jobs the server can receive before he starts serving them. To see this system using a stack, one could use a stack letter to represent a job and add one such letter to the stack every time a new job comes in. Dealing with a job (event out*) or aborting it (event* abort*) removes an element of the stack.*

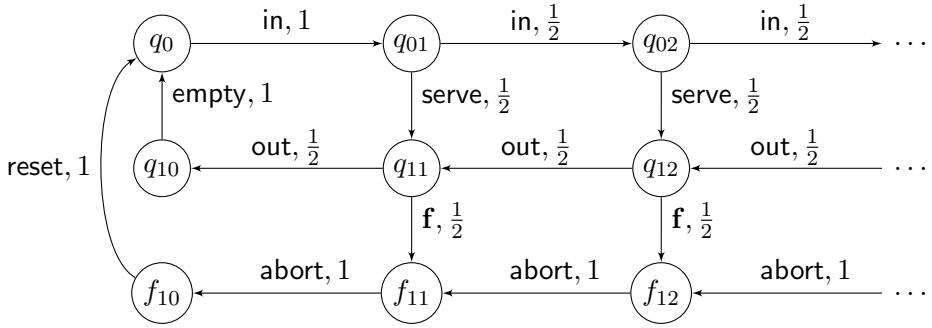


Figure 5.1: An infinite-state pLTS that can be represented by a pPDA.

We now define pPDA similarly to what can be found in [KEM06].

Definition 5.1. *A* probabilistic pushdown automaton *(pPDA) is defined by a tuple* $\mathcal{V} = (Q, q_0, \Sigma, \Gamma, \delta, \mathbf{P})$ *where:*

- Q *is a finite set of states with* q_0 *the initial state;*
- Σ *is a finite alphabet of events;*
- Γ *is a finite alphabet of stack symbols including a set of bottom stack symbols* $\Gamma_\perp$ *with initial symbol* $\perp_0 \in \Gamma_\perp$*;*
- $\delta \subseteq Q \times \Gamma \times \Sigma \times Q \times \Gamma^*$ *is the set of transitions such that for every* $(q, \gamma, a, q, w) \in \delta$*,* $|w| \leq 2$*,* $\gamma \in \Gamma_\perp$ *implies* $w \in \Gamma_\perp(\Gamma \setminus \Gamma_\perp)^*$ *and* $\gamma \notin \Gamma_\perp$ *implies* $w \in (\Gamma \setminus \Gamma_\perp)^*$*;*
- $\mathbf{P}$ *is the transition probability function fulfilling for every* $q \in Q$ *and* $\gamma \in \Gamma$*:*

$$\sum_{(q,\gamma,a,q',w)\in\delta} \mathbf{P}[(q, \gamma, a, q', w)] = 1.$$

A pPDA may be viewed as a pLTS equipped with a stack. The transitions of the pPDA can depend on the top symbol of the stack and modify it. The definition ensures that the stack is never empty: the bottom stack symbols $\Gamma_\perp$ are never removed. Moreover symbols of $\Gamma_\perp$ never occurs elsewhere in the stack. Let $T = (q, \gamma, a, q', w) \in \delta$ be a transition of a pPDA. If $|w| = 1$ (resp. $|w| = 2$, $|w| = 0$) then T is said to be a *local* (resp. *push*, *pop*) transition. A local transition can update the top symbol and a push transition can modify the top symbol and add another symbol on top of it. Notions such as runs are defined on pPDA analogously to what was done for pLTS. We call *configuration* the pair composed by a state and a stack.

The semantics of a pPDA is the (potentially infinite) pLTS representing its behaviour. The states of this pLTS are pairs consisting of a state and a stack contents. They therefore contains all the information that are necessary in the pPDA to determine the available transitions and their probabilities.

Definition 5.2. *A probabilistic pushdown automaton pPDA* $\mathcal{V} = (Q, q_0, \Sigma, \Gamma, \delta, \mathbf{P})$ *defines a pLTS* $\mathcal{A}_\mathcal{V} = (Q_\mathcal{V}, (q_0, \bot), \Sigma, T_\mathcal{V}, \mathbf{P}_\mathcal{V})$ *where:*

- $Q_\mathcal{V} = \{(q, z) \mid q \in Q \wedge z \in \bot\Gamma^*\}$;
- $T_\mathcal{V} = \{((q, z\gamma), a, (q', zw)) \mid z\gamma \in \bot\Gamma^* \wedge (q, \gamma, a, q', w) \in \delta\}$;
- *For every* $((q, z\gamma), a, (q', zw)) \in T_\mathcal{V}$, $\mathbf{P}_\mathcal{V}[((q, z\gamma), a, (q', zw))] = \mathbf{P}[(q, \gamma, a, q', w)]$.

As a pPDA has a finite number of states, the associated pLTS is finitely branching.

Example 5.2. *Figure 5.2 gives an example of a pPDA whose semantics is the pLTS from Figure 5.1. Indeed, the stack alphabet has only one letter. We could thus replace it by a counter giving the number of element in the stack. The pLTS of Figure 5.1 does exactly that by representing the configuration* (q_1, γ^n) *of the semantics of the pPDA by the state* q_{1n} *and similarly for the other configurations. A transition* $t = (q, \gamma, a, q', w)$ *is represented by an edge from state* q *to state* q' *and labelled by* $\mathbf{P}[t] \cdot \gamma, a, w$.

Observe that in this example the set of states is not partitioned between faulty and correct states as from the state f_1 *reached by a faulty run, one can go back to the initial state with the reset event (event* r*).*

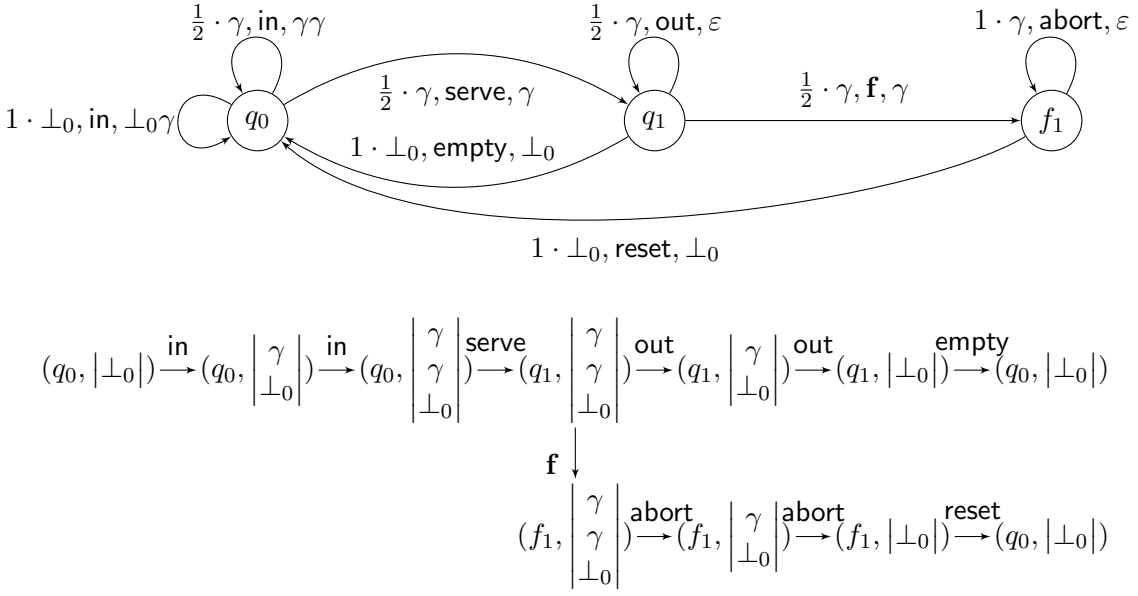


Figure 5.2: A pPDA generating the pLTS from Figure 5.1 and two finite of its runs.

As for pLTS, we enlarge pPDA with partial observation features. As discussed in Section 1.3 of Chapter 2, this can be done either by partitioning the alphabet of events into observable and unobservable events or by providing a mask function associating

with every event an observation. In the previous chapters we used the partition of the alphabet of events. Here, we use the mask function which is more appropriate for our needs.

Definition 5.3. *A* partially observable pPDA *(POpPDA) is a tuple* $\langle \mathcal{V}, \Sigma_o, \mathcal{P} \rangle$ *consisting of a pPDA* $\mathcal{V}$ *equipped with a mapping* $\mathcal{P} : \Sigma \to \Sigma_o \cup \{\varepsilon\}$ *where* Σ_o *is the set of observations.*

A POpPDA is diagnosable according to a notion of diagnosability if the pLTS it generates is diagnosable. As the generated pLTS are finitely branching, thanks to Theorem 3.1 a POpPDA is FF-diagnosable iff it is IF-diagnosable.

Example 5.3. *Consider the pPDA* $\mathcal{V}$ *of Figure 5.2, we define* $\Sigma_o = \{\text{in}, \text{out}, \text{loc}, \text{reset}\}$ *and two observation masks* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *with* $\mathcal{P}_1(\mathsf{in}) = \text{in}$, $\mathcal{P}_1(\mathsf{serve}) = \mathcal{P}_1(\mathsf{empty}) = \mathcal{P}_1(\mathsf{reset}) = \text{loc}$, $\mathcal{P}_1(\mathsf{abort}) = \mathcal{P}_1(\mathsf{out}) = \text{out}$ *and* $\mathcal{P}_1(\mathbf{f}) = \varepsilon$, $\mathcal{P}_2(\mathsf{reset}) = \text{reset}$ *and for every event* $t \neq \mathsf{reset}$, $\mathcal{P}_2(t) = \mathcal{P}_1(t)$. $\langle \mathcal{V}, \Sigma_o, \mathcal{P}_1 \rangle$ *and* $\langle \mathcal{V}, \Sigma_o, \mathcal{P}_2 \rangle$ *are two POpPDA differentiated only by the observation of the event* reset. *As a faulty run will inevitably contain a* reset *and a correct run that leaves* q_0 *will contain a* serve*, the POpPDA* $\langle \mathcal{V}, \Sigma_o, \mathcal{P}_2 \rangle$ *which distinguishes these two events is diagnosable for every non-uniform exact notion of diagnosability. However, the POpPDA* $\langle \mathcal{V}, \Sigma_o, \mathcal{P}_1 \rangle$ *is not diagnosable as serving the requests and going back to* q_0 *has the same observation as making a fault, aborting the requests and going back to* q_0.

1.2 Undecidability of diagnosability for POpPDA

Unfortunately, for every notion of diagnosability, the diagnosability problem for POpPDA is undecidable. The undecidability can be obtained by adapting the proof for diagnosability of *non-probabilistic* pushdown automata [MP09]. However, in order to show how robust the result is, we rather reduce from the Post Correspondence Problem (PCP). An instance of PCP is given by an integer $n \in \mathbb{N}$ and two families of non-empty words $\{v_i\}_{i \leq n}$ and $\{w_i\}_{i \leq n}$ on the alphabet $\{a, b\}$. The following question is undecidable [Pos46]: does there exist $k > 0$ and $i_1, \dots i_k \in \{1, \dots, n\}$ such that $w_{i_1} \dots w_{i_k} = v_{i_1} \dots v_{i_k}$?

We show in Theorems 5.1 and 5.2 that undecidability already holds for two (incomparable) subclasses of POpPDA with restriction on what is observable and on the number of phases of any run. A *phase* is a portion of a run in which the stack either never decreases or never increases.

Theorem 5.1. *The diagnosability problems are undecidable for POpPDA even when (1) a local transition does not update the top of the stack, (2) every event labelling a push transition is fully observable and corresponds to the pushed symbol, and (3) every run has at most two phases.*

The use of the stack is obviously central to the proof of the undecidability. Moreover conditions (1) and (2) limit the ability of push and local transitions to silently manipulate the stack. There is no such limitation for pop transitions however as the proof of undecidability heavily rely on hiding when the pop transitions are performed.

Let us sketch the proof. We reduce the PCP problem to the diagnosability problem. To do so, the POpPDA we build, from a PCP instance $(n, \{v_i\}_{i\leq n}, \{w_i\}_{i\leq n})$, is divided in three parts. In the first one, it will select and push onto its stack a sequence of numbers, $i_1 \dots i_k$, with $\forall j \leq k, i_j \leq n$. Then it goes randomly to one of the two other parts of the POpPDA, one part being accessed by a faulty transition. Each of these two parts is associated with a family of words of the PCP instance. Once it reached one of these parts, say the one associated with the w_i, the run will read the words of w_i induced by the numbers pushed on the stack. The resulting observation is $w = w_{i_1} \dots w_{i_k}$. On the other part, the observation is similarly $v = v_{i_1} \dots v_{i_k}$. The fault can be detected if and only if $v \neq w$. Having the pop transition undetected is fundamental as it allows to hide when the word w_{i_j} starts being read.

Proof. Let $(n, \{v_i\}_{i\leq n}, \{w_i\}_{i\leq n})$ be a PCP instance In this proof, we let ℓ_i (resp. m_i) be the length of v_i (resp. w_i). Also, given a word w and $k \leq |w|$ we use $w[k]$ to denote the k^{th}-letter of w.

We build a pPDA $\mathcal{V} = (Q, \Sigma, \Gamma, \delta, \mathbf{P})$ as follows:

- $Q = \{q_0, q_c, q_s, f_s\} \cup \{q_i^k \mid 1 \leq i \leq n, 1 \leq k \leq \ell_i\} \cup \{f_i^k \mid 1 \leq i \leq n, 1 \leq k \leq m_i\}$;
- $\Sigma = \{1, \dots, n, \natural, u, r, \mathbf{f}, a, b\}$;
- $\Gamma = \{1, \dots, n, \bot_0\}$ with $\Gamma_\bot = \{\bot_0\}$;
- δ consists of the following transitions:

$$\begin{aligned}
&\{(q_0, \bot_0, x, \bot_0 x, q_c) \mid 1 \leq x \leq n\} \cup \{(q_c, x, y, xy, q_c) \mid 1 \leq x, y \leq n\} \\
&\cup \{(q_i^k, z, v_i[k], z, q_i^{k+1}) \mid 1 \leq i \leq n, 1 \leq k < \ell_i, z \in \{\bot_0, 1, \dots, n\}\} \\
&\cup \{(f_i^k, z, w_i[k], z, f_i^{k+1}) \mid 1 \leq i \leq n, 1 \leq k < m_i, z \in \{\bot_0, 1, \dots, n\}\} \\
&\cup \{(q_i^{\ell_i}, z, v_i[\ell_i], z, q_s) \mid 1 \leq i \leq n, z \in \{\bot_0, 1, \dots, n\}\} \cup \{(q_s, x, r, \varepsilon, q_x^1) \mid 1 \leq x \leq n\} \\
&\cup \{(f_i^{m_i}, z, w_i[m_i], z, f_s) \mid 1 \leq i \leq n, z \in \{\bot_0, 1, \dots, n\}\} \cup \{(f_s, x, r, \varepsilon, f_x^1) \mid 1 \leq x \leq n\} \\
&\cup \{(q_c, x, u, x, q_s), (q_c, x, \mathbf{f}, x, f_s) \mid 1 \leq x \leq n\} \cup \{(q_s, \bot_0, \natural, \bot_0, q_s), (f_s, \bot_0, \natural, \bot_0, f_s)\}.
\end{aligned}$$

- $\mathbf{P}$ assigns arbitrary positive probabilities to transitions in δ:

$$\mathbf{P}(q, \gamma, a, q', w) > 0 \Leftrightarrow (q, \gamma, a, q', w) \in \delta \text{ and } \sum_{(q,\gamma,a,q',w)\in\delta} \mathbf{P}[(q, \gamma, a, q', w)] = 1.$$

We further consider the POpPDA $\langle \mathcal{V}, \Sigma_o, \mathcal{P} \rangle$ with $\Sigma_o = \Sigma \setminus \{r, u, \mathbf{f}\}$, and the masking function satisfies $\mathcal{P}(u) = \mathcal{P}(r) = \mathcal{P}(\mathbf{f}) = \varepsilon$ and $\mathcal{P}(x) = x$ for any other event x. This POpPDA is represented in Figure 5.3.

Let us prove that the instance of the PCP is positive if and only if the POpPDA is IF-, IA-, FA- and AFF-diagnosable.

First, observe that $\natural$ almost surely occurs in an infinite run of the pPDA $\mathcal{V}$. Thus, for any $\varepsilon > 0$, there exists $N_\varepsilon \in \mathbb{N}$ such that the measure of signalling runs with observable length N_ε that reach configurations $(q_s, \bot_0)$ or $(f_s, \bot_0)$ by an event $\natural$ is at least $1 - \varepsilon$.

• Assume that there exists a solution $i_1, \ldots, i_k$ to the PCP instance $(n, \{v_i\}_{i\leq n}, \{w_i\}_{i\leq n})$. Consider in the POpPDA the faulty run:

$$\rho_f = q_0(i_j q_c)_{j\leq k}\mathbf{f}(f_s r(f_{i_j}^p w_{i_j}[p])_{p\leq m_{i_j}})_{j\leq k}(f_s \natural)^\omega \ ,$$

and the correct run:

$$\rho_c = q_0(i_j q_c)_{j\leq k} u(q_s r(q_{i_j}^p v_{i_j}[p])_{p\leq \ell_{i_j}})_{j\leq k}(q_s \natural)^\omega \ .$$

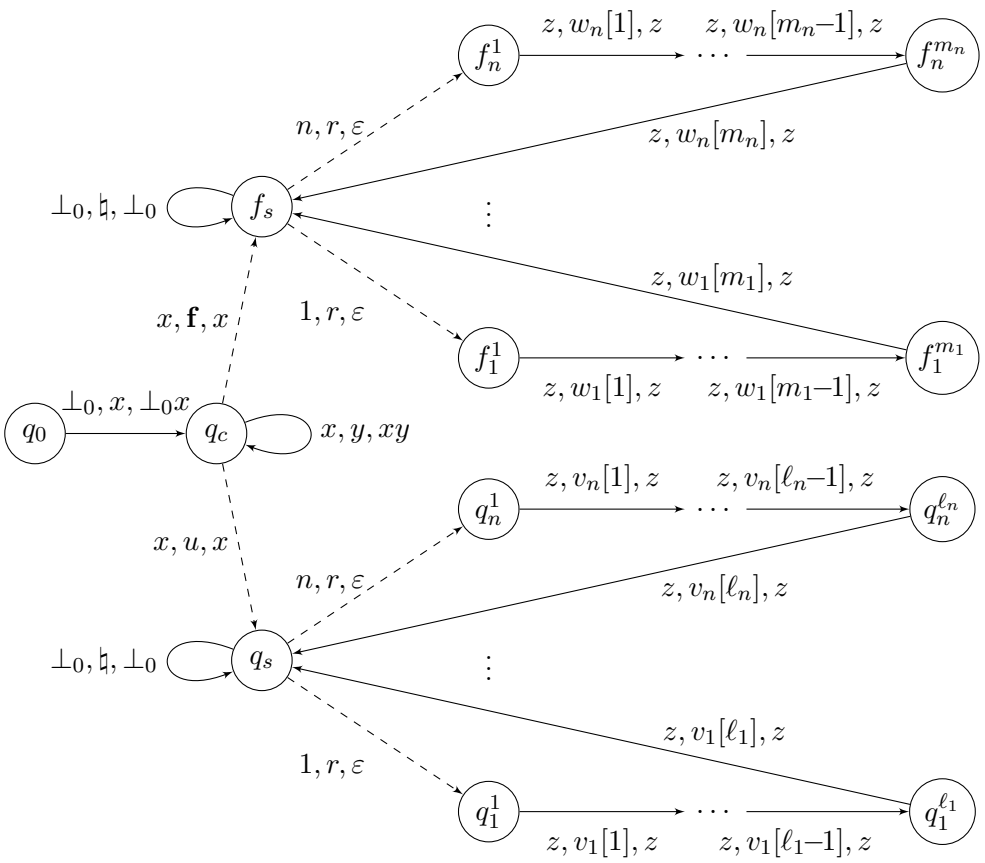


Figure 5.3: A POpPDA for the proof of Theorem 5.1.

These two runs have the same observed sequence: $\mathcal{P}(\rho_f) = \mathcal{P}(\rho_c) = i_1 \ldots i_k w \natural^\omega$ with $w = w_{i_1} \ldots w_{i_k} = v_{i_1} \ldots v_{i_k}$. Therefore, ρ_f is an infinite ambiguous faulty run. Given that $\mathbb{P}(\rho_f) > 0$, we deduce that the POpPDA $\langle \mathcal{V}, \Sigma_o, \mathcal{P} \rangle$ is not IF-diagnosable. From Theorem 3.1, it is also neither IA-diagnosable nor FA-diagnosable. Moreover, after the occurrence of a fault, there is no probabilistic choice. As a consequence the correctness proportion is either 0 or 1/2. As the correctness proportion of the faulty prefixes of ρ_f is never 0 as seen above, the POpPDA is not AFF-diagnosable.

• Conversely, assume that the PCP instance $(n, \{v_i\}_{i\leq n}, \{w_i\}_{i\leq n})$ has no solution. Let $\varepsilon > 0$, let $N_\varepsilon \in \mathbb{N}$ be the integer obtained with our earlier observation. Consider a correct run ρ_c with observable length N_ε, ending in $(q_s, \bot_0)$ and containing at least an occurrence of $\natural$. Its observed sequence is of the form $\mathcal{P}(\rho_c) = i_1 \dots i_k v_{i_1} \dots v_{i_k} \natural^m$ for some $i_1, \dots, i_k, m$. Due to the fact that $(n, \{v_i\}_{i\leq n}, \{w_i\}_{i\leq n})$ has no solution, no faulty run can have the same observed sequence. Therefore, ρ_c is surely correct. Symmetrically, any faulty run ending in $(f_s, \bot_0)$ after an occurrence of $\natural$ is surely faulty. We thus conclude that, for any $\varepsilon > 0$, there exists $N_\varepsilon \in \mathbb{N}$ such that $\mathbb{P}(\mathsf{FAmb}_{N_\varepsilon} \uplus \mathsf{CAmb}_{N_\varepsilon}) \leq \varepsilon$. As a consequence, the POpPDA $\langle \mathcal{V}, \Sigma_o, \mathcal{P}\rangle$ is FA-diagnosable. By Theorem 3.1 it is also IA-diagnosable, IF-diagnosable and AFF-diagnosable. □

A similar undecidability result holds for a class of POpPDA in which pop events are fully observable, and the number of phases is constant:

Theorem 5.2. *The diagnosability problems are undecidable for POpPDA even when (1) a local transition does not update the top of the stack, (2) every event labelling a pop transition is fully observable and corresponds to the popped symbol, and (3) every run has at most two phases.*

Proof. The proof follows the same line as the one for Theorem 5.1. The difference is that instead of choosing first the words that will be read by pushing them on the stack and later popping them off discreetly, the pPDA reads the words and silently push on the stack which words were read and at the end pop them off and verify if the same sequence could indeed be used for both family of words.
From an instance $(n, \{v_i\}_{i\leq n}, \{w_i\}_{i\leq n})$ of PCP, let us define a pPDA $\mathcal{V} = (Q, \Sigma, \Gamma, \delta, \mathbf{P})$ where:

- $Q = \{q_0, q_s, f_s, q_e, f_e\} \cup \{q_i^k \mid 1 \leq i \leq n, 1 \leq k \leq \ell_i\} \cup \{f_i^k \mid 1 \leq i \leq n, 1 \leq k \leq m_i\}$;
- $\Sigma = \{1, \dots, n, \natural, u, c, \mathbf{f}, a, b\}$;
- $\Gamma = \{1, \dots, n, \bot_0\}$ with $\Gamma_\bot = \{\bot_0\}$;
- δ consists of the following transitions:

$$
\begin{aligned}
&\{(q_0, \bot, u, \bot, q_s), (q_0, \bot, \mathbf{f}, \bot, f_s), (q_e, \bot, \natural, \bot, q_e), (f_e, \bot, \natural, \bot, f_e)\}\\
&\cup\ \{(q_i^k, z, v_i[k], z, q_i^{k+1}) \mid 1 \leq i \leq n, 1 \leq k < \ell_i, z \in \{\bot, 1, \dots, n\}\}\\
&\cup\ \{(f_i^k, z, w_i[k], z, f_i^{k+1}) \mid 1 \leq i \leq n, 1 \leq k < m_i, z \in \{\bot, 1, \dots, n\}\}\\
&\cup\ \{(q_i^{\ell_i}, z, v_i[\ell_i], z, q_s) \mid 1 \leq i \leq n, z \in \{\bot, 1, \dots, n\}\}\\
&\cup\ \{(f_i^{m_i}, z, w_i[m_i], z, f_s) \mid 1 \leq i \leq n, z \in \{\bot, 1, \dots, n\}\\
&\cup\ \{(q_s, z, c, zx, q_x^1) \mid z \in \{\bot, 1, \dots, n\}, x \in \{1, \dots, n\}\}\\
&\cup\ \{(f_s, z, c, zx, f_x^1) \mid z \in \{\bot, 1, \dots, n\}, x \in \{1, \dots, n\}\}\\
&\cup\ \{(q_s, x, x, \varepsilon, q_e) \mid x \in \{1, \dots, n\}\} \cup\ \{(f_s, x, x, \varepsilon, f_e) \mid x \in \{1, \dots, n\}\}\\
&\cup\ \{(q_e, x, x, \varepsilon, q_e) \mid x \in \{1, \dots, n\}\} \cup\ \{(f_e, x, x, \varepsilon, f_e) \mid x \in \{1, \dots, n\}\}.
\end{aligned}
$$

- **P** assigns arbitrary positive probabilities to transitions in δ.

We further consider the POpPDA $\langle \mathcal{V}, \Sigma_o, \mathcal{P} \rangle$ with $\Sigma_o = \Sigma \setminus \{c, u, \mathbf{f}\}$, and the masking function satisfies $\mathcal{P}(u) = \mathcal{P}(c) = \mathcal{P}(\mathbf{f}) = \varepsilon$ and $\mathcal{P}(x) = x$ for any other event x. This POpPDA is represented in Figure 5.4.

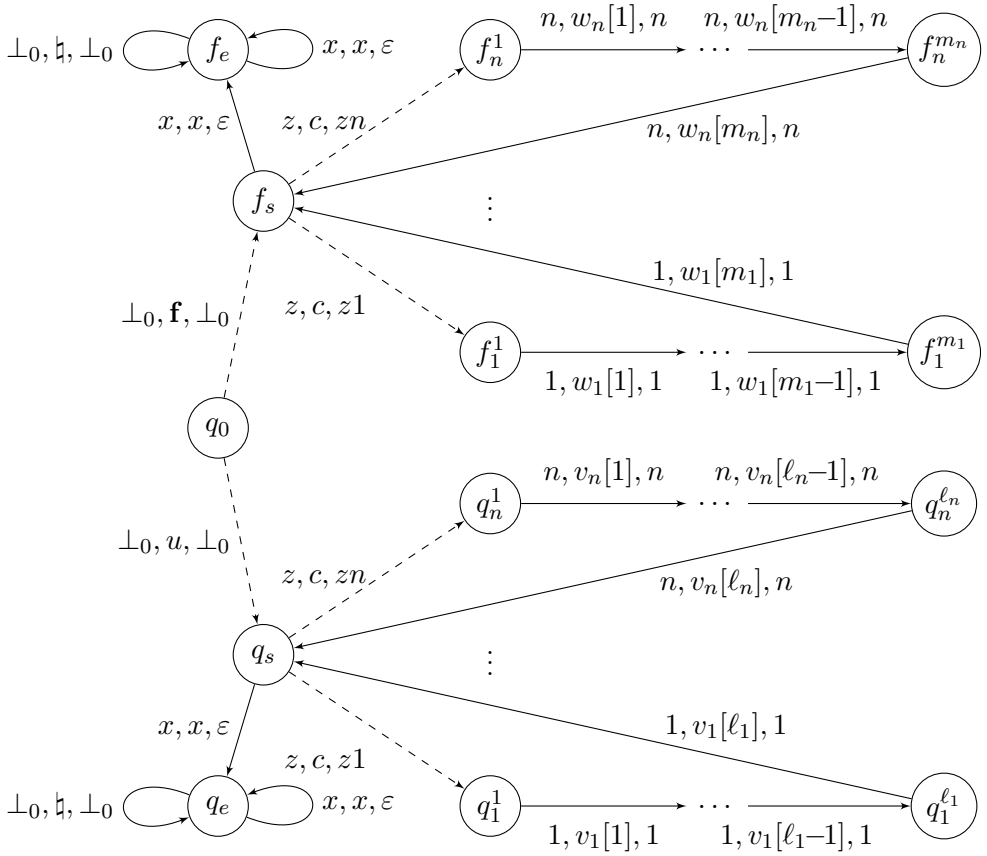


Figure 5.4: A POpPDA for the proof of Theorem 5.2.

Let us prove that the instance of the PCP is positive if and only if the POpPDA is IF-, IA-, FA- and AFF-diagnosable.

First, observe that $\natural$ almost surely occurs in an infinite run of the pPDA $\mathcal{V}$. Thus, for any $\varepsilon > 0$, there exists $N_\varepsilon \in \mathbb{N}$ such that the measure of signalling runs with observable length N_ε that reach configurations $(q_e, \bot_0)$ or $(f_e, \bot_0)$ by an event $\natural$ is at least $1 - \varepsilon$.
• Assume that there exists a solution $i_1, \ldots, i_k$ to the PCP instance $(n, \{v_i\}_{i \le n}, \{w_i\}_{i \le n})$. Consider the faulty run:

$$\rho_f = q_0 \mathbf{f} f_s (c(f_{i_j}^p w_{i_j}[p])_{p \le m_{i_j}} f_s)_{j \le k} (i_j f_e)_{j \le k} (\natural f_e)^\omega \ ,$$

and the correct run:

$$\rho_c = q_0 u q_s (c(q_{i_j}^p v_{i_j}[p])_{p \le \ell_{i_j}} q_s)_{j \le k} (i_j q_e)_{j \le k} (\natural q_e)^\omega \ .$$

These two runs have the same observed sequence: $\mathcal{P}(\rho_f) = \mathcal{P}(\rho_c) = wi_1 \dots i_k \natural^\omega$ with $w = w_{i_1} \dots w_{i_k} = v_{i_1} \dots v_{i_k}$. Therefore, ρ_f is an infinite ambiguous faulty run. Given that $\mathbb{P}(\rho_f) > 0$, we deduce that the POpPDA $\langle \mathcal{A}, \Sigma_o, \mathcal{P} \rangle$ is not IF-diagnosable. From Theorem 3.1, it is also neither IA-diagnosable nor FA-diagnosable. Moreover, after reaching the state f_e or q_e, there is no probabilistic choice. As a consequence the sequence of the correctness proportion of the faulty prefixes of ρ_f is stationary. As it is never 0 as seen above, the POpPDA is not AFF-diagnosable.

• Conversely, assume that the PCP instance $(n, \{v_i\}_{i \leq n}, \{w_i\}_{i \leq n})$ has no solution. Let $\varepsilon > 0$, let $N_\varepsilon \in \mathbb{N}$ be the integer obtained with our earlier observation. Consider a correct run ρ_c with observable length N_ε ending in $(q_e, \bot_0)$ and with an occurrence of $\natural$. Its observed sequence is of the form $v_{i_1} \dots v_{i_k} i_1 \dots i_k \natural^m$ for some $i_1, \dots, i_k, m$. Due to the fact that $(n, \{v_i\}_{i \leq n}, \{w_i\}_{i \leq n})$ has no solution, no faulty run can have the same observed sequence. Therefore, ρ_c is surely correct. Symmetrically, any faulty run ending in $(f_e, \bot_0)$ by an occurrence of $\natural$ is surely faulty. We thus conclude that, for any $\varepsilon > 0$, there exists $N_\varepsilon \in \mathbb{N}$ such that $\mathbb{P}(\mathsf{FAmb}_{N_\varepsilon} \uplus \mathsf{CAmb}_{N_\varepsilon}) \leq \varepsilon$. As a consequence, the POpPDA $\langle \mathcal{V}, \Sigma_o, \mathcal{P} \rangle$ is FA-diagnosable. By Theorem 3.1 it is also IA-diagnosable, IF-diagnosable and AFF-diagnosable. □

2 Diagnosability of probabilistic visibly pushdown automata

As diagnosability is undecidable for pPDA, we now turn to a more restrictive model: probabilistic visibly pushdown automata (pVPA) [AM04]. While keeping a significant expressive power, pVPA is a natural subclass of pushdown automata that is more tractable than the general model and which language has many of the desirable properties that regular languages have. In particular, there exists a method for the determinisation of a non-deterministic visibly pushdown automaton [AM04].

We formally define pVPA and describe how to build a diagnosis-oriented determinisation of a pVPA. Then, we give a decision procedure for diagnosability and study the hardness of the diagnosability problems.

2.1 Probabilistic visibly pushdown automata and diagnosis-oriented determinisation

A pVPA is a pPDA where events are partitioned into three sets depending on if they correspond to push, pop, or local transitions.

Definition 5.4. *A* probabilistic visibly pushdown automaton *(pVPA) is a pPDA* $\mathcal{V} = (Q, q_0, \Sigma, \Gamma, \delta, \mathbf{P})$ *whose event alphabet is partitioned into local, push and pop events* $\Sigma = \Sigma_\natural \uplus \Sigma_\sharp \uplus \Sigma_\flat$ *and such that for every transition* $T = (q, \gamma, a, q', w) \in \delta$, *T is a local (resp. push, pop) transition iff a is a local (resp. push, pop) event.*

A pVPA without the transition probability function $\mathbf{P}$ *is called a visibly pushdown automata (VPA).*

The definitions of pPDA carry on to pVPA in particular the semantics of a pVPA is an infinite-state finitely-branching pLTS.

Example 5.4. *Consider the pPDA of Figure 5.2. This pPDA is a pVPA as shown by the partition of events given by* $\Sigma_\sharp = \{in\}, \Sigma_\flat = \{out, abort\}$ *and* $\Sigma_\natural = \{serve, empty, reset, \mathbf{f}\}$.

To define partially observable pVPA, we equip a pVPA with a mask function and require that only local events may be unobservable, and that pushes and pops can still be distinguished. As a consequence, given the observed sequence of one run, one can deduce the height of the stack as the difference between pushes and pops, plus one (the bottom symbol).

Definition 5.5. *A* partially observable pVPA *(POpVPA) is a tuple* $\langle \mathcal{V}, \Sigma_o, \mathcal{P} \rangle$ *consisting of a pVPA* $\mathcal{V}$ *equipped with a mapping* $\mathcal{P} : \Sigma \to \Sigma_o \cup \{\varepsilon\}$ *such that:*

- $\Sigma_o = \Sigma_{o,\natural} \uplus \Sigma_{o,\sharp} \uplus \Sigma_{o,\flat}$ *is the set of observations;*
- $\mathcal{P}(\Sigma_\natural) \subseteq \Sigma_{o,\natural} \cup \{\varepsilon\}$, $\mathcal{P}(\Sigma_\sharp) \subseteq \Sigma_{o,\sharp}$ *and* $\mathcal{P}(\Sigma_\flat) \subseteq \Sigma_{o,\flat}$.

When we aimed to verify the diagnosability of finite pLTS in Chapter 4, one of the first step was to build a diagnosis-oriented determinisation of the pLTS (see Definition 4.1, page 91). While this could not be done for pPDA, a determinisation for pVPA was established by Alur and Madhusudan [AM04]. Following the same approach, we now explain how to adapt the determinisation of [AM04] for diagnosability. For a pVPA $\mathcal{V}$, its determinisation is called the estimate VPA of $\mathcal{V}$ and is denoted $\mathcal{A}(\mathcal{V})$. As in the finite case, we need tags that reflect the category of runs (faulty or correct) given an observed sequence with a distinction between "old" and "young" faulty runs. Due to its technicality, we postpone the formal definition of $\mathcal{A}(\mathcal{V})$: we first explain some features of the construction and illustrate them on an example (represented in Figure 5.5).

States and stack symbols. The VPA $\mathcal{A}(\mathcal{V})$ tracks all runs with same observation in parallel memorising their status w.r.t. faults. More precisely to the current set of runs corresponds the symbol on the top of the stack which is a set of tuples where each tuple is written as a fraction $\frac{\gamma,\mathsf{X},q}{\gamma^-,\mathsf{X}^-,q^-}$. Let us describe the meaning of this tuple:

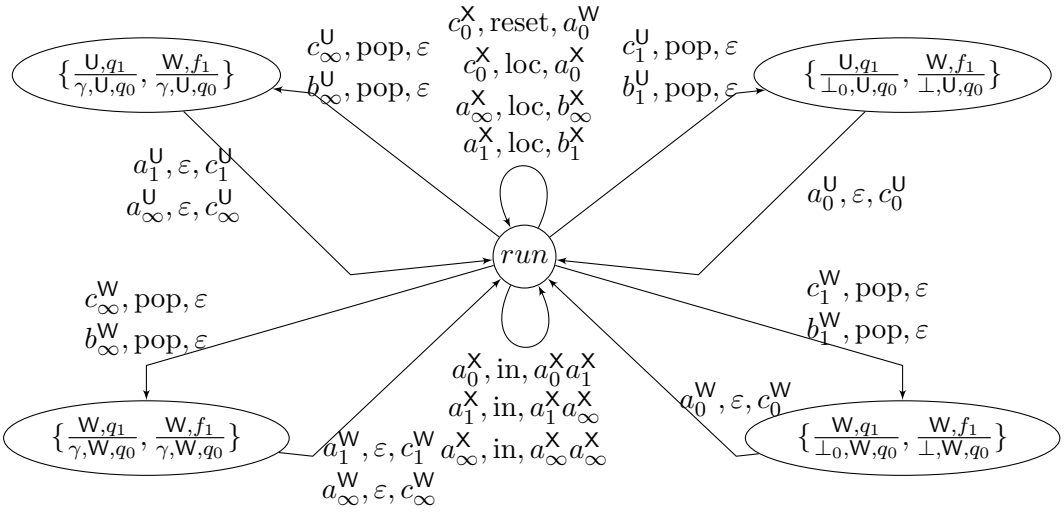
- q is the current state of the run and γ is the symbol on the top of its stack;
- $\mathsf{X} \in \mathsf{Tg} = \{\mathsf{U}, \mathsf{V}, \mathsf{W}\}$ is the status of the run: U for a correct run, V for a young faulty run and W for an old faulty run;
- The denominator $(\gamma^-, \mathsf{X}^-, q^-)$, is related to the configuration just after the last push event of the run: γ^- is the stack symbol under the top symbol, while X^- is the status of the run reaching this configuration and q^- the state of this configuration.

A priori, a single state *run* would be enough. However the simulation of a pop event in the original VPA is performed in two steps requiring some additional states that we explain later.

Example 5.5. *The initial configuration of the VPA* $\mathcal{A}(\mathcal{V})$ *of Figure 5.5* $(run, \left|\{\frac{\bot_0,\mathsf{U},q_0}{\bot_0,\mathsf{U},q_0}\}\right|)$ *corresponds to the empty run represented by a singleton. The denominator of bottom stack symbols is by convention* $(\bot_0, \mathsf{U}, q_0)$ *and is irrelevant for specifying the transitions of* $\mathcal{A}(\mathcal{V})$.

$$a_0^{\mathsf{X}} = \{\tfrac{\bot,\mathsf{X},q_0}{\bot,\mathsf{X},q_0}\}, a_1^{\mathsf{X}} = \{\tfrac{\gamma,\mathsf{X},q_0}{\bot,\mathsf{X},q_0}\}, a_\infty^{\mathsf{X}} = \{\tfrac{\gamma,\mathsf{X},q_0}{\gamma,\mathsf{X},q_0}\}, b_1^{\mathsf{X}} = \{\tfrac{\gamma,\mathsf{X},q_1}{\bot,\mathsf{X},q_0}\}, b_\infty^{\mathsf{X}} = \{\tfrac{\gamma,\mathsf{X},q_1}{\gamma,\mathsf{X},q_0}\}$$

$$c_0^{\mathsf{X}} = \{\tfrac{\bot,\mathsf{X},q_1}{\bot,\mathsf{U},q_0}, \tfrac{\bot,\mathsf{X},f_1}{\bot,\mathsf{U},q_0}\}, c_1^{\mathsf{X}} = \{\tfrac{\gamma,\mathsf{X},q_1}{\bot,\mathsf{X},q_0}, \tfrac{\gamma,\mathsf{X},f_1}{\bot,\mathsf{X},q_0}\}, c_\infty^{\mathsf{X}} = \{\tfrac{\gamma,\mathsf{X},q_1}{\gamma,\mathsf{U},q_0}, \tfrac{\gamma,\mathsf{X},f_1}{\gamma,\mathsf{U},q_0}\}, \qquad \mathsf{X} \in \{\mathsf{U},\mathsf{W}\}$$



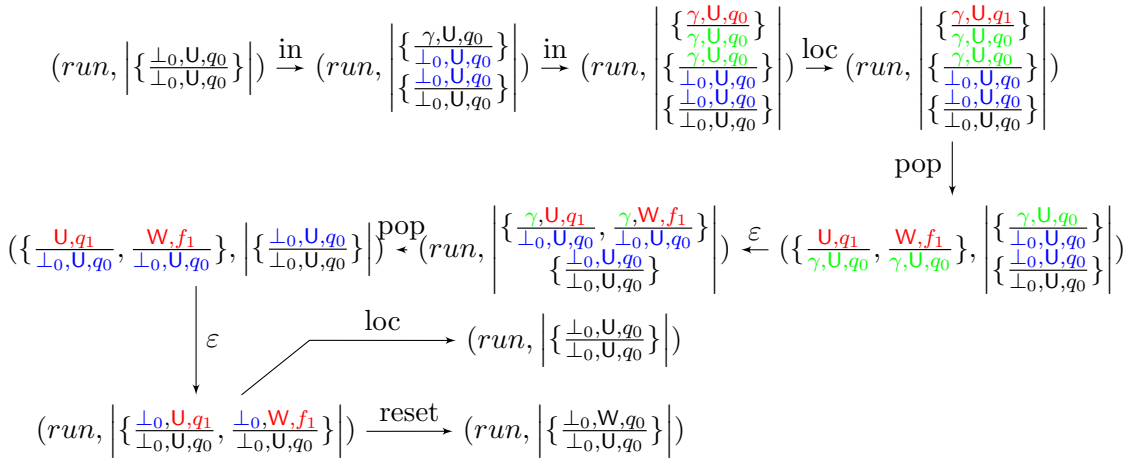


Figure 5.5: The VPA $\mathcal{A}(\mathcal{V})$ associated with the POpVPA $\langle \mathcal{V}, \Sigma_o, \mathcal{P}_2 \rangle$ of Example 5.3 with two runs. The tag V was ignored to remove redundancy and simplify the figure.

Tag updates. Let us explain how the tag X of an item $\frac{\gamma,\mathsf{X},q}{\gamma^-,\mathsf{X}^-,q^-}$ of the current stack symbol is determined. If this item corresponds to a correct run then $\mathsf{X} = \mathsf{U}$. When after a transition of $\mathcal{A}(\mathcal{V})$ a (tracked) correct run becomes faulty, there are two cases. Either there was no tag W in (the numerators of items of) the top stack symbol of the stack then the run is tagged by W. Otherwise it is tagged by V meaning that it is a young faulty run. The tag V (young) becomes W (old) when, in the previous state, there was no tag W in the top stack symbol. A tag W is unchanged along the run.
Local transitions. Given an observed local event $o \in \Sigma_{o,\natural}$, from the state *run* with top stack symbol *bel*, there is a local transition (run, bel, o, run, bel') in $\mathcal{A}(\mathcal{V})$ looping over *run* that encodes the possible signalling runs with observation o in $\mathcal{V}$. More precisely for every transition sequence $(q, \alpha) \stackrel{o}{\Rightarrow} (r, \beta)$ in $\mathcal{V}$ (*i.e.* a sequence of unobservable local events ended by an event e with $\mathcal{P}(e) = o$) and $\frac{\alpha,\mathsf{X},q}{\alpha^-,\mathsf{X}^-,q^-} \in bel$ one inserts $\frac{\beta,\mathsf{Y},r}{\alpha^-,\mathsf{X}^-,q^-}$ in bel'. The value of Y follows the rules of tag updates.

Example 5.6. *In the VPA $\mathcal{A}(\mathcal{V})$ of Figure 5.5 there are several transitions corresponding to the transition $(q_0, \gamma, \mathsf{serve}, q_1, \gamma)$ of $\mathcal{V}$ including $(run, \{\frac{\gamma,\mathsf{U},q_0}{\gamma,\mathsf{U},q_0}\}, \mathrm{loc}, run, \{\frac{\gamma,\mathsf{U},q_1}{\gamma,\mathsf{U},q_0}\})$. For example, the runs represented in Figure 5.5 use this transition.*

Push transitions. Given an observed push event $o \in \Sigma_{o,\sharp}$, from the state *run* with top stack symbol *bel*, there is a push transition $(run, bel, o, run, bel'bel'')$ in $\mathcal{A}(\mathcal{V})$ looping over run that encodes the possible signalling runs with observation o in $\mathcal{V}$. More precisely for every transition sequence $(q, \alpha) \stackrel{o}{\Rightarrow} (r, \beta^-\beta)$ in $\mathcal{V}$ and $\frac{\alpha,\mathsf{X},q}{\alpha^-,\mathsf{X}^-,q^-} \in bel$ one inserts $\frac{\beta^-,\mathsf{Y},r}{\alpha^-,\mathsf{X}^-,q^-}$ in bel' and $\frac{\beta,\mathsf{Y},r}{\beta^-,\mathsf{Y},r}$ in bel''. The value of Y follows the rules of tag updates.

Example 5.7. *In Figure 5.5 several transitions of $\mathcal{A}(\mathcal{V})$ correspond to the transition $(q_0, \bot_0, \mathrm{in}, q_0, \bot_0\gamma)$ of $\mathcal{V}$, including $(run, \{\frac{\bot_0,\mathsf{U},q_0}{\bot_0,\mathsf{U},q_0}\}, \mathrm{in}, run, \{\frac{\bot_0,\mathsf{U},q_0}{\bot_0,\mathsf{U},q_0}\}\{\frac{\gamma,\mathsf{U},q_0}{\bot_0,\mathsf{U},q_0}\})$ and several transitions of $\mathcal{A}(\mathcal{V})$ correspond to the transition $(q_0, \gamma, \mathrm{in}, q_0, \gamma\gamma)$ of $\mathcal{V}$, including $(run, \{\frac{\gamma,\mathsf{U},q_0}{\bot_0,\mathsf{U},q_0}\}, \mathrm{in}, run, \{\frac{\gamma,\mathsf{U},q_0}{\bot_0,\mathsf{U},q_0}\}\{\frac{\gamma,\mathsf{U},q_0}{\gamma,\mathsf{U},q_0}\})$. Here, the specification of the tag updates is straightforward since it does not involve faulty runs. The runs represented in Figure 5.5 use these two transitions from the initial state.*

Pop transitions. Given an observed pop event $o \in \Sigma_{o,\flat}$, from the state *run* with top stack symbol *bel*, the "pop operation" is performed by a sequence of two transitions: a pop transition labelled by o reaching another state that contains some information. This information is then used by the next (local) transition labelled by ε to move back to state *run* with a consistent stack symbol. Given an intermediate stack symbol, there is exactly one possible such transition. Thus despite these transitions, $\mathcal{A}(\mathcal{V})$ is still deterministic. The first transition $(run, bel, o, \ell, \varepsilon)$ in $\mathcal{A}(\mathcal{V})$ is specified as follows. The next state ℓ is a set of items of the following shape $\frac{\mathsf{X},q}{\alpha^-,\mathsf{X}^-,q^-}$. More precisely for every transition sequence $(q, \alpha) \stackrel{o}{\Rightarrow} (r, \varepsilon)$ in $\mathcal{V}$ (i.e. a sequence of unobservable local events ended by an event e with $\mathcal{P}(e) = o$) and $\frac{\alpha,\mathsf{X},q}{\alpha^-,\mathsf{X}^-,q^-} \in bel$ one inserts $\frac{\mathsf{Y},r}{\alpha^-,\mathsf{X}^-,q^-}$ in ℓ. The value of Y follows the rules of tag updates. A transition $(\ell, bel, \varepsilon, run, bel')$ is specified as follows. For every $\frac{\mathsf{X}',q'}{\gamma,\mathsf{X},q}$ in ℓ and $\frac{\gamma,\mathsf{X},q}{\gamma^-,\mathsf{X}^-,q^-}$ in bel (*i.e.* the denominator of the first fraction and the numerator of the second fraction match), one inserts $\frac{\gamma,\mathsf{X}',q'}{\gamma^-,\mathsf{X}^-,q^-}$ in bel'.

Example 5.8. *Let us describe how the* pop *is performed by two transitions in the runs of the VPA of Figure 5.5 from the state reached after event* serve*. From q_1 with γ as top of the stack there are two transitions whose observation is pop: $(q_1, \gamma, \mathsf{out}, q_1, \varepsilon)$ and $(q_1, \gamma, \mathsf{abort}, f_1, \varepsilon)$. Thus starting from run with top stack symbol $\{\frac{\gamma,\mathsf{U},q_1}{\gamma,\mathsf{U},q_0}\}$, one reaches state $\ell = \{\frac{\mathsf{U},q_1}{\gamma,\mathsf{U},q_0}, \frac{\mathsf{W},f_1}{\gamma,\mathsf{U},q_0}\}$. The faulty run is tagged with W as there was no tag W in the former top stack symbol. In the next configuration, the top stack symbol is $\{\frac{\gamma,\mathsf{U},q_0}{\perp_0,\mathsf{U},q_0}\}$. So the transition labelled by ε moves back to state run with updated top stack symbol $\{\frac{\gamma,\mathsf{U},q_1}{\perp_0,\mathsf{U},q_0}, \frac{\gamma,\mathsf{W},f_1}{\perp_0,\mathsf{U},q_0}\}$.*

We now give the definition of the estimate VPA $\mathcal{A}(\mathcal{V})$ associated with a given POpVPA $\mathcal{V}$. Let $\mu \in \{g, c, f\}$ we write $(q, \gamma) \stackrel{o}{\Rightarrow}_\mu (q', w)$ with $o \in \Sigma_o$ if when $\mu = g$ (resp. c, f), there exists a (resp. correct, faulty) run of transitions starting from (q, γ) to (q', w) such that all transitions are unobservable except the last one labelled by e with $\mathcal{P}(e) = o$. Let ρ be such a run then we also write $(q, \gamma) \stackrel{\rho}{\Rightarrow}_\mu (q', w)$ All transitions of such runs are local except the last one whose type depends on the type of o.

Definition 5.6. *Given $\langle \mathcal{V}, \mathcal{P}, \Sigma_o \rangle$ a POpVPA with $\mathcal{V} = (Q, \Sigma, \Gamma, \delta, \mathbf{P})$, its* estimate VPA *is the deterministic VPA $\mathcal{A}(\mathcal{V}) = (Q^e, \Sigma_o, \Gamma^e, \delta^e)$ defined by:*

- *$Q^e = \{run\} \uplus (2^{\Gamma \times (\mathsf{Tg} \times Q)^2} \setminus \emptyset)$ is the set of states with initial state $q_0^e = run$;*
- *$\Gamma^e = 2^{(\Gamma \times \mathsf{Tg} \times Q)^2} \setminus \emptyset$ is the stack alphabet with set of bottom stack symbols $\Gamma^e_\perp = 2^{Init} \setminus \emptyset$ where $Init = \{\frac{\perp_0,X,q}{\perp_0,U,q_0} \mid (X, q) \in \mathsf{Tg} \times Q\}$ and initial stack symbol $\perp_0^e = \{\frac{q_0,\mathsf{U},\perp_0}{q_0,\mathsf{U},\perp_0}\}$;*
- *The transition relation δ^e is defined as follows.*

local transitions *$(run, bel, o, run, bel') \in \delta^e$ if:*

- *$\frac{\beta,\mathsf{U},r}{\alpha^-,\mathsf{U},q^-} \in bel'$ iff there exists $\frac{\alpha,\mathsf{U},q}{\alpha^-,\mathsf{U},q^-} \in bel$ and $(q, \alpha) \stackrel{o}{\Rightarrow}_c (r, \beta)$.*
- *If W occurs in bel, $\frac{\beta,\mathsf{W},r}{\alpha^-,\mathsf{X},q^-} \in bel'$ iff there exists $\frac{\alpha,\mathsf{W},q}{\alpha^-,\mathsf{X},q^-} \in bel$ and $(q, \alpha) \stackrel{o}{\Rightarrow}_g (r, \beta)$.*
- *If W occurs in bel, $\frac{\beta,\mathsf{V},r}{\alpha^-,\mathsf{X},q^-} \in bel'$ iff*
  (1) there exists $\frac{\alpha,\mathsf{U},q}{\alpha^-,\mathsf{U},q^-} \in bel$ and $(q, \alpha) \stackrel{o}{\Rightarrow}_f (r, \beta)$ or
  (2) there exists $\frac{\alpha,\mathsf{V},q}{\alpha^-,\mathsf{X},q^-} \in bel$ and $(q, \alpha) \stackrel{o}{\Rightarrow}_g (r, \beta)$.
- *If W does not occur in bel, $\frac{\beta,\mathsf{W},r}{\alpha^-,\mathsf{X},q^-} \in bel'$ iff*
  (1) there exists $\frac{\alpha,\mathsf{U},q}{\alpha^-,\mathsf{X},q^-} \in bel$ and $(q, \alpha) \stackrel{o}{\Rightarrow}_f (r, \beta)$ or
  (2) there exists $\frac{\alpha,\mathsf{V},q}{\alpha^-,\mathsf{X},q^-} \in bel$ and $(q, \alpha) \stackrel{o}{\Rightarrow}_g (r, \beta)$.

push transitions *$(run, bel, o, run, bel' bel'') \in \delta^e$ if:*

- *$\frac{\beta^-,\mathsf{U},r}{\alpha^-,\mathsf{U},q^-} \in bel'$ and $\frac{\beta,\mathsf{U},r}{\beta^-,\mathsf{U},r} \in bel''$ iff there exists $\frac{\alpha,\mathsf{U},q}{\alpha^-,\mathsf{U},q^-} \in bel$ and $(q, \alpha) \stackrel{o}{\Rightarrow}_c (r, \beta^-\beta)$.*

- *If* W *occurs in bel,* $\frac{\beta^-,\mathsf{W},r}{\alpha^-,\mathsf{X},q^-} \in bel'$ *and* $\frac{\beta,\mathsf{W},r}{\beta^-,\mathsf{W},r} \in bel''$ *iff there exists* $\frac{\alpha,\mathsf{W},q}{\alpha^-,\mathsf{X},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_g (r,\beta^-\beta)$.
- *If* W *occurs in bel,* $\frac{\beta^-,\mathsf{V},r}{\alpha^-,\mathsf{X},q^-} \in bel'$ *and* $\frac{\beta,\mathsf{V},r}{\beta^-,\mathsf{V},r} \in bel''$ *iff*
  (1) there exists $\frac{\alpha,\mathsf{U},q}{\alpha^-,\mathsf{U},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_f (r,\beta^-\beta)$ *or*
  (2) there exists $\frac{\alpha,\mathsf{V},q}{\alpha^-,\mathsf{X},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_g (r,\beta^-\beta)$.
- *If* W *does not occur in bel,* $\frac{\beta^-,\mathsf{W},r}{\alpha^-,\mathsf{X},q^-} \in bel'$ *and* $\frac{\beta,\mathsf{W},r}{\beta^-,\mathsf{W},r} \in bel''$ *iff*
  (1) there exists $\frac{\alpha,\mathsf{U},q}{\alpha^-,\mathsf{U},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_f (r,\beta^-\beta)$ *or*
  (2) there exists $\frac{\alpha,\mathsf{V},q}{\alpha^-,\mathsf{X},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_g (r,\beta^-\beta)$.

pop transitions $(run, bel, o, \ell, \varepsilon) \in \delta^e$ *with* $\ell \in Q^e \setminus \{run\}$ *if:*

- $\frac{\mathsf{U},r}{\alpha^-,\mathsf{U},q^-} \in \ell$ *iff* $\frac{\alpha,\mathsf{U},q}{\alpha^-,\mathsf{U},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_c (r,\varepsilon)$.
- *If* W *occurs in bel,* $\frac{\mathsf{W},r}{\alpha^-,\mathsf{X},q^-} \in \ell$ *iff there exists* $\frac{\alpha,\mathsf{W},q}{\alpha^-,\mathsf{X},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_g (r,\varepsilon)$.
- *If* W *occurs in bel,* $\frac{\mathsf{V},r}{\alpha^-,\mathsf{X},q^-} \in \ell$ *iff*
  (1) there exists $\frac{\alpha,\mathsf{U},q}{\alpha^-,\mathsf{U},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_f (r,\varepsilon)$ *or*
  (2) there exists $\frac{\alpha,\mathsf{V},q}{\alpha^-,\mathsf{X},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_g (r,\varepsilon)$.
- *If* W *does not occur in bel,* $\frac{\mathsf{W},r}{\alpha^-,\mathsf{X},q^-} \in \ell$ *iff*
  (1) there exists $\frac{\alpha,\mathsf{U},q}{\alpha^-,\mathsf{U},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_f (r,\varepsilon)$ *or*
  (2) there exists $\frac{\alpha,\mathsf{V},q}{\alpha^-,\mathsf{X},q^-} \in bel$ *and* $(q,\alpha) \overset{o}{\Rightarrow}_g (r,\beta^-\beta)$.

ε-transitions $(\ell, bel, \varepsilon, run, bel') \in \delta^e$ *if:*
$\frac{\alpha,\mathsf{X}',r}{\alpha^-,\mathsf{X}^-,q^-} \in bel'$ *iff there exists* $\frac{\alpha,\mathsf{X},q}{\alpha^-,\mathsf{X}^-,q^-} \in bel$ *and* $\frac{\mathsf{X}',r}{\alpha,\mathsf{X},q} \in \ell$.

We say that a configuration is *stable* if its associated state is *run*.

Example 5.9. *Let us explain the runs given in the Figure 5.5. It starts in the initial configuration* $(run, \left|\{\frac{\perp_0,\mathsf{U},q_0}{\perp_0,\mathsf{U},q_0}\}\right|)$ *which represents the empty run.*

From q_0 *there exists only one path of observation* in *in the POpVPA. As this path is correct, by reading* in *on the estimate VPA we reach* $(run, \left|\begin{matrix}\{\frac{\gamma,\mathsf{U},q_0}{\perp_0,\mathsf{U},q_0}\}\\ \{\frac{\perp_0,\mathsf{U},q_0}{\perp_0,\mathsf{U},q_0}\}\end{matrix}\right|)$. *The new element of the stack* $\{\frac{\gamma,\mathsf{U},q_0}{\perp_0,\mathsf{U},q_0}\}$ *means that the stack of the possible run has head* γ *and its current state is* q_0 *after a correct run, moreover the run entered* q_0 *when it pushed this* γ *and it does not have a second non-terminal element in our stack. Reading a second* in *is still doable by a single run, we reach* $(run, \left|\begin{matrix}\{\frac{\gamma,\mathsf{U},q_0}{\gamma,\mathsf{U},q_0}\}\\ \{\frac{\gamma,\mathsf{U},q_0}{\perp_0,\mathsf{U},q_0}\}\\ \{\frac{\perp_0,\mathsf{U},q_0}{\perp_0,\mathsf{U},q_0}\}\end{matrix}\right|)$ *which modifies one information compared to before: we know from the bottom part of the head stack that the stack has at least a second* γ.

Reading a serve *then is possible as there exists a correct signalling run from* q_0 *to* q_1 *with observation* serve. *The estimate VPA modifies the head stack so as to represent that the run we follow is now in* q_1 *but without modifying anything else.*

Reading a pop event is more involved: from q_1 *with head of stack* γ, *triggering a* pop *can be done by a correct run staying in* q_1 *or by a faulty run going in* f_1. *To represent this and the popping of the stack, we go in two steps. In the first step, we go to the state* $\{\frac{\mathsf{U},q_1}{\gamma,\mathsf{U},q_0}, \frac{\mathsf{W},f_1}{\gamma,\mathsf{U},q_0}\}$ *which keeps the information of the two possibilities of current configuration and we pop the stack. In the second step, we deterministically take an* ε *transition that transfer this information from the state to the stack. In order to transfer the information, the estimate VPA checks which of the current possible runs (represented by* $\frac{\mathsf{U},q_1}{\gamma,\mathsf{U},q_0}$ *and* $\frac{\mathsf{W},f_1}{\gamma,\mathsf{U},q_0}$*) corresponds to each of the new head of stack. This is done by comparing the bottom part of the run with the top part of the head of stack, here* γ, U, q_0 *in every cases. Reading a second* pop *realises a similar process reaching* $(run, \left|\{\frac{\perp_0,\mathsf{U},q_1}{\perp_0,\mathsf{U},q_0}, \frac{\perp_0,\mathsf{W},f_1}{\perp_0,\mathsf{U},q_0}\}\right|)$. *An* empty *would lead to* $(run, \left|\{\frac{\perp_0,\mathsf{U},q_0}{\perp_0,\mathsf{U},q_0}\}\right|)$ *as there is a correct run from* q_1 *to* q_0 *labelled by* empty *but no run from* f_1 *with such label. Similarly a* reset *cannot be taken from* q_1 *but it can be read from* f_1, *thus we reach* $(run, \left|\{\frac{\perp_0,\mathsf{W},q_0}{\perp_0,\mathsf{U},q_0}\}\right|)$.

The following proposition links runs of $\mathcal{V}$ and observed sequences of $\mathcal{A}(\mathcal{V})$.

Proposition 5.1. *Let* σ *be an observed sequence of* $\mathcal{A}(\mathcal{V})$ *and* ρ^* *be its corresponding finite run with successive stable configurations* $(run, w_0) \dots (run, w_n)$. *Let* $w_n = bel_1 \dots bel_h$ *and for* $i < n$, $bel^{(i)}$ *be the top stack symbol of* w_i. *Then:*
• *For all* $\frac{\gamma_h,\mathsf{X}_h,q_h}{\gamma_{h-1},\mathsf{X}_{h-1},q_{h-1}} \in bel_h$, *there exists a sequence* $(\frac{\gamma_i,\mathsf{X}_i,q_i}{\gamma_{i-1},\mathsf{X}_{i-1},q_{i-1}})_{0<i<h}$ *such that for all* i, $\frac{\gamma_i,\mathsf{X}_i,q_i}{\gamma_{i-1},\mathsf{X}_{i-1},q_{i-1}} \in bel_i$ *and a signalling run* ρ *of* $\mathcal{V}$ *such that* $\mathcal{P}(\rho) = \sigma$ *that reaches configuration* $(q_h, \gamma_1 \dots \gamma_h)$. *In addition:*

- *if* $\mathsf{X}_h = \mathsf{U}$ *then* ρ *may be chosen correct;*
- *if* $\mathsf{X}_h \neq \mathsf{U}$ *then* ρ *may be chosen faulty;*
- *if* $\mathsf{X}_h = \mathsf{W}$ *then there exists* $0 < k \leq n$, *such that* $\rho_{\downarrow k}$ *is faulty and* W *does not occur in* $bel^{(k-1)}$.

• *Conversely, let* ρ *be a signalling run of* $\mathcal{V}$ *such that* $\mathcal{P}(\rho) = \sigma$ *reaching configuration* $(q_h, \gamma_1 \dots \gamma_h)$, *there exists a sequence* $(\frac{\gamma_i,\mathsf{X}_i,q_i}{\gamma_{i-1},\mathsf{X}_{i-1},q_{i-1}})_{0<i\leq h}$ *such that for all* i, $\frac{\gamma_i,\mathsf{X}_i,q_i}{\gamma_{i-1},\mathsf{X}_{i-1},q_{i-1}} \in bel_i$. *In addition:*

- *if* ρ *is correct then* $\mathsf{X}_h = \mathsf{U}$;
- *if* ρ *is faulty then* $\mathsf{X}_h \neq \mathsf{U}$;
- *if there exists* $0 < k \leq n$, *such that* $\rho_{\downarrow k}$ *is faulty and* W *does not occur in* $bel^{(k-1)}$ *then* $\mathsf{X}_h = \mathsf{W}$.

The difficulty of this proof is the number of cases that have to be studied: what is the tag and which kind of transition (local, push or pop) is considered. As a consequence,

we only detail the most involved case (when the event is a pop and the tag is W. The result is obtained by induction on the size of the observed sequence and mostly consists in understanding the definitions of $\mathcal{A}(\mathcal{V})$ and especially of the tag updates.

Proof. We prove the result by induction on $|\sigma|$. The basis case is straightforward. For the inductive step, we only detail the most involved case: $\sigma[n] \in \Sigma_{o,\flat}$. For the properties related to tags, we only detail the ones related to W. Denote $\sigma' = \sigma[1]\dots\sigma[n-1]$ and $w_{n-1} = bel'_1 \dots bel'_h bel'_{h+1}$.

• Let $\frac{\gamma_h,\mathsf{X}_h,q_h}{\gamma_{h-1},\mathsf{X}_{h-1},q_{h-1}} \in bel_h$. By construction, there exists $\frac{\gamma'_{h+1},\mathsf{X}'_{h+1},q'_{h+1}}{\gamma'_h,\mathsf{X}'_h,q'_h} \in bel'_{h+1}$ with $\gamma'_h = \gamma_h$, a signalling run $(q'_{h+1},\gamma'_{h+1}) \overset{\rho''}{\Longrightarrow} (q_h,\varepsilon)$ with $proj(\rho'') = \sigma[n]$, $\frac{\gamma'_h,\mathsf{X}'_h,q'_h}{\gamma'_{h-1},\mathsf{X}'_{h-1},q'_{h-1}} \in bel'_h$ where $(\gamma'_{h-1},\mathsf{X}'_{h-1},q'_{h-1}) = (\gamma_{h-1},\mathsf{X}_{h-1},q_{h-1})$ and X_h is obtained by updating X'_{h+1} w.r.t. bel'_{h+1} and ρ''. In particular if $\mathsf{X}_h = \mathsf{W}$ then:
(1) $\mathsf{X}'_{h+1} = \mathsf{W}$, or
(2) W does not occurs in bel'_{h+1} and (a) $\mathsf{X}'_{h+1} = V$ or (b) $\mathsf{X}'_{h+1} = U$ and ρ'' is faulty.
By inductive hypothesis, there exists a sequence $(\frac{\gamma'_i,\mathsf{X}'_i,q'_i}{\gamma'_{i-1},\mathsf{X}'_{i-1},q'_{i-1}})_{0<i\leq h}$ such that for all i, $\frac{\gamma'_i,\mathsf{X}'_i,q'_i}{\gamma'_{i-1},\mathsf{X}'_{i-1},q'_{i-1}} \in bel'_i$ and a signalling run ρ' of $\mathcal{V}$ such that $\mathcal{P}(\rho') = \sigma'$ reaching configuration $(q'_{h+1},\gamma'_1\dots\gamma'_{h+1})$. Consider the signalling run $\rho = \rho'\rho''$; it reaches configuration $(q_h,\gamma'_1\dots\gamma'_h)$. Since for all $i < h$, $bel'_i = bel_i$, the sequence $(\frac{\gamma'_i,\mathsf{X}'_i,q'_i}{\gamma'_{i-1},\mathsf{X}'_{i-1},q'_{i-1}})_{0<i<h}$ and the run ρ are appropriate. The three additional properties follow from the rules of tag updates. In particular, if $\mathsf{X}_h = \mathsf{W}$, then:
∘ the assertion (1) holds and then the property comes from the inductive hypothesis, or
∘ the assertion (2) holds which implies that W does not occur in bel'_{h+1} and ρ is faulty.
• Let ρ be a signalling run of $\mathcal{V}$ such that $\mathcal{P}(\rho) = \sigma$ which reaches configuration $(q_h,\gamma_1\dots\gamma_h)$. Let us write $\rho = \rho_{\downarrow n-1}\rho''$ with $(q'_{h+1},\gamma'_{h+1}) \overset{\rho''}{\Longrightarrow} (q_h,\varepsilon)$. By the inductive hypothesis, there exists a sequence $(\frac{\gamma'_i,\mathsf{X}'_i,q'_i}{\gamma'_{i-1},\mathsf{X}'_{i-1},q'_{i-1}})_{0<i\leq h+1}$ such that for all i, $\frac{\gamma'_i,\mathsf{X}'_i,q'_i}{\gamma'_{i-1},\mathsf{X}'_{i-1},q_{i-1}} \in bel'_i$ and for all $i \leq h$, $\gamma'_i = \gamma_i$. By construction, $\frac{\gamma_h,\mathsf{X}_h,q_h}{\gamma'_{h-1},\mathsf{X}'_{h-1},q'_{h-1}} \in bel_h$ for some X_h. Since $bel_i = bel'_i$ for all $i < h$, we obtain the required sequence of items.

The three additional properties follow from the rules of tag updates. In particular, assume there exists $0 < k \leq n$, such that $\rho_{\downarrow k}$ is faulty and W does not occur in bel^{k-1}.
∘ If $\rho_{\downarrow n-1}$ is correct then, as ρ is faulty, ρ'' is faulty and W does not occur in $bel^{n-1} = bel'_{h+1}$. So by construction $\mathsf{X}_h = \mathsf{W}$.
∘ If $\rho_{\downarrow n-1}$ is faulty then:

- either $\mathsf{X}'_{h+1} = \mathsf{W}$ and by construction $\mathsf{X}_h = \mathsf{W}$,
- or $\mathsf{X}'_{h+1} = \mathsf{V}$. By induction hypothesis there does not exist $0 < k \leq n-1$, such that $\rho_{\downarrow k}$ is faulty and W does not occur in bel^{k-1}. So W does not occur in $bel^{n-1} = bel'_{h+1}$. Therefore $\mathsf{X}_h = \mathsf{W}$.

□

2.2 Decidability of diagnosability for POpVPA

Our goal in this subsection is to use the characterisations from the Section 3 of Chapter 3 to decide the diagnosability of POpLTS generated by POpVPA. To do so, we face the difficulty that the Borel sets that characterise IF-, IA- and FF-diagnosability are not *a priori* regular, even in the finitely-branching case. Yet, for POpVPA, we circumvent this problem, and manage to specify these sets by pLTL formula on a product of the POpVPA with its estimate VPA, the tags are used to define the atomic propositions. The decidability of the qualitative model checking for recursive probabilistic systems [EY12] then yields the decidability of the above three diagnosability notions for POpVPA.

The first step is to build the product of the POpVPA and its estimate VPA. From this point on, we assume that the set of states of the POpVPA is separated between correct and faulty states $Q = Q_c \cup Q_f$. This can be done without loss of generality: the transformation ensuring this is similar to the one shown for pLTS in Section 1.4 of Chapter 2. We build $\mathcal{V}_{\mathcal{A}(\mathcal{V})} = \mathcal{V} \times \mathcal{A}(\mathcal{V})$ the product automaton of $\mathcal{V}$ and $\mathcal{A}(\mathcal{V})$ synchronised on the alphabet of observed events Σ_o. The transitions of $\mathcal{V}$ labelled by unobservable events do not change the second component of the state and the transitions of $\mathcal{A}(\mathcal{V})$ labelled by ε do not change the first component of the state. Due to the determinism of $\mathcal{A}(\mathcal{V})$, $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$ has the same probabilistic behaviour as the one of $\mathcal{V}$ except that it memorises additional information along the run. More precisely, let ρ be a run of $\mathcal{V}$, then $\bar{\rho}$, a run of $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$, is obtained from ρ by following the same transitions and adding the single $\ominus$ transition firable after any pop transition. One immediately gets $\mathbb{P}_{\mathcal{V}_{\mathcal{A}(\mathcal{V})}}(\bar{\rho}) = \mathbb{P}_{\mathcal{V}}(\rho)$. Formally we have:

Definition 5.7. *Given $\langle \mathcal{V}, \mathcal{P}, \Sigma_o \rangle$ a POpVPA $\mathcal{V} = (Q, \Sigma, \Gamma, \delta, \mathbf{P})$ and its estimate VPA $\mathcal{A}(\mathcal{V}) = (Q^e, run, \Sigma_o, \Gamma^e, \delta^e)$, their synchronised product is the pVPA $\mathcal{V}_{\mathcal{A}(\mathcal{V})} = (Q^{\mathcal{A}}, \Sigma \cup \{\ominus\}, \Gamma^{\mathcal{A}}, \delta^{\mathcal{A}}, \mathbf{P}^{\mathcal{A}})$ where:*

- *$Q^{\mathcal{A}} = Q \times Q^e$ is the set of states with initial state $q_0^{\mathcal{A}} = (q_{0,c}, run)$;*
- *$\Gamma^{\mathcal{A}} = \Gamma \times \Gamma^e$ is the stack alphabet with $\Gamma_\perp \times \Gamma^e_\perp$ the set of bottom stack symbols and $\perp_0^{\mathcal{A}} = (\perp_0, \frac{\perp_0, \mathsf{U}, q_0}{\perp_0, \mathsf{U}, q_0})$ the initial symbol;*
- *The transition relation $\delta^{\mathcal{A}}$ consists of:*

  local transitions.

  - *For all $(q, \gamma, a, q', \gamma') \in \delta$ with a unobservable and $bel \in \Gamma^e$, we have $((q, run), (\gamma, bel), a, (q', run), (\gamma', bel)) \in \delta^{\mathcal{A}}$;*
  - *For all $(q, \gamma, a, q', \gamma') \in \delta$ and $(run, bel, o, run, bel') \in \delta^e$ with $\mathcal{P}(a) = o$, we have $((q, run), (\gamma, bel), a, (q', run), (\gamma', bel')) \in \delta^{\mathcal{A}}$;*
  - *For all $(\ell, bel, \varepsilon, run, bel') \in \delta^e$, $q \in Q$ and $\gamma \in \Gamma$, we have*

  $$((q, \ell), (\gamma, bel), \ominus, (q, run), (\gamma, bel')) \in \delta^{\mathcal{A}};$$

  push transitions.

  - *For all $(q, \gamma, a, q', \gamma'\gamma'') \in \delta$ and $(run, bel, o, run, bel'bel'') \in \delta^e$ with $\mathcal{P}(a) = o$, we have $((q, run), (\gamma, bel), a, (q', run), (\gamma', bel')(\gamma'', bel'')) \in \delta^{\mathcal{A}}$;*

pop transitions.

• *For all* $(q, \gamma, a, q', \varepsilon) \in \delta$ *and* $(run, bel, o, \ell, \varepsilon) \in \delta^e$ *with* $\mathcal{P}(a) = o$*, we have* $((q, run), (\gamma, bel), a, (q', \ell), \varepsilon) \in \delta^{\mathcal{A}}$*;*

- *The transition probability function* $\mathbf{P}^{\mathcal{A}}$ *is defined by:*
  - $\mathbf{P}^{\mathcal{A}}((q, run), (\gamma, bel), a, (q', run), (\gamma', bel')) = \mathbf{P}(q, \gamma, a, q', \gamma')$*;*
  - $\mathbf{P}^{\mathcal{A}}((q, run), (\gamma, bel), a, (q', run), (\gamma', bel')(\gamma'', bel'')) = \mathbf{P}(q, \gamma, a, q', \gamma'\gamma'')$*;*
  - $\mathbf{P}^{\mathcal{A}}((q, run), (\gamma, bel), a, (q', \ell), \varepsilon) = \mathbf{P}(q, \gamma, a, q', \varepsilon)$*;*
  - *for* $\ell \in Q^e \setminus \{run\}, \mathbf{P}^{\mathcal{A}}((q, \ell), (\gamma, bel), \ominus, (q, run), (\gamma, bel')) = 1$*.*

Example 5.10. *The product POpVPA contains the current run of the POpVPA and the information given by the estimate POpVPA. Let us consider the faulty run given as example in the Figure 5.2. This POpVPA does not follow the separation between correct and faulty states. Here we write* q_c *(resp.* q_f*) if the state* q *was reached by a correct (resp. faulty) run. After reading in, we are in state* $(q_{0,c}, run)$ *meaning that the state of the possible run is* q_0*, it was reached by a correct run and our estimate VPA is in state run, the head of stack is* $(\gamma, \left|\begin{smallmatrix}\{\frac{\gamma,\mathsf{U},q_{0,c}}{\perp_0,\mathsf{U},q_{0,c}}\}\\ \{\frac{\perp_0,\mathsf{U},q_{0,c}}{\perp_0,\mathsf{U},q_{0,c}}\}\end{smallmatrix}\right|)$*, meaning our real head is* γ *and the rest is the head of the estimate VPA. If we follow the faulty run until after the first pop, we reach the state* $(f_{1,f}, \{\frac{\mathsf{U},q_1}{\gamma,\mathsf{U},q_{0,c}}, \frac{\mathsf{W},f_{1,f}}{\gamma,\mathsf{U},q_{0,c}}\})$*, we are thus in* f_1 *with a faulty run and the estimate VPA is in one of the temporary states. In order to leave this state, we read* $a \ominus$ *which leads to the state* $(f_{1,f}, run)$*.* $\ominus$ *is an event only affecting the part of the POpVPA corresponding to the estimate VPA, allowing it to realise the* ε *transition that follows the observation of a pop event.*

Given a finite run ρ of $\mathcal{V}$, we inductively define the run $\bar{\rho}$ of $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$ as follows. First $\overline{(q_0, \perp_0)} = (q_0^{\mathcal{A}}, \perp_0^{\mathcal{A}})$. Let ρ of length $n \geq 1$, $a \in \Sigma$ and $q \in Q$ and $\gamma_1, \ldots, \gamma_h \in \Gamma$ such that $\rho = \rho' a(q, \gamma_1 \ldots \gamma_h)$. If $a \notin \Sigma_\flat$ then $\bar{\rho} = \bar{\rho}' a((q, run), (\gamma_1, bel_1) \ldots (\gamma_h, bel_h))$ where $(run, bel_1 \ldots bel_h)$ is the configuration reached by $\mathcal{P}(\rho)$ in $\mathcal{A}(\mathcal{V})$. If $a \in \Sigma_\flat$ then $\bar{\rho} = \bar{\rho}' a((q, \ell), (\gamma_1, bel_1) \ldots (\gamma_h, bel_h)) \ominus ((q, run), (\gamma_1, bel_1) \ldots (\gamma_{h-1}, bel_{h-1})(\gamma_h, bel'_h))$ where $(\ell, bel_1 \ldots bel_h)$ is the configuration reached by $\mathcal{P}(\rho)$ in $\mathcal{A}(\mathcal{V})$ and $(run, bel_1 \ldots bel_{h-1} bel'_h)$ is the single next configuration reached by an ε transition. As previously observed, $\mathbb{P}(\rho) = \mathbb{P}(\bar{\rho})$.

In order to prove decidability of diagnosability for a POpVPA $\mathcal{V}$, one wants to check whether the formulae characterising diagnosability defined in Chapter 3 hold on $\mathcal{V}$. Let us first recall the relevant results of Chapter 3. We defined three path formulae:

- $\mathfrak{f}$: for every run ρ, $\mathfrak{f}(\rho) = \mathsf{true}$ if ρ is faulty;
- $\mathfrak{U}$: for every run ρ, $\mathfrak{U}(\rho) = \mathsf{true}$ if there exists a correct signalling run ρ' with $\mathcal{P}(\rho) = \mathcal{P}(\rho')$;
- $\mathfrak{W}$: $\mathfrak{W}(\varepsilon) = \mathsf{false}$ and $\mathfrak{W}(q_0 a_0 \ldots q_{n+1}) = \mathsf{true}$ if

$$\mathsf{firstf}(\mathcal{P}(q_0 a_0 \ldots q_{n+1})) = \mathsf{firstf}(\mathcal{P}(q_0 a_0 \ldots q_n)) < \infty$$

where $\mathsf{firstf}(\sigma) = \min\{k \mid \exists \rho$ signalling run $\mathcal{P}(\rho) = \sigma \wedge \rho_{\downarrow k}$ is faulty$\}$ with the convention that $\min(\emptyset) = \infty$.

Using these path formulae, given a pLTS $\mathcal{A}$, we obtained the following results:

- $\mathcal{A}$ is FF-diagnosable iff $\mathcal{A} \models \mathbb{P}^{=0}(\Diamond\Box(\mathfrak{f} \wedge \mathfrak{U}))$;
- if $\mathcal{A}$ is finitely branching, $\mathcal{A}$ is IA-diagnosable iff $\mathcal{A} \models \mathbb{P}^{=0}(\Diamond\Box(\mathfrak{U} \wedge \mathfrak{W}))$.

The paths formulae $\mathfrak{f}$, $\mathfrak{U}$ and $\mathfrak{W}$ depends on the past of the run and not only on the current configuration. We therefore transform the pathL formulae into pLTL properties that are checked on $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$. First, for each path formula we define an atomic propositions on the pairs $((q, run)(\gamma, bel))$ consisting of a state of $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$ together with a top stack contents.

Definition 5.8. *Let $bel \subseteq 2^{(\Gamma \times \mathsf{Tg} \times Q)^2}$, we say that the tag* X occurs *in bel if there exists $\frac{\gamma,\mathsf{X},q}{\gamma^-,\mathsf{X}^-,q^-} \in bel$.*

The atomic propositions ν_f, ν_u and ν_w corresponding to the path formulae $\mathfrak{f}$, $\mathfrak{U}$ and $\mathfrak{W}$ are defined by:

- $\nu_f((q, run)(\gamma, bel)) = \mathsf{true}$ *if and only if $q \in Q_f$;*
- $\nu_u((q, run)(\gamma, bel)) = \mathsf{true}$ *if and only if* U *occurs in bel;*
- $\nu_w((q, run)(\gamma, bel)) = \mathsf{true}$ *if and only if* W *occurs in bel.*

We extend ν_f, ν_u and ν_w over configurations $cf = ((q, \ell), w))$ with $\ell \neq run$ by $\nu_f(cf) = \nu_u(cf) = \nu_w(cf) = \mathsf{true}$.

The atomic propositions ν_f and ν_u perfectly reflect the paths formula $\mathfrak{f}$ and $\mathfrak{U}$, and ν_w is eventually forever true if and only if $\mathfrak{W}$ is.

Proposition 5.2. *Let ρ be an infinite run of $\mathcal{V}$. Then:*

- *For all $k \in \mathbb{N}$, $\mathfrak{f}(\rho_{\downarrow k}) \Leftrightarrow \nu_f(\mathsf{last}(\bar{\rho}_{\downarrow k}))$ and $\mathfrak{U}(\rho_{\downarrow k}) \Leftrightarrow \nu_u(\mathsf{last}(\bar{\rho}_{\downarrow k}))$;*
- $\rho \models \Diamond\Box\mathfrak{W} \Leftrightarrow \exists K \forall k \geq K.\ \nu_w(\mathsf{last}(\bar{\rho}_{\downarrow k})) = \mathsf{true}$.

The second component of $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$ representing $\mathcal{A}(\mathcal{V})$, one can use the results of Proposition 5.1 to link the tags and the runs associated with the observed sequence. This is what we do here. The most complicated (and interesting) case being the link between $\mathfrak{W}$ and W. The idea is the following. When the tag W disappears after following an observation in $\mathcal{A}(\mathcal{V})$, let n be the observed length of the last time W was not tagging any state, then the oldest fault in the current run occurred after the n'th observation. Thus every time W is not present, the longest prefix of the run that is surely correct increased, ensuring that $\mathfrak{W}$ is $false$. Of course, $\mathfrak{W}$ can be $false$ more often than the absences of W. However, if after the n'th observation, for $n \in \mathbb{N}$, W always tag a state of the belief, it means that there exists a run consistent with the observation for which a fault occurred at most at the n'th step. Therefore firstf is bounded, which implies that $\mathfrak{W}$ will eventually become forever $true$.

Proof. First, note that $\mathfrak{f}$ and ν_f obviously coincide: they both express that a fault occurred.
To prove the second item, about $\mathfrak{U}$ and ν_u, we use the link between observed sequences and the tag U in $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$. Let σ be an observed sequence triggered by a run of $\mathcal{V}$. Then bel_σ is the top stack symbol of the stable configuration in $\mathcal{A}(\mathcal{V})$ reached by the run accepting σ (so ending by an ε-transition if the last event is a pop event). Due to Proposition 5.1, U occurs in bel_σ iff there is a correct signalling run of $\mathcal{V}$ with observed sequence σ. According to the definition of ν_u, we thus deduce that for any finite signalling run ρ of $\mathcal{V}$, $\nu_u(\mathsf{last}(\rho)) = \mathsf{true}$ iff $\mathfrak{U}(\rho) = \mathsf{true}$.
We now establish the link between $\mathfrak{W}$ and ν_W. To show the left-to-right implication, let $\rho \in \Omega$ and $K_0 \in \mathbb{N}$ be such that $\rho, K_0 \models \Box\mathfrak{W}$. By definition of $\mathfrak{W}$, $\mathsf{firstf}(\mathcal{P}(\rho_{\downarrow k}))$ is constant and bounded by K_0 for $k \geq K_0$. For all $k \in \mathbb{N}$, let bel_k be the top stack symbol reached in $\mathcal{A}(\mathcal{V})$ after reading the observed sequence $\mathcal{P}(\rho_{\downarrow k})$. If for all $k \geq K_0$, W occurs in bel_k, then for all $k \geq K_0$, $\nu_w(\mathsf{last}(\bar{\rho}_{\downarrow k})) = \mathsf{true}$. Otherwise there exists $K_1 \geq K_0$ such that W does not occur in bel_{K_1}. Let $k > K_1$, as $\mathsf{firstf}(\mathcal{P}(\rho_{\downarrow k})) \leq K_0$, there exists a faulty run ρ' of $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$ such that $\mathcal{P}(\rho') = \mathcal{P}(\bar{\rho}_{\downarrow n})$ and $\rho'_{\downarrow K_0}$ is faulty. W does not occur in bel_{K_1} and $\rho'_{\downarrow K_1+1}$ is faulty. Thus by Proposition 5.1, W occurs in bel_k. Therefore for all $n > K_1$, $\nu_w(\mathsf{last}(\bar{\rho}_{\downarrow n})) = \mathsf{true}$.
Let us show the right-to-left implication. Let $\rho \in \Omega$ and $K \in \mathbb{N}$ be such that for all $k \geq K$, $\nu_w(\mathsf{last}(\bar{\rho}_{\downarrow k})) = \mathsf{true}$. By definition of ν_w for all $k \geq K$, W occurs in bel_k (defined as above). Let $k \geq K$, by Proposition 5.1, there exists a run ρ' of $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$ such that $\mathcal{P}(\rho') = \mathcal{P}(\bar{\rho}_{\downarrow k})$ and there exists $n \leq k$ such that $\rho'_{\downarrow n}$ is faulty and W does not occur in bel_{n-1}. Thus $n \leq K$. Therefore for all $k \geq K$, $\mathsf{firstf}(\mathcal{P}(\rho_{\downarrow k})) \leq K$. Since beyond K, firstf is bounded, it is non decreasing and then ultimately constant. Let K' such that for all $k \geq K'$, $\mathsf{firstf}(\mathcal{P}(\rho_{\downarrow k})) = \mathsf{firstf}(\mathcal{P}(\rho_{\downarrow k-1}))$. So $\rho, K' \models \Box\mathfrak{W}$ and thus $\rho \models \Diamond\Box\mathfrak{W}$. □

Thanks to the relationships between the path formulae, and the atomic propositions, and using the characterisations from Section 3 of Chapter 3, we reduce the FF-, IF- and IA-diagnosability to the model checking of a pLTL formula on the product VPA $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$. Model checking qualitative pLTL for probabilistic pushdown automata is doable in polynomial space in the size of the model [EY12]. In our case, $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$ is exponential in the size of $\mathcal{V}$. We thus obtain the decidability and a complexity upper-bound for the diagnosability problems for POpVPA.

Theorem 5.3. FF*-diagnosability,* IF*-diagnosability and* IA*-diagnosability are decidable in* EXPSPACE *for POpVPA.*

Proof. Thanks to the Propositions 5.2 and 5.1 and the characterisations of Propositions 3.7 (page 80) and 3.8 (page 81), we can derive pLTL characterisations of diagnosability for POpVPA. Namely, for $\mathcal{V}$ a POpVPA, as $\mathcal{V}$ and $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$ have the same probabilistic behaviour,

- $\mathcal{V}$ is FF-diagnosable iff $\mathcal{V}_{\mathcal{A}(\mathcal{V})} \models \mathbb{P}^{=0}(\Diamond\Box(\nu_f \wedge \nu_u))$;
- $\mathcal{V}$ is IA-diagnosable iff $\mathcal{V}_{\mathcal{A}(\mathcal{V})} \models \mathbb{P}^{=0}(\Diamond\Box(\nu_u \wedge \nu_w))$.

Moreover, since POpPDA have finitely many states, the POpLTS they generate are finitely-branching. Therefore, IF-diagnosability coincides with FF-diagnosability according to Theorem 3.1. The two above qualitative pLTL formulae can be checked on probabilistic pushdown automata thanks to [EY09]. More precisely, one can transform $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$ into a recursive Markov chain (the transformation is linear) [EY12]. Then, the model checking of qualitative pLTL on recursive Markov chains is doable in PSPACE in the size of the Recursive Markov Chain and EXPTIME in the size of the formulae [EY09]. In our case, the product VPA $\mathcal{V}_{\mathcal{A}(\mathcal{V})}$ is exponential in the size of $\mathcal{V}$ and the size of the formulae is constant. This yields an EXPSPACE algorithm for checking diagnosability of POpVPA. □

2.3 EXPTIME-hardness of the diagnosability for POpVPA

While the notions of diagnosability we studied in the previous section are decidable in EXPSPACE (Theorem 5.3), this is not necessarily optimal. Here, we only show an EXPTIME lower bound on the complexity. This lower bound is obtained by reducing the universality problem for VPA, which is known to be EXPTIME-complete [AM04]. This reduction also applies to FA-diagnosability for which the decidability status is unknown.

Theorem 5.4. FF-, IF-, FA- *and* IA-*diagnosability are* EXPTIME-*hard for POpVPA.*

Proof. Let us start with FF-diagnosability. The proof is done by reduction from the universality problem for VPA, which is known to be EXPTIME-hard [AM04]. Recall the universality problem for VPA: given a VPA $\mathcal{A}$ and a set of final states Q_F, do we have $\mathcal{P}(\{\rho \in \mathsf{SR} \mid \mathsf{last}(\rho) \in Q_F\}) = \Sigma_o^\omega$?

Starting from a VPA $\mathcal{A}$ we build a pVPA $\mathcal{V}'$ (see Figure 5.6) with two components: one correct and one faulty, both reachable in one step from the initial state. The correct component is a copy of $\mathcal{A}$ with a positive probability of making a reset (emptying its stack and going back to the initial state of $\mathcal{A}$) in a final state. Every reset starts by a new observable event $\natural$,followed by some pop event $\flat$ and ends by a second $\natural$. In the faulty component, one can read any observation of Σ_o^* and also has the possibility to produce $\natural$ and $\flat$ in a way that mimics a reset. If a $\natural$ is read, then the faulty component triggers some $\flat$ and a $\natural$ as would be done in the correct component. This way, the observation associated with a reset does not give any information on the correctness of a system. What matters is after which observed sequence can a reset occur. If an observed sequence cannot end in an accepting state of $\mathcal{A}$, then in a faulty run, with probability 1 this observed sequence will be read in between two resets, revealing the fault. Reciprocally, if $\mathcal{A}$ is universal, everything that can be observed on a faulty run can also be observed in the correct component establishing that $\mathcal{V}'$ is not diagnosable.

Formally, from a VPA $\mathcal{A} = (Q, \Sigma, \Gamma, \delta)$ and a subset of accepting states $Q_F \subseteq Q$, we build a pVPA $\mathcal{V}' = (Q', \Sigma', \Gamma', \delta', \mathbf{P}')$ as follows:

- $Q' = Q \cup \{f_0, f_\flat, q'_0, q_\flat\}$ and q'_0 is the initial state;
- $\Sigma' = \Sigma \uplus \{\mathbf{f}, u, \flat, \natural\}$;

- $\Gamma' = \Gamma \uplus \{B\}$ and $\Gamma'_\perp = \Gamma_\perp$;
- Writing $\delta_\natural$, resp. $\delta_\sharp$ and $\delta_\flat$ for the set of local resp. push and pop transitions of $\mathcal{V}$, δ' consists of the following transitions:

  local $\delta_\natural \cup \{(q'_0, \perp_0, u, \perp_0, q_0), (q'_0, \perp_0, \mathbf{f}, \perp_0, f_0), (f_0, \gamma, \natural, \gamma, f_\flat) \mid \gamma \in \Gamma \cup \{\perp_0\}\} \cup \{(q, \gamma, \natural, \gamma, q_\flat) \mid q \in Q_F, \gamma \in \Gamma \cup \{\perp_0\}\} \cup \{(f_0, \gamma, a, \gamma, f_0) \mid a \in \Sigma_\natural, \gamma \in \{B, \perp_0\}\} \cup \{(q_\flat, \perp_0, \natural, \perp_0, q_0), (f_\flat, \perp_0, \natural, \perp_0, f_0)\}$;

  push $\delta_\sharp \cup \{(f_0, \gamma, a, \gamma B, f_0) \mid a \in \Sigma_\sharp, \gamma \in \{B, \perp_0\}\}$;

  pop $\delta_\flat \cup \{(f_0, B, a, \varepsilon, f_0) \mid a \in \Sigma_\flat\} \cup \{(f_\flat, B, \flat, \varepsilon, f_\flat)\} \cup \{(q_\flat, \gamma, \flat, \varepsilon, q_\flat) \mid \gamma \in \Gamma\}$;
- $\mathbf{P}'$ is such that for every $\gamma \in \Gamma$, $\mathbf{P}'(f_0, \gamma, \natural, \gamma, f_\flat) = \frac{1}{2}$, and assigns arbitrary positive probabilities to the other transitions in δ'.

We further consider the POpVPA $\langle \mathcal{V}', \Sigma_o, \mathcal{P} \rangle$ with $\Sigma_o = \Sigma \cup \{\flat, \natural\}$ and the masking function satisfies $\mathcal{P}(u) = \mathcal{P}(\mathbf{f}) = \varepsilon$ and $\mathcal{P}(x) = x$ for any other event $x \in \Sigma'$. This construction is illustrated in Figure 5.6. The figure uses the following shortcuts: $a_\flat \in \Sigma_\flat$, $a_\natural \in \Sigma_\natural$, $a_\sharp \in \Sigma_\sharp$, $\gamma \in \Gamma$, $\gamma' \in \{B, \perp_0\}$ and $z \in \Gamma \setminus \{\perp_0\}$.

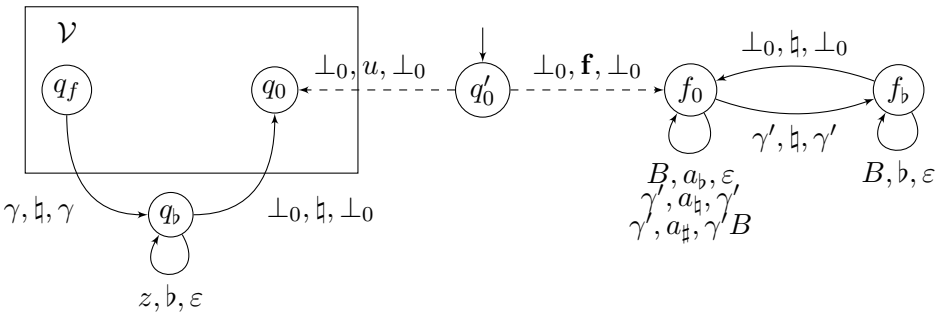


Figure 5.6: A POpVPA for the EXPTIME-hardness of FF-diagnosability.

The observed sequences corresponding to correct runs in $\langle \mathcal{V}', \Sigma_o, \mathcal{P} \rangle$ are either of the form $w_1 \natural \flat^{k_1} \natural w_2 \dots \natural \flat^{k_{n-1}} \natural w_n$ or of the form $w_1 \natural \flat^{k_1} \natural w_2 \dots \natural w_{n-1} \natural \flat^m$. In these decompositions, w_i, for $i < n$, is a sequence corresponding to a run of $\mathcal{V}$ starting in q_0 and ending in some accepting state $q_f \in Q_F$, k_i is the number of elements in the stack after reading w_i in $\mathcal{V}$ and also in $\mathcal{V}'$ (apart from the bottom stack symbol $\perp_0$), w_n is the sequence associated with a run of $\mathcal{V}$ starting in q_0, and m is at most the number of elements in the stack after reading w_{n-1} in $\mathcal{V}$. Note that k_i only depends on w_i, and does not depend on the exact run over w_i, since $\mathcal{V}$ is a VPA.
The observed sequences corresponding to faulty runs in $\langle \mathcal{V}', \Sigma_o, \mathcal{P} \rangle$ are less constrained. They are of one of the two forms presented above, however the words w_i for $i \leq n$ can be any word of Σ^*.

Let us show that $\mathcal{V}$ is not universal if and only if $\langle \mathcal{V}', \Sigma_o, \mathcal{P} \rangle$ is FF-diagnosable.
First assume that $\mathcal{V}$ is not universal. Then there exists a word $w \in \Sigma^*$ such that no run of $\mathcal{V}$ reading w ends in an accepting state q_f. However, the observed sequence of

any faulty run almost-surely contains the factor $\natural w \natural$. Indeed, faulty runs almost surely visit infinitely often the configuration $(f_\flat, \bot_0)$, and from there, the probability λ to read $\natural w \natural$ is positive. Let ρ be an infinite faulty run. Its observed sequence is of the form $\mathcal{P}(\rho) = w_1 \natural \flat^{k_1} \natural w_2 \natural \flat^{k_2} \natural w_3 \dots$ with $k_i \leq |w_i|$ for every i. If there exists $i \leq n$ such that $w_i = w$ then ρ is surely faulty, since it has no corresponding correct run. The latter statement can be refined. For $n \geq |w|$, if, for every $i \leq n$, $|w_i| \leq n$ and there exists $i \leq n$ such that $w_i = w$ then $\rho_{\downarrow 2n^2+n}$ is surely faulty. Indeed, $|w_i \natural \flat^{k_i}| \leq 2n+1$, w occurs at the latest for $i = n$, and once it occurs the prefix is surely faulty. Let us therefore consider faulty runs that do not satisfy this property. We let $\mathsf{Avoid}_n = \{\rho \in \mathsf{F} \mid \mathcal{P}(\rho) = w_1 \natural \flat^{k_1} \natural w_2 \natural \flat^{k_2} \natural w_3 \cdots \wedge (\forall i \leq n \ w_i \neq w \vee \exists i \leq n \ |w_i| > n)\}$. By construction, $\mathsf{FAmb}_{2n^2+n} \subseteq \mathsf{Avoid}_n$. Moreover, using standard union-sum inequalities, $\mathbb{P}(\mathsf{Avoid}_n) \leq (1-\lambda)^n + \frac{n}{2^n}$ (recall that λ is the probability to read $\natural w \natural$ from $(f_0, \bot_0)$). Thus $\lim_{n\to\infty} \mathbb{P}(\mathsf{Avoid}_n) = 0$ and hence $\lim_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = 0$ so that $\langle \mathcal{V}', \Sigma_o, \mathcal{P} \rangle$ is FF-diagnosable.

Assume now that $\mathcal{V}$ is universal. Let ρ be an infinite surely faulty run of $\langle \mathcal{V}', \Sigma_o, \mathcal{P} \rangle$. We write ρ' for the greatest ambiguous prefix of ρ and $a \in \Sigma_o \cup \{\natural, \flat\}$ such that $\rho' a$ is again a prefix of ρ. Observe that a cannot be $\flat$ since the number of $\flat$'s between two $\natural$'s, whether on the left or right-hand-side of $\mathcal{V}'$, is entirely determined by the word of Σ_o^* read before the first $\natural$. For the same reason, if $a = \natural$, $\mathcal{P}(\rho')$ ends with a word of Σ_o^* (*i.e.* the number of $\natural$'s in $\mathcal{P}(\rho')$ is even). Let w be the greatest suffix of $\mathcal{P}(\rho')$ contained in Σ_o^*. If $a = \natural$, we deduce that there is no run starting in q_0 with observed sequence w and ending in an accepting state of $\mathcal{V}$. Therefore, $\mathcal{V}$ is not universal. Similarly, if $a \in \Sigma_o$, then there is no run starting in q_0 and with observed sequence wa. In that case also, $\mathcal{V}$ is not universal. We hence conclude that there is no infinite surely faulty run in $\langle \mathcal{V}', \Sigma_o, \mathcal{P} \rangle$. As the probability to generate faulty runs is positive, this implies that $\langle \mathcal{V}', \Sigma_o, \mathcal{P} \rangle$ is not IF-diagnosable. Now, IF-diagnosability is equivalent to FF-diagnosability for finitely-branching POpLTS (see Theorem 3.1), and so $\langle \mathcal{V}', \Sigma_o, \mathcal{P} \rangle$ is not FF-diagnosable.

Let us now argue for the $\mathsf{EXPTIME}$-hardness of FA-diagnosability and IA-diagnosability. The reduction is very similar to the previous one: it only requires an additional state in the correct component that ensures that almost surely any correct run will be identified as being correct. Therefore the problem may only come from the faulty runs which are dealt with exactly as above. From the VPA $\mathcal{V} = (Q, \Sigma, \Gamma, \delta)$ and pVPA $\mathcal{V}' = (Q', \Sigma', \Gamma', \delta')$ defined above, we construct a pVPA $\mathcal{V}'' = (Q'', \Sigma'', \Gamma'', \delta'', \mathbf{P}'')$ such that

- $Q'' = Q' \cup \{q_c\}$ and q_0' is the initial state;
- $\Sigma'' = \Sigma \cup \{\mathbf{f}, u, \sharp, \alpha\}$;
- $\Gamma'' = \Gamma$;
- $\delta'' = \delta' \cup \ \{(q, \alpha, \gamma, q_c) \mid \gamma \in \Gamma \cup \{\bot_0\}, q \in Q \cup \ \{q_c\}\}$;
- $\mathbf{P}''$ assigns arbitrary positive probabilities to transitions in δ''.

We further consider the POpVPA $\langle \mathcal{V}'', \Sigma_o, \mathcal{P} \rangle$ with $\Sigma_o = \Sigma'' \setminus \{\mathbf{f}, u\}$, and the masking function satisfies $\mathcal{P}(\mathbf{f}) = \mathcal{P}(u) = \varepsilon$ and $\mathcal{P}(x) = x$ for any other event x. The construction is illustrated in Figure 5.7, where we use the shortcuts: $a_\flat \in \Sigma_\flat$, $a_\natural \in \Sigma_\natural$, $a_\sharp \in \Sigma_\sharp$, $\gamma \in \Gamma$, $\gamma' \in \{B, \bot_0\}$ and $z \in \Gamma \setminus \{\bot_0\}$.

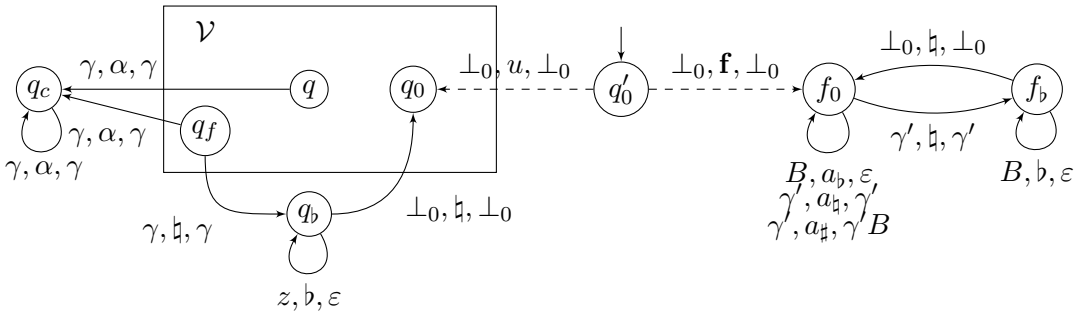


Figure 5.7: A POpVPA for EXPTIME-hardness of FA-diagnosability and IA-diagnosability.

$\mathcal{V}''$ is a slight modification of $\mathcal{V}'$: from any state of $\mathcal{V}$ (accepting or not), reading the new letter α leads to the sink state q_c. As a consequence, for any correct run of $\langle \mathcal{V}'', \Sigma_o, \mathcal{P} \rangle$, there is a positive probability at each step to perform event α and become surely correct. This implies $\lim_{n\to\infty} \mathbb{P}(\mathsf{CAmb}_n)_{n\in\mathbb{N}} = 0$. Observe that the above proof for $\mathcal{V}'$ also applies to $\mathcal{V}''$: $\mathcal{V}$ is not universal if and only if $\langle \mathcal{V}'', \Sigma_o, \mathcal{P} \rangle$ is FF-diagnosable. Now, since $\lim_{n\to\infty} \mathbb{P}(\mathsf{CAmb}_n)_{n\in\mathbb{N}} = 0$, FF-diagnosability, FA-diagnosability and IA-diagnosability coincide for $\langle \mathcal{V}'', \Sigma_o, \mathcal{P} \rangle$. We conclude that $\mathcal{V}$ is not universal if and only if $\langle \mathcal{V}'', \Sigma_o, \mathcal{P} \rangle$ is diagnosable (for any notion of diagnosability). □

While diagnosability was undecidable for POpPDA, the situation is more complex with POpVPA. Some notions become decidable, although we may not have the exact complexity, while for others decidability remains open. The method used here requires a characterisation that can be translated in pLTL. This is not possible for FA-diagnosability or the approximate notions of diagnosability which cannot be expressed in pathL. Recall that pathL is more expressive than LTL according to Proposition 3.6, page 79.

3 Diagnosability of infinite pLTS represented by stochastic Petri nets

In this section we study infinite-state pLTS generated by stochastic Petri nets (SPN). This model is incomparable to pPDA and therefore generates different kinds of pLTS. In Subsection 3.1 we formally define SPN and the infinite-state pLTS generated by an SPN. We then show in Subsection 3.2 that, as for pPDA, diagnosability is undecidable

in SPN, and that a restriction similar to what pVPA are to pPDA is not enough to regain decidability.

3.1 Stochastic Petri nets

A Petri net contains places and transitions. In each place there are tokens and a transition consumes tokens from input places and produces tokens in output places. From a manufacturing point of view, these tokens can be seen as different items that are received, assembled with other items, processed and the final product can be exported. Due to the locality of transitions, Petri nets are appropriate for modelling concurrent systems. The infinite behaviour comes from the potentially unbounded number of tokens inside the net. The sets of infinite-state pLTS that can be generated by pushdown systems and Petri nets are incomparable.

Definition 5.9. *A* Petri net *(PN) is a structure $N = (P, M_0, T, Pre, Post)$, where P is a set of m places; M_0 is the initial* marking, i.e. *a vector $M : P \to \mathbb{N}$ that assigns to each place of a PN a non-negative integer number of tokens; T is a set of n transitions; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the pre– and post– incidence functions that specify the arcs. We also define $C = Post - Pre$ as the incidence matrix of the net.*

For $\mathsf{Mat} \in \{Pre, Post, C\}$ and $t \in T$, we write $\mathsf{Mat}(\cdot\,, t)$ for the column vector which, for every $i \in \mathbb{N}$, contains at the row i the value $\mathsf{Mat}(i, t)$. A transition t is enabled from M iff $M \geq Pre(\cdot\,, t)$ and may fire yielding the marking $M' = M + C(\cdot\,, t)$. One writes $M\ [\sigma\rangle$ to denote that the sequence of transitions $\sigma = t_{j_1} \cdots t_{j_k}$ is enabled from M, and $M\ [\sigma\rangle\ M'$ to denote that the firing of σ yields M'. One writes $t \in \sigma$ to denote that a transition t is contained in σ. The length of the sequence σ (denoted $|\sigma|$) is the number of transitions in the sequence, here k.

Example 5.11. *Consider the PN of Figure 5.8. The initial marking is $M_0 = [2, 0, 0, 0, 0]$. Two tokens are needed to fire t_2, one in place p_1 and one in place p_2. In order to take the manufacturing analogy again, this means two items must be assembled here. In p_1 two items are already here, ready for assembling, however p_2 is empty. Firing t_1 delivers one item to p_2, enabling the transition t_2. Once t_2 was fired, one token in p_1 and one token in p_2 are consumed and one token (the assembled product) is produced in p_3. There two transitions can be fired. For example, the token can be consumed by t_4 producing a new token in p_4 which enables t_6. This last transition consumes a token without creating any new one, so it could correspond to the finished product being sent, and thus removed from consideration by this system. A sequence of transitions corresponding to the arrival of new products, their processing and removing from the system is $\sigma = t_0 t_1 t_2 t_4 t_6$. Firing this sequence uses exactly the tokens that are created inside it. Therefore it can be repeated: σ^k is enabled from M_0 for all $k \in \mathbb{N}$.*

The set of all sequences that are enabled at the initial marking M_0 is denoted $L(N)$, i.e., $L(N) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}$. A marking M is *reachable* in N iff there exists a firing sequence σ such that $M_0\ [\sigma\rangle\ M$. The set of all markings reachable from M_0 defines the *reachability set* of N and is denoted $R(N)$. Given $k \in \mathbb{N}$, a place p of a PN N is

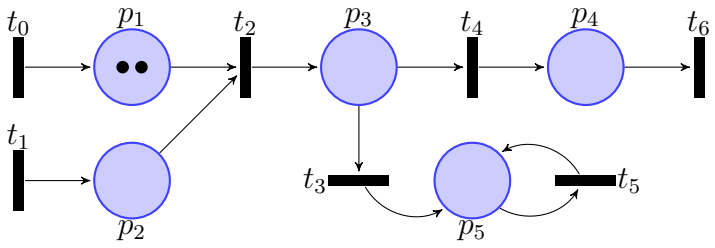


Figure 5.8: A Petri net. Circles are places and rectangles are transitions. In the initial marking, p_1 has two tokens represented by the two black dots.

k-bounded if for all $M \in R(N)$, $M(p) \leq k$. It is bounded if there exists $k \in \mathbb{N}$ such that p is k-bounded. A PN is bounded (resp. k-bounded) iff all of its places are bounded (resp. k-bounded).

Example 5.12. *Consider again the PN of Figure 5.8. Firing t_1 k times in M_0 leads to the marking $M_1 = [2, k, 0, 0, 0]$. Therefore the place p_2 is not bounded.*

Probabilities are added to a PN by adding a fire rate to every transition in the following way.

Definition 5.10. *A* Stochastic Petri Net *(SPN) is a pair $\mathcal{N} = (N, \mu)$ where N is a PN and for all $t \in T, \mu(t) \in \mathbb{R}^+$ is the rate of firing of transition t.*

The usual interpretation of rates is that, in a given marking, a delay is computed for every enabled transition t with an exponential probability distribution function of parameter $\mu(t)$, *i.e.* the probability distribution function for the delay of transition t is $f_t : x \in \mathbb{R}^+ \mapsto \mu(t)e^{-\mu(t)x}$. Multiple time semantics [HM09] can be chosen in an SPN to decide how these delays are used to determine which transition is fired. For instance, one could use (a) a *single server policy*: each transition can only be fired once by a given marking, (b) a *race policy*: the transition whose firing delay elapses first is assumed to be the one that will fire next and (c) a *resampling memory policy*: at the entrance in a marking, the remaining delays associated with all transitions are forgotten. Observe that as we use an exponential probability distribution, whether the delays are forgotten or not does not modify the probabilistic semantic. Using these choices, one could then define the semantics of the SPN as a continuous time Markov chain. However, as we only focus on discrete-time semantics here, we simplify the definition of the probabilistic behaviours of the SPN. We remove the time consideration from the semantics, and only keep the discrete time Markov chain induced by the continuous time Markov chain. This semantics keeps enough information to answer questions expressed for example by pLTL or pCTL formulae, but cannot address time-related issues such as mean reaction time.

Using the simplified interpretation of rates, as for pLTS, a probability measure can be defined on the sequences of transitions of a PN. Given a sequence $\sigma \in T^*$, we write $C(\sigma)$ for the set of infinite sequences prefixed by σ, $C(\sigma) = \{\sigma' \in T^\omega \mid \exists \sigma'' \in$

$T^\omega : \sigma' = \sigma\sigma''\}$. The set of infinite sequences is the support of a probability measure defined by Caratheodory's extension theorem from the probabilities of the cylinders: the probability of the cylinder starting by the empty sequence ε is equal to 1 and, for σt a sequence, the probability of $C(\sigma t)$ in M_0, written $\mathbb{P}(\sigma t)$, satisfies

$$\mathbb{P}(\sigma t) = \mathbb{P}(\sigma) \times \frac{\mu(t)}{\sum_{t' \in T, M_0[\sigma t'\rangle} \mu(t')}$$

As for pPDA, we enhance SPN with a mask function.

Definition 5.11. *A* partially observable SPN *(POSPN) is a tuple $\langle \mathcal{N}, \Sigma_o, \mathcal{P}\rangle$ consisting of an SPN $\mathcal{N}$ equipped with a mapping $\mathcal{P} : T \to \Sigma_o \cup \{\varepsilon\}$ where Σ_o is the set of observations.*

From now on, we assume that there does not exist a marking M reachable from M_0 and an infinite sequence $\sigma \in T^\omega$ such that $\mathcal{P}(\sigma) = \varepsilon$ and $M[\sigma\rangle$. This assumption corresponds to the assumption of convergence that was made for pLTS. The observed sequence w of observations associated with the sequence σ is $w = \mathcal{P}(\sigma)$. Note that the length of a sequence σ is always greater than or equal to the length of the corresponding observed sequence w (denoted $|w|$). Given a word $w \in L^*$, we write $\mathbb{P}(w) = \sum_{\sigma \in P_e^{-1}(w)} \mathbb{P}(\sigma)$. Thanks to our earlier assumption, this sum is finite.

Example 5.13. *Consider again the PN N of Figure 5.8. We define the POSPN $\langle (N, \mu), \{a, b, c\}, \mathcal{P}\rangle$ such that for all $t \in T, \mu(t) = 1$ and $\mathcal{P}(t_0) = \mathcal{P}(t_1) = b, \mathcal{P}(t_2) = a, \mathcal{P}(t_3) = \mathcal{P}(t_4) = \varepsilon$ and $\mathcal{P}(t_5) = \mathcal{P}(t_6) = c$. The observed sequence bac corresponds to the sequences $t_1t_2t_3t_5$ and $t_1t_2t_4t_6$, each of which has a probability $\frac{1}{72}$. Therefore $\mathbb{P}(bac) = \frac{1}{36}$.*

The (potentially infinite) pLTS associated with a POSPN is based on the reachability graph of the PN: every state corresponds to a reachable marking.

Definition 5.12. *A POSPN $\langle \mathcal{N}, \Sigma_o, \mathcal{P}\rangle$ defines a pLTS $\mathcal{A}_\mathcal{V} = (Q_\mathcal{N}, M_0, T, T_\mathcal{N}, \mathbf{P}_\mathcal{N})$ where:*

- *$Q_\mathcal{N} = R(M_0)$;*
- *$T_spn = \{(M, t, M') \mid M[t\rangle M'\}$;*
- *For every $(M, t, M') \in T_\mathcal{N}$, $\mathbf{P}_\mathcal{N}[(M, t, M')] = \frac{\mu(t)}{\sum_{t' \in T, M[t'\rangle} \mu(t')}$.*

This pLTS is infinite when the reachability set is infinite. This happen iff the PN is not bounded. If the PN is k-bounded, for $k \in \mathbb{N}$, then the size of the generated pLTS is exponential in the size of the PN and in k. A POSPN is diagnosable according to a notion of diagnosability if the pLTS it generates is diagnosable.

In order to mirror the POpVPA restriction of POpPDA, we introduce the notion of visible POSPN.

Definition 5.13. *A* visible POSPN *(VSPN) is a POSPN such that*

- *an unobservable transition does not modify the number of tokens in the system;*
- *for every pair of two transitions t_1 and t_2 with $\mathcal{P}(t_1) = \mathcal{P}(t_2)$ $Post(\cdot, t_1)1_v - Pre(\cdot, t_1)1_v = Post(\cdot, t_2)1_v - Pre(\cdot, t_2)1_v$ where 1_v is the vector with 1 in every position.*

This second condition means that the number of tokens is modified similarly by t_1 and t_2. An observer of a VSPN thus knows at all time how many tokens are present in the system.

3.2 Undecidability of diagnosability for stochastic Petri nets

The exact diagnosability problems for k-bounded Petri nets are decidable as the generated pLTS is exponential and the exact diagnosability problems for finite pLTS are decidable. Moreover deciding if a Petri net is bounded is also decidable [Rac78]. For unbounded Petri nets however, while non-stochastic variants of diagnosability are known to be decidable on Petri nets, this is not the case for the stochastic notions of exact diagnosability. In order to show the undecidability, we reduce the problem of the language inclusion for Petri nets, namely: given two PN N^1 and N^2, an observation alphabet Σ_o and a mask function $\mathcal{P}$ does $\mathcal{P}(L(N^1)) \subseteq \mathcal{P}(L(N^2))$ hold? This problem is known to be undecidable(see the survey [EN94]).

Theorem 5.5. *The* FF*-,* IA*- and* FA*-diagnosability problems of POSPN are undecidable.*

Given two PN N^1 and N^2, we build an SPN where the initial transition (which can be faulty), produces tokens in one among two components. This component corresponds to an enhanced copy of one of the two given PN. Then, a sequence of this PN is triggered. At any moment during this sequence, a transition starting a reset operation can be taken. This reset operation removes all the tokens from the PN then produces the tokens corresponding to the initial marking so that a new sequence can be read by the PN. The goal is that the fault is detected iff an observed sequence that can only be triggered from N^1 is observed in between two resets operation.

The difficulty of this reduction lies in the reset operation as one cannot test directly whether the places of the PN were correctly emptied. This information however, can be encoded in the observation. Let us now describe what happens in a reset. The reset starts and ends by an observable $\sharp$ and, in between, produces a certain number of $\flat$. Each of these $\flat$ removes a token that was left inside the PN, so that at any moment, the observer knows precisely the number of tokens within the system. If there is still at least one token in the system when the second $\sharp$ occurs, a gadget is used to allow the system to trigger any observed sequence so that no information is given to the observer. In other words, the observed sequence in between two reset operations give an information on the system iff the previous reset had correctly emptied the PN.

Proof. Let $N^1 = (P^1, M_0^1, T^1, Pre^1, Post^1)$ and $N^2 = (P^2, M_0^2, T^2, Pre^2, Post^2)$ be two PN, with the mask function $\mathcal{P}$ and the observation alphabet Σ_o.

Without loss of generality, we assume that the initial marking M_0^i has a single token in a place p_0^i for $i = 1, 2$, that every transition is observable and that there exists an integer $k \in \mathbb{N}$ such that the number of tokens in the system is equal to k times the length of the sequence plus 1. This last assumption could be ensured by choosing k as the maximum number of tokens added by a transition (this means number of tokens created minus number of tokens consumed) and adding an additional place where the unnecessary tokens are put (*i.e.* if a transition adds k' tokens with $k' < k$, then this new place receives $k - k'$ tokens).

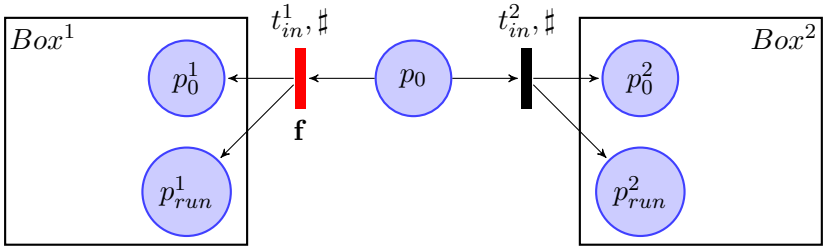


Figure 5.9: Reduction from language inclusion. The Figure 5.10 represents the content of the box Box^1, it is similar for Box^2. Transitions are labelled by their name and their observation.

We build the POSPN $(P, M_0, T, Pre, Post, \mu, \Sigma_o \cup \{\sharp, \flat\}, \mathcal{P}')$ (represented in Figure 5.9) where:

- $P = P^1 \cup P^2 \cup \{p_0\} \cup \{p_{emp}^i, p_{run}^i, p_{err}^i \mid i = 1, 2\}$;
- $T = T^1 \cup T^2 \cup \{t_{in}^i, t_\sharp^i, t_{rese}^i, t_{resn}^i \mid i = 1, 2\} \cup \{t_a^i \mid a \in \Sigma, i = 1, 2\} \cup \{t_{emp}^p, t_{err}^p \mid p \in P^1 \cup P^2\}$;
- for $i \in \{1, 2\}, p \in P^i, t \in T^i, Pre(p, t) = Pre^i(p, t)$ and $Pre(p_{run}^i, t) = 1$, $Pre(p_0, t_{in}^i) = 1$, $Pre(p_{emp}^i, t_\sharp^i) = 1$, $Pre(p_{run}^i, t_{resn}^i) = 1$, $Pre(p_{err}^i, t_{rese}^i) = 1$, for $a \in \Sigma, Pre(p_{err}^i, t_a^i) = 1$, $Pre(p, t_{emp}^p) = Pre(p_{emp}^i, t_{emp}^p) = 1$, $Pre(p, t_{err}^p) = Pre(p_{emp}^i, t_{err}^p) = 1$. When undefined, $Pre(p, t) = 0$;
- for $i \in \{1, 2\}, p \in P^i, t \in T^i, Post(p, t) = Post^i(p, t)$ and $Post(p_{run}^i, t) = 1$, $Post(p_0^i, t_{in}^i) = Post(p_{run}^i, t_{in}^i) = 1$, $Post(p_{run}^i, t_\sharp^i) = Post(p_0^i, t_\sharp^i) = 1$, for $a \in \Sigma, Post(p_{err}^i, t_a^i) = 1, Post(p_0^i, t_a^i) = k$, $Post(p_{emp}^i, t_{resn}^i) = Post(p_{emp}^i, t_{rese}^i) = 1$, $Post(p_{emp}^i, t_{emp}^p) = 1$, $Post(p_0^i, t_{err}^p) = 2, Post(p_{err}^i, t_{err}^p) = 1$. When undefined, $Post(p, t) = 0$;
- $\mathcal{P}'$ extends $\mathcal{P}$ on N by, for $p \in P^1 \cup P^2, i \in \{1, 2\}, a \in \Sigma$, $\mathcal{P}'(t_{in}^i) = \sharp$, $\mathcal{P}'(t_\sharp^i) = \mathcal{P}'(t_{err}^p) = \mathcal{P}'(t_{rese}^i) = \mathcal{P}'(t_{resn}^i) = \sharp$, $\mathcal{P}'(t_a^i) = a$, $\mathcal{P}'(t_{emp}^p) = \flat$;
- for $i \in \{1, 2\}$, $p \in P_i$, $\mu(t_{rese}^1) = \mu(t_{resn}^1) = \mu(t_{emp}^p) = 2k(|\Sigma| + |T_1|)$ (assuming $|T_1| \geq 1$) and for every other transition t $\mu(t) = 1$.

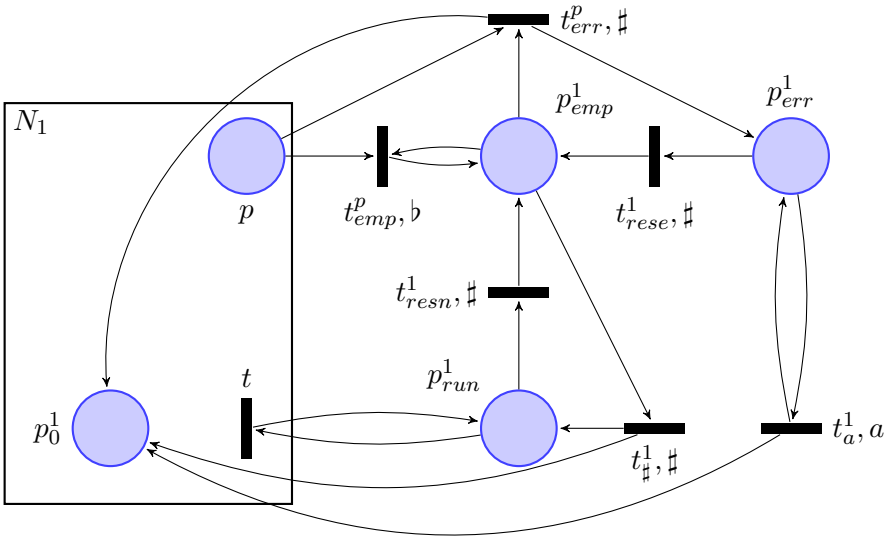


Figure 5.10: Content of the box Box^1.

Moreover, $t^1_{in} = \mathbf{f}$ is the fault transition.

We show that the system is FF-diagnosable iff $\mathcal{P}(L(N^1)) \not\subseteq \mathcal{P}(L(N^2))$.

First note that the set of observed sequences associated with the infinite sequences starting by the transition t^i_{in}, denoted L^i, contains exactly the words of the form $\sharp w_1 \sharp \flat^{n_1} \sharp \dots w_k \sharp \flat^{n_k} \sharp \dots$ where for all $1 \leq j \leq k$, (1) $w_j \in \Sigma_o^*$, (2) $\sum_{m=1}^{j} k|w_m| + 1 \geq \sum_{m=1}^{j} n_m$ and (3) $\sum_{m=1}^{j-1} k|w_m| + 1 = \sum_{m=1}^{j-1} n_m$ implies $w_j \in \mathcal{P}(L(N^i))$.

Suppose that $\mathcal{P}(L(N^1)) \subseteq \mathcal{P}(L(N^2))$. Let σ be an infinite faulty sequence. As σ is faulty, it initially fired t^1_{in}, thus $\mathcal{P}(\sigma) \in L^1$. Thanks to the above remark on the languages L_i, and as $\mathcal{P}(L(N^1)) \subseteq \mathcal{P}(L(N^2))$, $\mathcal{P}(\sigma) \in L^2$, therefore there exists a sequence σ' starting by the transition t^2_{in} with same observation as σ. Moreover this transition is not faulty as it did not fire t^1_{in} initially and cannot fire it after the first transition. Therefore $\mathcal{P}(\sigma)$ is not surely faulty. As this is true for every faulty sequence, the system is not IF- diagnosable. The pLTS generated by this POSVN being finitely branching, according to Theorem 3.1, this implies that the POSPN is not FF-diagnosable.

Suppose now that $\mathcal{P}(L(N^1)) \not\subseteq \mathcal{P}(L(N^2))$. There thus exists a word w such that $w \in \mathcal{P}(L(N^1)) \setminus \mathcal{P}(L(N^2))$. The observed sequences of L_1 such that there exists $i \in \mathbb{N}$ with $\sum_{m=1}^{i-1} k|w_m| + 1 = \sum_{m=1}^{i-1} n_m$ and $w_i = w$ are surely faulty as they do not belong to L^2. We denote SL_1 the set of these observed sequences. Let us show now that with probability 1 an infinite faulty sequence belongs to SL_1.

While a token is in p^1_{err} or p^1_{run}, every transition taken with observation other than $\sharp$ produces k tokens in the copy of N_1. Moreover, there are at most $|\Sigma| + |T_1|$ such transitions, each with rates 1. As the transition triggering $\sharp$ has rate $2k(|\Sigma| + |T_1|)$, the expectation of the number of tokens produced before a $\sharp$ is below $1 + 2k$. During a reset

operation, as for all places $p \in P_1$, $\mu(t^p_{emp}) = 2k(|\Sigma| + |T_1|)$ and the two transitions producing $\sharp$ have rates 1, the expectation of the number of token removed from the copy of N_1, assuming there are enough tokens within the system, is greater than $1 + k(|\Sigma|+|T_1|)$[1]. Thus, with probability 1, a faulty sequence will infinitely often remove all the tokens from P^1. Therefore with probability 1, the observation of an infinite faulty sequence will be of the form $\sharp w_1 \sharp \flat^{n_1} \sharp \dots w_k \sharp \flat^{n_k} \sharp \cdots \in L^1$ with infinitely many $i \in \mathbb{N}$ such that $\sum_{m=1}^{i-1} k|w_m| + 1 = \sum_{m=1}^{i-1} n_m$. There is a probability $p > 0$ that for any such i, $w_i = w$ as $w \in \mathcal{P}(L(N^1))$. Therefore with probability 1, there exists $i \in \mathbb{N}$ such that $w_i = w$. Hence with an infinite faulty sequence almost surely belongs to SL_1. This implies that the POSPN is IF-diagnosable and thus FF-diagnosable.

In order to reduce the problem to FA-diagnosability and IA-diagnosability, we proceed similarly to the proof of Theorem 5.4: we add another place p_c and a transition t_c that takes a token from p^2_{run} and puts it in p_c. This transition has firing rate 1 and observation $\natural$. This is thus the only transition with this observation. As a consequence, taking this transition ensures the run is surely correct and remains that way. As a run entering Box^2 almost surely infinitely often contains a token in p^2_{run}, every run almost surely either becomes faulty or surely correct. Therefore $\lim_{n\to\infty} \mathsf{CAmb}_n = 0$. This implies that in this POSVN, FF-diagnosability is equivalent to FA-diagnosability and IA-diagnosability. The rest of the proof above then applies. □

An interesting feature of this proof is that the number of tokens in the POSPN used in the reduction can be deduced from the observation at all time. Therefore it is a VSPN. This gives the following result.

Corollary 5.1. *The* FF-*,* IA- *and* FA-*diagnosability problems of VSPN is undecidable.*

Thus, a restriction similar to what allowed us to regain decidability in POpPDA is not enough for POSPN.

4 Conclusion

The study of diagnosability for infinite-state pLTS depends heavily on the model used to finitely represent such a pLTS. Choosing a model that is too powerful leads quickly to undecidability. This has been shown with the undecidability proofs established for restricted classes of POpPDA and POSPN. These proof contains important differences. For instance, while undecidability is proven for every notion of stochastic (and in fact even non-stochastic) diagnosability in POpPDA, it is only proven for the exact notions of diagnosability in POSPN. Moreover, it is known that non-stochastic diagnosability is decidable in PN. In this sense, PN is a model for which there is still hope to get decidability results. We did not use the notion of coverability graph here, which gives a finite over-approximation of the reachability graph. Maybe an analysis of its language coupled with a study of the pathological behaviours (due to the over-approximation) may help in solving AFF-diagnosability. Moreover, the restriction that was used for

[1]This value is obtained by analysing the case where only one place contains tokens.

POSPN was chosen to mimic the one used on POpPDA, but is not necessarily the most suited to the model.

Even if the POpPDA model has the strongest undecidability results, the appropriate restriction allows to regain decidability. For POpVPA, we could use model-checking methods to verify pLTL formula equivalent to the logical characterisation of Section 3 of Chapter 3. This only gave decidability of the notions for which a logical characterisation was known. Many questions are still left open partially as a consequence.

First, it would be interesting to find ways to close the complexity gap between our upper and lower bound for the decidable diagnosability notions. The complexity of the current decision procedure comes from an exponential determinisation and the use of a PSPACE model-checking result. As we are interested in specific simple formula, there may be a way to verify them in PTIME instead. The exponential of the determinisation seems harder to remove.

Second, we would want to determine the decidability status of FA-diagnosability. If it is undecidable, it would confirm the difference in complexity with the other notions of exact diagnosability that the logical characterisations showed. However, this difference was shown for infinite-state pLTS in general, not for pLTS generated by POpVPA. It is possible that, as for finite systems, FA-diagosability could be decided for POpVPA with the same complexity as the other exact diagnosability notions.

Finally, one may be interested in considering the case of the approximate diagnosability notions. The method used here cannot be applied. Moreover, it is unclear now what the POpVPA restriction simplifies for approximate notions of diagnosability. Recall that in the finite case, no determinisation were used to solve these notions, allowing for a PTIME algorithm. We conjecture that this notion remains undecidable.

Part III

Controlling Information in Active Systems

Chapter 6

Control of the degradation in probabilistic systems

Embedded systems are often equipped with one (or more) controller(s) that can modify the behaviour of the system in reaction to the environment. Controllers can, for example, be used in order to maintain some vital functionalities of the system when facing a failure of a component. As controllers need to detect failures to react efficiently, it is tempting to add to controllers a diagnosis task. In other words, the system will contain some choices that can be made and which will alter the behaviour of the system while satisfying its specification. Controllers will then resolve these choices in order to render the system diagnosable. Controllers can be formalised in multiple different ways. For example, controllers could be within the system and thus have have full knowledge of the behaviour of the system or they could rely on partial observation similarly to diagnosers. Since the goal is for controllers to deduce the existence of a fault, we cannot assume they know exactly the state of the system, and thus it must rely on partial observation. Formally, some of the observable events are controllable and, considering its current observation, the controller chooses which subset of events the system can trigger. A system is then said to be *actively diagnosable* if there exists a controller ensuring its diagnosability. In [SLT98], the authors showed that the active diagnosability problem is decidable in doubly exponential time for non-probabilistic systems. Then in [HHMS17], the authors designed a single exponential time algorithm and proved this complexity to be optimal. In the probabilistic case, the controllable system can be represented by a weighted transition system in the active case. This weighted transition system, coupled with a controller, produces a pLTS that can have infinitely many states (depending on the memory required by the controller). Thus, unsurprisingly, the active probabilistic diagnosability is more complicated than the corresponding passive problem: exact diagnosability is PSPACE-complete in pLTS (see Chapter 4) while it is EXPTIME-complete (see [BFH$^+$14]) for controllable weighted LTS (a controllable variant of pLTS).

However the choices performed by the controller ensuring active diagnosis may have a pernicious effect: to detect faults, controllers sometimes could favour the occurrence of these faults! Forcing a fault in the system easily ensures diagnosability but contradicts

the initial goal of trying to maintain important functionalities of the system. Additional requirements can thus be made to controllers in order to manage the degradation of the system. Thus, a controller ensures *safe active diagnosability* if the controlled system is diagnosable and there is a positive probability that an infinite run is correct. In other words, the controller is allowed to increase the probability of a fault in order to ensure diagnosability, however it must maintain a positive probability of correct behaviours. A quantitative version of this requirement fixes a threshold ε to the probability of correct runs that the controller must achieve. Unfortunately, safe active probabilistic diagnosability is undecidable [BFH$^+$14]. However, when limited to finite-memory controllers, the problem becomes decidable in NEXPTIME [BFH$^+$14]. Safe active diagnosability may be too strong a requirement for some real systems. Indeed, systems age and whatever control is applied, their components will eventually fail. Thus, in many cases, the fault can be considered unavoidable by the system. As a consequence, some systems are designed to behave correctly for a long period of time at the end of which they will be replaced by a new system. Instead of trying to force runs to stay correct, a controller could try to slow the speed at which the system fails. This expresses a different kind of requirements for the degradation control of a system. We formalise the framework and these requirements in Section 1, establishing a few semantical results along the way. Then, in Section 2.1 we present the algorithmic results.

This chapter develops and extends some of the results from [BHL17b].

1 Degradation of a probabilistic system

In this section, we give formal definitions of the degradation of a system. These degradation notions have to be satisfied by the system simultaneously to diagnosability, ensuring that any fault is detected and the system does not produce faults too often or too quickly. As this combination depends on the notion of diagnosability chosen and our focus here is more on degradation, we only use FF-diagnosability (which is the simplest notion of exact diagnosability that we introduced in Chapter 2).

In terms of observation, we use in this chapter a partition between observable and unobservable events (see discussion of Section 1.3 of Chapter 2).

In Subsection 1.1, we give the definitions of degradation for pLTS. Then, in Subsection 1.2, we show how to add a form of control and state the problems we are interested in.

1.1 Degradation in passive systems

When protecting a system from degradation, we want that it has a sufficient probability not to trigger a fault, or at least, that if a fault has to occur, it can be postponed as much as possible. We study different notions of the degradation of a system: safety, fault freeness and resiliency.

A pLTS is *safe* [BFH$^+$14] if it guarantees a positive probability of infinite correct runs. A pLTS that is not safe is thus doomed to trigger a fault with probability 1. The probability to stay correct could however be arbitrarily low. So we can quantify the

notion in order to refine it: for $\varepsilon > 0$, a pLTS is ε-safe if this probability is greater or equal to ε.

Definition 6.1. *Let $\mathcal{A}$ be a pLTS, $\varepsilon > 0$. $\mathcal{A}$ is* ε-safe *if $\mathbb{P}(\mathsf{C}_\infty) \geq \varepsilon$. It is* safe *if $\mathbb{P}(\mathsf{C}_\infty) > 0$.*

As pointed out in the introduction, in some cases, safety is a too strong requirement. We formalise now two alternatives: fault freeness and resiliency. Fault freeness aims at quantifying the period of time during which the pLTS is correct. We introduce a discount factor $\gamma \leq 1$ on duration in order to vary the importance given to the length of the correct runs. When γ is chosen small, only the beginning of the runs matter. This focus on the short-term is useful for systems that are regularly replaced for example. A greater γ will on the opposite be chosen if one wants the system to be correctly performing for a longer time. The expectation of this discounted value is then compared to a threshold v.

Definition 6.2. *Let $\mathcal{A}$ be a pLTS, $0 < \gamma \leq 1$ and $v \in [0, \infty]$.*

- *$\mathcal{A}$ is* (γ, v)-fault free *if $\sum_{n\geq 1} \mathbb{P}(\mathsf{C}_n)\gamma^n \geq v$.*
- *$\mathcal{A}$ is* lasting fault free *if it is $(1, \infty)$-fault free.*

Clearly, for any fixed value of γ, the greater v is, the better the system. Remark also that for $\gamma < 1$, the sum $\sum_{n\geq 1} \mathbb{P}(\mathsf{C}_n)\gamma^n$ is finite and smaller than $\frac{1}{1-\gamma}$. For $\gamma = 1$, $\sum_{n\geq 1} \mathbb{P}(\mathsf{C}_n)\gamma^n$ is the mean observable length of the maximal correct signalling prefix of a random run, which can be infinite. This justifies the name *lasting fault free* when the expectation is infinite.

The notion of resiliency is an alternative measure of degradation based on a factor of degradation ratio per time unit $\alpha < 1$. A pLTS is *α-resilient* if the proportion of finite correct runs which stays correct on the next occurrence of an observable event is asymptotically greater than α. This requirement has two qualitative variants: strong resiliency (resp. weak resiliency) requires α-resiliency for every (resp. for at least one) $\alpha < 1$. In other words, a system is weakly resilient if asymptotically, the probability to be in a correct run of observable length n is greater than an exponential α^n. And a system is strongly resilient if this probability is asymptotically greater than all such exponential.

Definition 6.3 (Resilient pLTS)**.** *Let $\mathcal{A}$ be a pLTS.*

- *Let $0 < \alpha < 1$. $\mathcal{A}$ is* α-resilient *if $\limsup_{n\to\infty} \frac{\alpha^n}{\mathbb{P}(\mathsf{C}_n)} = 0$;*
- *$\mathcal{A}$ is* strongly resilient *if for all $0 < \alpha < 1$, $\mathcal{A}$ is α-resilient;*
- *$\mathcal{A}$ is* weakly resilient *if there exists $0 < \alpha < 1$ such that $\mathcal{A}$ is α-resilient.*

Example 6.1. *Let us consider the pLTS $\mathcal{A}$ of Figure 6.1. We give examples of the different notions of degradation by studying some choices of probabilities $(p_i)_{i\in\mathbb{N}}$.*

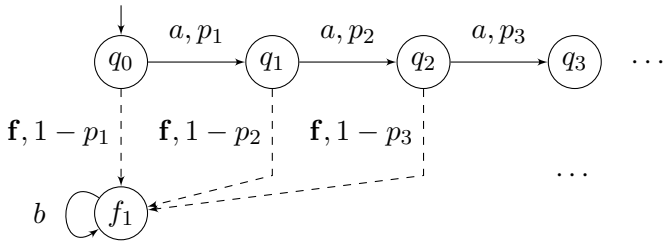


Figure 6.1: An example of infinite pLTS with parametric probabilities $(p_i)_{i\in\mathbb{N}}$.

$\mathcal{A}$ has a single correct run $\rho = q_0aq_1aq_2\ldots$, with observation a^ω while every faulty run contains an infinite number of 'b'. $\mathcal{A}$ is thus FF*-diagnosable. Moreover, the probability of ρ is $\prod_{n\geq 1} p_n$ and the probability of its prefix of length n is $r_n = \prod_{i\leq n} p_i$. Consequently, $\mathcal{A}$ is safe iff $\lim_{n\to\infty} r_n > 0$. This can be achieved by choosing $p_i = 1 - \frac{1}{2^i}$ for example.*

Also, by direct application of the definition, $\mathcal{A}$ is lasting fault free iff $\sum_{n\geq 1} r_n = \infty$. Let us consider different values of $(p_i)_{i\in\mathbb{N}}$.

- *Let $p_i = \frac{i}{i+1}$. Then $r_n = \frac{1}{n+1}$. Thus $\mathcal{A}$ is not safe but is lasting fault free. For every $\alpha < 1$, $\lim_{n\to\infty}(n+1)\alpha^n = 0$. Thus $\mathcal{A}$ is also strongly resilient.*
- *Let $p_i = \frac{i^2}{(i+1)^2}$. Then $r_n = \frac{1}{(n+1)^2}$. Thus $\mathcal{A}$ is neither safe nor lasting fault free. For every $\alpha < 1$, $\lim_{n\to\infty}(n+1)^2\alpha^n = 0$. Thus $\mathcal{A}$ is strongly resilient.*
- *We inductively define two sequences m_k and n_k by:*

  $$n_k = 2^{\sum_{j<k} m_j} \text{ (hence } n_0 = 1\text{) and } m_k = n_k + \sum_{j<k} m_j + n_j.$$

  We also define the intervals:

  - $I_k = [n_k + \sum_{j<k} m_j + n_j, \sum_{j\leq k} m_j + n_j[$;
  - $J_k = [\sum_{j\leq k} m_j + n_j, n_{k+1} + \sum_{i\leq k} m_j + n_j[$.

When $i \in I_k$, we choose $p_i = \frac{1}{2}$. When $i \in J_k$, we choose $p_i = 1$.

Observe that for all $n \in J_k$, $r_n = 2^{-\sum_{j\leq k} m_j}$. Consequently

$$\sum_{n\geq 1} r_n \geq \sum_{k\geq 0}\sum_{n\in J_k} r_n = \sum_{k\geq 0} 2^{\sum_{j\leq k} m_j} 2^{-\sum_{j\leq k} m_j} = \infty.$$

Thus $\mathcal{A}$ is lasting fault free.

Let $k \in \mathbb{N}$ and $n = \sum_{j\leq k} m_j + n_j$. Consequently, $r_n = 2^{-\sum_{j\leq k} m_j}$. Fix $\alpha = \frac{1}{\sqrt{2}}$.

$$\frac{\alpha^n}{r_n} = 2^{\sum_{j\leq k} m_j}(\sqrt{2})^{-\sum_{j\leq k} m_j + n_j} \geq 2^{m_k}(\sqrt{2})^{-2m_k} = 1.$$

Therefore $\mathcal{A}$ is not α-resilient.

The next theorem establishes the precise links between the qualitative versions of the three degradation notions for pLTS.

Theorem 6.1. *Let $\mathcal{A}$ be a pLTS.*

- *If $\mathcal{A}$ is safe then $\mathcal{A}$ is lasting fault free and strongly resilient;*
- *If $\mathcal{A}$ if finite then:*
  $\mathcal{A}$ is safe iff $\mathcal{A}$ is lasting fault free iff $\mathcal{A}$ is strongly resilient;
- *There exists a lasting fault free pLTS that is not strongly resilient;*
- *There exists a strongly resilient pLTS that is not lasting fault free.*

The first assertion is quickly obtained from the definitions and the last two come directly from the previous examples. The second one requires a bit more development. As the pLTS $\mathcal{A}$ is finite, one can use the notion of bottom strictly connected component used in Chapter 4 to characterise the diagnosability notions for finite pLTS. A notable difference is that, in Chapter 4, we had to consider the BSCC of an enriched pLTS. Here, we show that every notion of degradation is equivalent to the existence of a reachable correct BSCC of the pLTS $\mathcal{A}$.

Proof. Let $\mathcal{A}$ be a safe pLTS. There exists $\varepsilon > 0$ such that for all n, $\mathbb{P}(\mathsf{C}_n) \geq \varepsilon$. Thus, $\sum_{n\geq 1} \mathbb{P}(\mathsf{C}_n) \geq \sum_{n\geq 1} \varepsilon = \infty$. Moreover, for all $\alpha < 1$, $\lim_{n\to\infty} \frac{\alpha^n}{\mathbb{P}(\mathsf{C}_n)} \leq \lim_{n\to\infty} \frac{\alpha^n}{\varepsilon} = 0$. Thus $\mathcal{A}$ is both lasting fault free and strongly resilient.

Let $\mathcal{A}$ be a finite pLTS. Observe that every BSCC of $\mathcal{A}$ contains either only correct states or only faulty states. Accordingly we can speak of faulty BSCC or correct BSCC. As $\mathcal{A}$ is a finite pLTS, we know that almost surely an infinite run reaches a BSCC and that the mean time to reach a BSCC is finite (see *e.g.* [BK08]). Due to the first result, $\mathcal{A}$ is safe iff there exists a reachable correct BSCC.

Suppose that $\mathcal{A}$ is not safe.

• Every reachable BSCC are faulty which implies that the mean time to reach a faulty BSCC if finite. This mean time is an upper bound on the mean observable length of the maximal signalling prefix of a correct run. Thus $\mathcal{A}$ is not lasting fault free.

• We note $m = |Q|$. For all $q \in Q_c$, there exists ρ_q a run starting in q composed of an elementary run from q to a faulty BSCC followed by an elementary run (or circuit) in the BSCC of which only the last event is observable (by convergence). This run has an observable length smaller or equal to m. We note μ_q, the probability of that run and $\mu = \min_{q\in Q_c} \mu_q$. Consider a signalling run ρ of observable length n for an arbitrary n and ending in $q \in Q_c$. From the existence of ρ_q, $\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+m} \cap \mathsf{C} \mid \rho \preceq \rho'\}) \leq (1-\mu)\mathbb{P}(\rho)$. Thus $\mathbb{P}(\mathsf{C}_{n+m}) \leq (1-\mu)\mathbb{P}(\mathsf{C}_n)$. So, $\mathbb{P}(\mathsf{C}_n) \in O((1-\mu)^{\frac{n}{m}})$. Choosing $\alpha = (1-\mu)^{\frac{1}{m}}$, $\mathcal{A}$ is not α-resilient and thus not strongly resilient. □

1.2 Controlled systems

Extending the pLTS formalism in order to express control requires to fix at least two features of this formalism: the nature of the control and the distribution of probabilities of the controlled system. Intuitively, we want the control and the diagnosis to be realised by the same device: from its observations, it restricts the system in order to

diagnose it and limit its degradation. The control is thus done with partial observation. So we recall the *Controllable Labelled Transition System* (CLTS) from [BFH$^+$14]. In this model, in order to specify the control, a subset of observable events is considered controllable. The controller forbids a subset of controllable events depending on the sequence of observations it has received. Thus the controller cannot modify its choice between two observations. The transitions of the system are no more labelled by (rational) probabilities but by (integer) weights which measure their relative possibility of occurrence. Given a state and a set of forbidden controllable actions, the weights of the transitions exiting this state and labelled by uncontrollable or allowed controllable actions are normalised to obtain a probability distribution. If the controller does not introduce any deadlock, the controlled system is a live pLTS.

Definition 6.4. *A* Controllable Labelled Transition System *(CLTS) is a tuple* $\mathcal{C} = \langle Q, q_0, \Sigma, T\rangle$ *where:*

- Q *is a set of states with an initial state* $q_0 \in Q$*;*
- $\Sigma = \Sigma_o \uplus \Sigma_u$ *is a finite set of events partitioned into the set of observable events* Σ_o *containing controllable events* $\Sigma_c \subseteq \Sigma_o$ *and the set of unobservable events* Σ_u *containing the fault* $\mathbf{f}$*;*
- $T : Q \times \Sigma \times Q \to \mathbb{N}$ *is the transition function that associates an integer weight with each transition.*

A CLTS induces a labelled transition system which transition relation is defined by $q \xrightarrow{a} q'$ if $T(q, a, q') > 0$. The extended relation $\Rightarrow$ and the other usual definitions are defined as for pLTS. We assume that the CLTS is convergent and live.

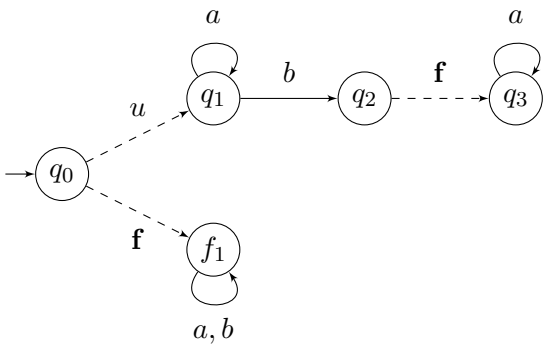


Figure 6.2: An example of CLTS. Weights are all equal to 1 and omitted on the figure. The only controllable event is b.

Example 6.2. *A CLTS* $\mathcal{C}$ *is represented in Figure 6.2. If the control enables every event, the run* $q_0uq_1aq_1bq_2$ *has probability* $1/8$*. If the control always forbids 'b', this same run has probability* 0*. And if it only allows 'b' after observing one 'a', it has probability* $1/4$*.*

We now formalise the ingredients necessary to define how to control CLTS. Let $\Sigma^\bullet \subseteq \Sigma$ and $q \in Q$, let us write $G^{\Sigma^\bullet}(q)$ for the sum of the weights of the transitions exiting q and labelled by an event of $\Sigma^\bullet$. Using this sum, we define a normalisation of the transition relation restricted to the events of $\Sigma^\bullet$ by:

$$T^{\Sigma^\bullet}(q,a,q') = \begin{cases} \frac{T(q,a,q')}{G^{\Sigma^\bullet}(q)} \text{ if } a \in \Sigma^\bullet \text{ and } G^{\Sigma^\bullet}(q) > 0 \\ 0 \text{ otherwise.} \end{cases}$$

A *strategy* of a CLTS $\mathcal{C}$ is a function $\pi : \Sigma_o^* \to \mathsf{Dist}(2^\Sigma)$ such that for all $w \in \Sigma_o^*$ and all $\Sigma^\bullet \in \mathsf{Supp}(\pi(w))$, $\Sigma \setminus \Sigma_c \subseteq \Sigma^\bullet$. In other words, given an observation, a strategy chooses (possibly with randomisation) a set of allowed events that contains the uncontrollable events. Let $\mathcal{C}$ be a CLTS and π be a strategy, we consider the *configurations* of the form $(w, q, \Sigma^\bullet) \in \Sigma_o^* \times Q \times 2^\Sigma$ with w the observed sequence, q the current state and $\Sigma^\bullet$ the set of allowed events by π after observation of w. We inductively define the set $\mathsf{Reach}_\pi(\mathcal{C})$ of the reachable configurations under π by:

- for all $\Sigma^\bullet \in \mathsf{Supp}(\pi(\varepsilon))$, we have $(\varepsilon, q_0, \Sigma^\bullet) \in \mathsf{Reach}_\pi(\mathcal{C})$;
- for all $(w, q, \Sigma^\bullet) \in \mathsf{Reach}_\pi(\mathcal{C})$ and all $a \in \Sigma_u$ such that $q \xrightarrow{a} q'$, we have $(w, q', \Sigma^\bullet) \in \mathsf{Reach}_\pi(\mathcal{C})$, and the corresponding transition is denoted by $(w, q, \Sigma^\bullet) \xrightarrow{a}_\pi (w, q', \Sigma^\bullet)$;
- for all $(w, q, \Sigma^\bullet) \in \mathsf{Reach}_\pi(\mathcal{C})$, all $a \in \Sigma_o \cap \Sigma^\bullet$ such that $q \xrightarrow{a} q'$ and all $\Sigma^{\bullet\prime} \in \mathsf{Supp}(\pi(wa))$, we have $(wa, q', \Sigma^{\bullet\prime}) \in \mathsf{Reach}_\pi(\mathcal{C})$, and the corresponding transition is denoted by $(w, q, \Sigma^\bullet) \xrightarrow{a}_\pi (wa, q', \Sigma^{\bullet\prime})$.

A strategy π is called *live* if for every configuration $(w, q, \Sigma^\bullet) \in \mathsf{Reach}_\pi(\mathcal{C})$, we have $G^{\Sigma^\bullet}(q) \neq 0$. Only the live strategies are relevant as the other strategies create deadlocks. We are now in a position to introduce the semantics of a CLTS controlled by a live strategy π in terms of a live pLTS. Its set of states is $\mathsf{Reach}_\pi(\mathcal{C})$ augmented by an initial state to randomly choose the initial control according to $\pi(\varepsilon)$. The probability distributions are based on $T^{\Sigma^\bullet}$ if the current control is $\Sigma^\bullet$ combined with the random choice of π in case of an observable event occurrence.

Definition 6.5. *Let $\mathcal{C}$ be a CLTS and π be a live strategy, the pLTS $\mathcal{C}_\pi$ induced by the strategy π on $\mathcal{C}$ is defined by $\mathcal{C}_\pi = \langle Q_\pi, \Sigma, q_{0\pi}, T_\pi, \mathbf{P}_\pi \rangle$ where:*

- $Q_\pi = \{q_{0\pi}\} \cup \mathsf{Reach}_\pi(\mathcal{C})$;
- *for every* $(\varepsilon, q_0, \Sigma^\bullet) \in \mathsf{Reach}_\pi(\mathcal{C})$, $(q_{0\pi}, u, (\varepsilon, q_0, \Sigma^\bullet)) \in T_\pi$;
- *for every* $(w, q, \Sigma^\bullet), (w', q', \Sigma^{\bullet\prime}) \in \mathsf{Reach}_\pi(\mathcal{C})$, $\big((w, q, \Sigma^\bullet), a, (w', q', \Sigma^{\bullet\prime})\big) \in T_\pi$ *iff* $(w, q, \Sigma^\bullet) \xrightarrow{a}_\pi (w', q', \Sigma^{\bullet\prime})$;
- *for every* $(\varepsilon, q_0, \Sigma^\bullet) \in \mathsf{Reach}_\pi(\mathcal{C})$, $\mathbf{P}_\pi(q_{0\pi}, u, (\varepsilon, q_0, \Sigma^\bullet)) = \pi(\varepsilon)(\Sigma^\bullet)$;
- *for every* $((w, q, \Sigma^\bullet), a, (w, q', \Sigma^\bullet)) \in T_\pi$ *and every* $a \in \Sigma_u$, $\mathbf{P}_\pi\left((w, q, \Sigma^\bullet), a, (w, q', \Sigma^\bullet)\right) = T^{\Sigma^\bullet}(q, a, q')$;

- *for every* $\Big((w, q, \Sigma^\bullet), a, (wa, q', \Sigma^{\bullet'})\Big) \in T_\pi$ *and every* $a \in \Sigma_o \cap \Sigma^\bullet$,
  $\mathbf{P}_\pi\Big((w, q, \Sigma^\bullet), a, (wa, q', \Sigma^{\bullet'})\Big) = T^{\Sigma^\bullet}(q, a, q') \cdot \pi(w.a)(\Sigma^{\bullet'})$.

Example 6.3. *Consider the CLTS* $\mathcal{C}$ *depicted in Figure 6.2. There are two possible enabled subsets:* Σ *and* $\Sigma \setminus \{b\}$ *that we denote* Σ^-*. Let us define the strategy* π *by* $\pi(a^n) = p_n \cdot \Sigma^- + r_n \cdot \Sigma$ *with* $p_n + r_n = 1$ *for all* $n \in \mathbb{N}$ *and* $\pi(w) = \mathbf{1}_\Sigma$ *otherwise. The generated pLTS* $\mathcal{C}_\pi$ *is infinite. A part of it is represented in Figure 6.3. Let us develop the distribution of probabilities exiting the configuration* $(\varepsilon, q_1, \Sigma)$*. The two transitions exiting* q_1 *are enabled with equal probabilities, thus normalised to 0.5. Since 'a' and 'b' are observable, the new control is chosen, in the case where a 'a' is observed, by a probabilistic choice* $p_1 \cdot \Sigma^- + r_1 \cdot \Sigma$ *while if a 'b' is observed, there is a deterministic choice* $\mathbf{1}_\Sigma$*. This results in three transitions with probability* $0.5p_1$*,* $0.5r_1$ *and* 0.5 *respectively.*

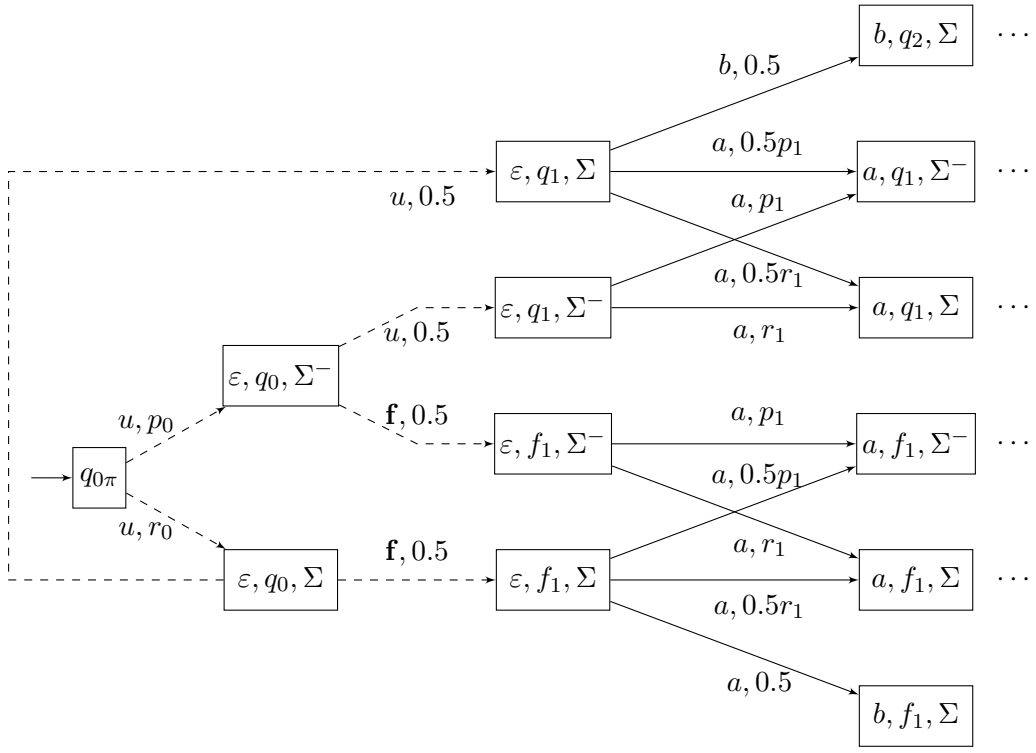


Figure 6.3: An example of controlled CLTS.

In Definition 3.2, page 58, we introduced finite-memory diagnosers. Similarly, one can formally define finite-memory strategies for CLTS using a set of memory states, a memory update function indicating how observations modify the memory state and a decision function mapping every memory state to a choice of the strategy. The size of the memory is the number of memory states. If the size of the memory of a strategy π of a CLTS $\mathcal{C}$ is finite, then $\mathcal{C}_\pi$ is also finite.

Let us define the problems of active diagnosis in the context of the degradation control. Roughly speaking, given a CLTS, one asks whether there exists a strategy such that the associated pLTS is FF-diagnosable and satisfies the required property related to degradation. We distinguish, as usually done, the quantitative problems and the qualitative ones (such as safety, lasting fault freeness and strong/weak resiliency).

Definition 6.6 (Quantitative problems)**.** *Given a CLTS $\mathcal{C}$, $0 < \varepsilon, \alpha < 1$, $0 < \gamma \leq 1$ and $v \in [0, \infty]$:*

- *The* ε-safe active diagnosis *problem consists in deciding if there exists a strategy π such that $\mathcal{C}_\pi$ is* FF*-diagnosable and ε-safe;*
- *The* (γ, v)-fault free active diagnosis *problem consists in deciding if there exists a strategy π such that $\mathcal{C}_\pi$ is* FF*-diagnosable and (γ, v)-fault free;*
- *The* α-resilient active diagnosis *problem consists in deciding if there exists a strategy π such that $\mathcal{C}_\pi$ is* FF*-diagnosable and α-resilient.*

Definition 6.7 (Qualitative problems)**.** *Given a CLTS $\mathcal{C}$:*

- *The* safe active diagnosis *problem consists in deciding if there exists a strategy π such that $\mathcal{C}_\pi$ is* FF*-diagnosable and safe;*
- *The* lasting fault free active diagnosis *problem consists in deciding if there exists a strategy π such that $\mathcal{C}_\pi$ is* FF*-diagnosable and lasting fault free;*
- *The* strongly resilient active diagnosis *problem consists in deciding if there exists a strategy π such that $\mathcal{C}_\pi$ is* FF*-diagnosable and strongly resilient;*
- *The* weakly resilient active diagnosis *problem consists in deciding if there exists a strategy π such that $\mathcal{C}_\pi$ is* FF*-diagnosable and weakly resilient.*

When tackling problems on strategies, the first step is to wonder if one can restrict the strategies that are considered. For example, can we use strategies with finite memory. This cannot be done as shown in the following example.

Example 6.4. *In order to illustrate the impact of taking into account infinite memory strategies, let us examine the CLTS $\mathcal{C}$ of Figure 6.2. The only ambiguous observed sequence is a^ω. A strategy π thus makes it* FF*-diagnosable iff the probability of faulty runs with this observed sequence in $\mathcal{C}_\pi$ is 0. This is done by allowing 'b' often enough so that it occurs with probability 1. However, the only correct run is $\rho = q_o u(q_1 a)^\omega$ with observation a^ω. Thus, $\mathcal{C}$ is not actively safely diagnosable.*

Let us denote, as in Example 6.3, by p_n the probability to forbid 'b' after the observed sequence a^n given by the strategy π. Then $\mathbb{P}_{\mathcal{C}_\pi}(q_o u(aq_1)^n) = \frac{1}{2}\prod_{i \leq n} \frac{1+p_i}{2}$. Thus, by choosing $p_n = 1 - \frac{1}{n+1}$, $\mathcal{C}_\pi$ is FF*-diagnosable, lasting fault free and strongly resilient. On the other hand, no finite-memory strategy could achieve this goal since otherwise by Theorem 6.1, $\mathcal{C}$ would be actively safely diagnosable.*

Restricting one-self to finite-memory strategies is thus a loss of generality. It can however be useful to regain decidability of difficult problems as we will see later.

2 Algorithmic analysis of degradation

In this section we answer the problems listed above by establishing if they are decidable and in the positive case by giving their exact complexities. We start in Subsection 2.1 by proving the undecidability of the quantitative problems. As a consequence, in Subsection 2.2, we focus on the qualitative problems and prove the EXPTIME-completeness of all of them except the safe active diagnosis problem. Lastly, in Subsection 2.3, we see that the safe active diagnosis problem is more difficult but can still be decided efficiently when restricted to finite-memory strategies.

2.1 Undecidability of the quantitative problems

The quantitative problems turn out to all be undecidable. The proofs of these results are obtained by reductions from the emptiness problem of probabilistic automata[1].

We start by showing the result for the ε-safe diagnosis problem, with $\varepsilon > 0$.

Proposition 6.1. *The ε-safe active diagnosis problem is undecidable.*

The idea of this proof is the following. Given a probabilistic automaton A with alphabet Σ, one builds a CLTS $\mathcal{C}$ composed of two independent parts each one initially entered with probability $\frac{1}{2}$ by an unobservable transition. The unobservable event leading to the first part is the fault $\mathbf{f}$ which can only be detected almost surely if the observable event $\sharp \notin \Sigma$ occurs with probability 1. The second part is constituted of a CLTS version of A augmented by exiting transitions. One can exit A by allowing a $\sharp$. When this happens, if the system was in a final state of A it goes to a correct BSCC of the CLTS, ensuring the run will remain correct. Else a fault is triggered on the next step. This construction ensures the following properties. If there exists a word w with an acceptance probability at least 2ε, the strategy which consists in forcing the observed sequence $w\sharp$ ensures a probability of the set of infinite correct runs of at least ε. In the opposite case, we show that no strategy can achieve this threshold.

Proof. Let $0 < \varepsilon < 1/2$. We proceed here by reduction from the problem of the existence of a word w such that $\mathbf{P}_A(w) \geq 2\varepsilon$. We consider a probabilistic automaton $A = \langle Q, q_0, \Sigma, (\mathbf{P}_a)_{a\in\Sigma}, F\rangle$ for which w.l.o.g. we assume that: (1) $\Sigma \cap \{u, \mathbf{f}, \sharp, \natural\} = \emptyset$ and (2) the probabilities are fractions $\frac{n}{d}$ with fixed denominator $d \in \mathbb{N}$. One builds the CLTS $\mathcal{C} = \langle Q', q'_0, \Sigma', T\rangle$ described in Figure 6.4 and defined by:

- $Q' = Q \cup \{q'_0, q_c, q_f, f_1, f_2\}$;
- $\Sigma' = \Sigma \cup \{\mathbf{f}, u, \sharp, \natural\}$, $\Sigma_u = \{\mathbf{f}, u\}$ and $\Sigma_c = \Sigma \cup \{\sharp\}$;
- the transition function T is defined as follows.

  1.

$$T(q'_0, \mathbf{f}, f_1) = T(q'_0, u, q_0) = T(q_c, \sharp, q_c) = T(q_f, \mathbf{f}, f_2) = T(f_2, \natural, f_2) = T(f_1, \sharp, f_2) = 1;$$

[1] see page 115.

2. for every $a \in \Sigma$, $T(f_1, a, f_1) = 1$;
3. for every $s, s' \in Q$ and every $a \in \Sigma$, $T(s, a, s') = d \cdot \mathbf{P}_a(s, s')$;
4. for every $s \in F$, $T(s, \sharp, q_c) = 1$ and for every $s \in S \setminus F$, $T(s, \sharp, q_f) = 1$;
5. for every other triplet, T is equal to 0.

As detailed above, the probabilities in A are all multiplied by their common denominator d, to obtain integer weights, and we write $d \cdot A$ in the figure to represent this scaling.

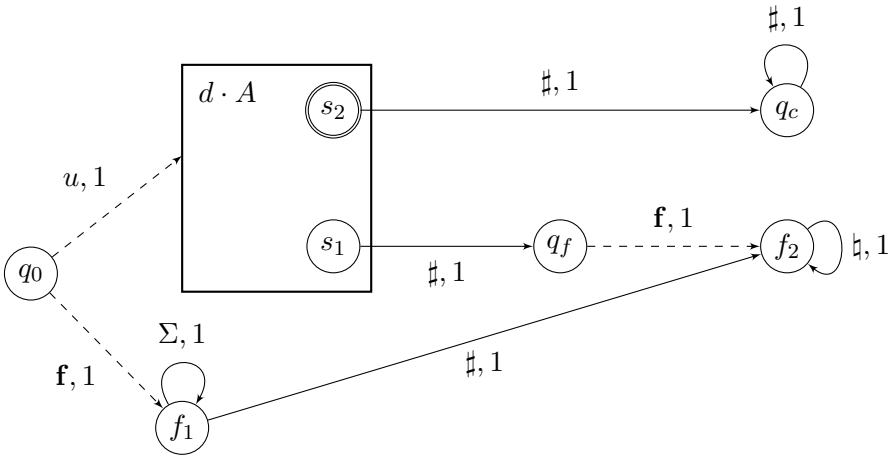


Figure 6.4: Reduction to ε-safe diagnosability.

Let us show that there exists a strategy π such that $\mathcal{C}_\pi$ is ε-safe and FF-diagnosable iff there exists a word w accepted in A with probability at least 2ε.

Remark first that, for π an arbitrary strategy, $\mathcal{C}_\pi$ is FF-diagnosable iff $\sharp$ occurs almost surely in a run. Indeed, an observed sequence $w \in \Sigma^*$ is ambiguous. On the other hand every faulty run ρ triggering a $\sharp$ will produce a $\natural$ removing the ambiguity.

• Assume there exists a word $w = a_1 \dots a_k \in \Sigma^*$ such that $\mathbf{P}_A(w) \geq 2\varepsilon$. We define the deterministic strategy π by:

- $\pi(w) = \{\mathbf{f}, u, \sharp, \natural\}$;
- for all $0 \leq i < k$, $\pi(a_1 \dots a_i) = \{\mathbf{f}, u, a_{i+1}, \natural\}$;
- $\pi(w') = \Sigma'$ for any other word w'.

Observe that after at most $k+1$ observable events, any run leaves $Q \cup \{f_1\}$ and thus $\natural$ occurs almost surely implying that $\mathcal{C}_\pi$ is FF-diagnosable. Moreover, the probability of correct runs with observation $w\sharp\sharp$ is equal to $\frac{\mathbf{P}_A(w)}{2}$: it is the probability to take u initially times the probability to end the observation of w in an accepting state of A. As $\mathbf{P}_A(w) \geq 2\varepsilon$, this ensures that $\mathcal{C}_\pi$ is ε-safe.

• Assume now that for all $w \in \Sigma^*$, $\mathbf{P}_A(w) < 2.\varepsilon$. Let π be a strategy such that $\mathcal{C}_\pi$ is FF-diagnosable, thus with probability 1 an infinite run contains a $\sharp$. Moreover, this run is correct iff the first $\sharp$ is followed by a second $\sharp$. Then we have:

$$\begin{aligned}\mathbb{P}_\pi(\mathsf{C}_\infty) &= \sum_{w\in\Sigma^*} \mathbb{P}_\pi(w\sharp\sharp) \\ &= \sum_{w\in\Sigma^*} \frac{\mathbb{P}_\pi(w\sharp)\cdot \mathbf{P}_A(w)}{2} \\ &< \varepsilon \sum_{w\in\Sigma^*} \mathbb{P}_\pi(w\sharp) \\ &= \varepsilon.\end{aligned}$$

Therefore, $\mathcal{C}_\pi$ is not ε-safe, which concludes the reduction and proves undecidability of the ε-safe active diagnosis problem. □

We now turn to the (γ, v)-fault free active diagnosis problem. It is done once again by reduction from the emptiness problem of PA. In fact, it has many similarities with the previous proof, but instead of reaching a state q_c where the run will stay correct, being accepted by the PA only postpones the fault by one step.

Proposition 6.2. *The (γ, v)-fault free active diagnosis problem is undecidable.*

The idea of this proof is the following. Given a probabilistic automaton A with alphabet Σ, one builds a CLTS $\mathcal{C}$ composed of two independent parts each one initially entered with probability $\frac{1}{2}$ by an unobservable transition. The unobservable event leading to the first part is the fault $\mathbf{f}$ which can only be detected almost surely if the observable event $\sharp \notin \Sigma$ occurs with probability 1. The second part is constituted of a CLTS version of A augmented by exiting transitions. One exits A with probability $\frac{1}{2}$ at every step towards a faulty sub-part except if the $\sharp$ event is triggered. In this case, if the system was in a final state of A it leaves the states of A and postpones the occurrence of a fault by one time step compared to if it stayed in A. This construction ensures the following properties. If there exists a word w with an acceptance probability at least $\frac{1}{2}$, the strategy which consists in forcing the observed sequence $w\sharp$ as long as the run stays in A ensures an average observable length (without discount) of the maximal correct signalling prefix greater or equal to 1. In the opposite case, we show that no strategy can achieve this threshold.

Proof. We proceed here by reduction from the problem of the existence of a word w such that $\mathbf{P}_A(w) \geq \frac{1}{2}$. We consider the probabilistic automaton $A = \langle Q, q_0, \Sigma, (\mathbf{P}_a)_{a\in\Sigma}, F\rangle$ for which w.l.o.g. we assume that: (1) $\Sigma \cap \{u, \mathbf{f}, \sharp, \natural\} = \emptyset$ and (2) the probabilities are fractions $\frac{n}{d}$ with fixed denominator d. One builds the CLTS $\mathcal{C} = \langle Q', q'_0, \Sigma', T\rangle$ described in Figure 6.5 and defined by:

- $Q' = Q \cup \{q'_0, q_c^1, q_c^2, q_c^3, f_1, f_2\}$;
- $\Sigma' = \Sigma \cup \{\mathbf{f}, u, \sharp, \natural\}$, $\Sigma_u = \{\mathbf{f}, u\}$ and $\Sigma_c = \Sigma \cup \{\sharp\}$;

- the transition function T is defined as follows.

  1.
  $$T(q_0', \mathbf{f}, f_1) = T(q_0', u, q_0) = T(q_c^1, \sharp, q_c^3) = T(q_c^3, \sharp, q_c^3) = T(q_c^3, \mathbf{f}, f_2)$$
  $$= T(q_c^2, \mathbf{f}, f_2) = T(f_2, \natural, f_2) = T(f_1, \sharp, f_2) = 1;$$

  2. for every $a \in \Sigma$, $T(f_1, a, f_1) = 1$;
  3. for every $s, s' \in Q$ and every $a \in \Sigma$, $T(s, a, s') = d \cdot \mathbf{P}_a(s, s')$ and $T(s, a, q_c^2) = d$;
  4. for every $s \in F$, $T(s, \sharp, q_c^1) = 1$ and for every $s \in S \setminus F$, $T(s, \sharp, q_c^2) = 1$;
  5. for every other triplet, T is equal to 0.

Here again, the probabilities in A are multiplied by the constant d, which we abbreviate in the figure by $d \cdot A$.

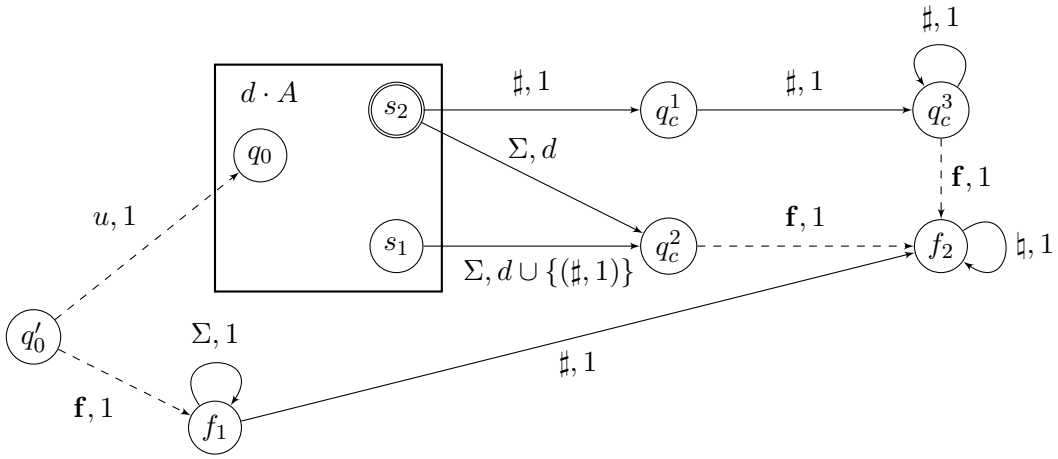


Figure 6.5: Reduction to (γ, v)-fault free active diagnosability.

Let us show that there exists a strategy π such that $\mathcal{C}_\pi$ is $(1,1)$-fault free and FF-diagnosable iff there exists a word w accepted in A with probability at least $\frac{1}{2}$.

Remark first that, for π an arbitrary strategy, $\mathcal{C}_\pi$ is FF-diagnosable iff $\natural$ occurs almost surely in a run. Indeed an observed sequence $w \in \Sigma^*$ is ambiguous. On the other hand every run ρ leaving $Q \cup \{f_1\}$ almost surely reaches f_2 where $\natural$ occurs and, whatever ρ, a fault has occurred.

• Assume that there exists $w = w_1 \dots w_k \in \Sigma^*$ such that $\mathbf{P}_A(w) \geq \frac{1}{2}$. We define the deterministic strategy π by:

- $\pi(w) = \{\mathbf{f}, u, \sharp, \natural\}$;
- for all $0 \leq i < k$, $\pi(w_1 \dots w_i) = \{\mathbf{f}, u, w_{i+1}, \natural\}$;

- $\pi(w') = \Sigma'$ for any other word w'.

Observe that after at most $k+1$ observable events, any run leaves $S \cup \{f_1\}$ and thus $\natural$ occurs almost surely implying that $\mathcal{C}_\pi$ is diagnosable.

By definition of $\mathcal{C}$ and π, a correct signalling run ρ such that $\mathcal{P}(\rho) = w_1 \dots w_i$ for $i < k$ has probability $\frac{1}{2}$ of staying correct at the next step depending on if the current state is q_c^2 or belongs to Q. Similarly, a correct signalling run ρ such that $\mathcal{P}(\rho) = w_1 \dots w_k$ has a probability $\mathbf{P}_A(w)$ of being at the next step in q_c^1 and $1 - \mathbf{P}_A(w)$ in q_c^2. Moreover, in state q_c^3, a correct signalling run has a probability $\frac{1}{2}$ of staying correct and in q_c^3 at the next step. Therefore for all $n \in \mathbb{N}$, we have $n \leq k$ implies $\mathbb{P}(\mathsf{C}_n) = (\frac{1}{2})^n$ and $n > k$ implies $\mathbb{P}(\mathsf{C}_n) = (\frac{1}{2})^{n-1}\mathbf{P}_A(w) \geq (\frac{1}{2})^n$. Finally: $\sum_{n=1}^{\infty} \mathbb{P}(\mathsf{C}_n) \geq \sum_{n=1}^{\infty} (\frac{1}{2})^n = 1$.

• Assume that for all $w \in \Sigma^*$, $\mathbf{P}_A(w) < \frac{1}{2}$. Let π be a strategy such that $\mathcal{C}_\pi$ is diagnosable. Observe that (using a slight and understandable abuse of language):

$$\mathbb{P}_\pi(\mathsf{C}_n) = \sum_{w \in \Sigma^n} \mathbb{P}_\pi(w \wedge \mathsf{C}) + \sum_{w \in \Sigma^{n-1}} \mathbb{P}_\pi(w\sharp \wedge \mathsf{C}) + \sum_{1<k\leq n} \sum_{w \in \Sigma^{n-k}} \mathbb{P}_\pi(w\sharp^k \wedge \mathsf{C}).$$

Let us show that $\mathbb{P}_\pi(\mathsf{C}_{n+1}) \leq \frac{\mathbb{P}_\pi(\mathsf{C}_n)}{2}$ with a strict inequality if there exists $w \in \Sigma^{n-1}$ with $\mathbb{P}_\pi(w\sharp) > 0$.

$$\mathbb{P}_\pi(\mathsf{C}_{n+1}) = \sum_{w \in \Sigma^n} \sum_{x \in \Sigma \cup \{\sharp\}} \mathbb{P}_\pi(wx \wedge \mathsf{C}) + \sum_{w \in \Sigma^{n-1}} \mathbb{P}_\pi(w\sharp^2 \wedge \mathsf{C}) + \sum_{1<k\leq n} \sum_{w \in \Sigma^{n-k}} \mathbb{P}_\pi(w\sharp^{k+1} \wedge \mathsf{C})$$

Let us examine the three terms.

∘ A correct run ρ with observed sequence w has a conditional equiprobability that $\mathsf{last}(\rho) \in Q$ or $\mathsf{last}(\rho) = q_c^2$. Thus, $\sum_{w \in \Sigma^n} \sum_{x \in \Sigma \cup \{\sharp\}} \mathbb{P}_\pi(wx) = \frac{1}{2} \sum_{w \in \Sigma^n} \mathbb{P}_\pi(w)$.

∘ A correct run ρ with observed sequence $w\sharp^k$ such that $k > 1$ verifies $\mathsf{last}(\rho) = q_c^3$. Thus, $\sum_{1<k\leq n} \sum_{w \in \Sigma^{n-k}} \mathbb{P}_\pi(w\sharp^{k+1} \wedge \mathsf{C}) = \frac{1}{2} \sum_{1<k\leq n} \sum_{w \in \Sigma^{n-k}} \mathbb{P}_\pi(w\sharp^k \wedge \mathsf{C})$

∘ A correct run ρ of observed sequence $w\sharp$ has a conditional probability $\mathbf{P}_A(w)$ that $\mathsf{last}(\rho) = q_c^1$ and $1 - \mathbf{P}_A(w)$ that $\mathsf{last}(\rho) = q_c^2$. Thus:

$$\sum_{w \in \Sigma^{n-1}} \mathbb{P}_\pi(w\sharp^2 \wedge \mathsf{C}) = \sum_{w \in \Sigma^{n-1}} \mathbf{P}_A(w)\mathbb{P}_\pi(w\sharp \wedge \mathsf{C}) \leq \frac{1}{2} \sum_{w \in \Sigma^{n-1}} \mathbb{P}_\pi(w\sharp \wedge \mathsf{C})$$

with a strict inequality if there exists a word $w \in \Sigma^{n-1}$ with $\mathbb{P}_\pi(w\sharp) > 0$.

By assumption, $\mathcal{C}_\pi$ is diagnosable. Thus, according to our characterisation of a strategy ensuring FF-diagnosability, there exists a word w such that $\mathbb{P}_\pi(w\sharp) > 0$. As a consequence, $\sum_{n=1}^{\infty} \mathbb{P}(\mathsf{C}_n) < \sum_{n=1}^{\infty} (\frac{1}{2})^n = 1$, thus $\mathcal{A}$ is not $(1,1)$ fault free. □

Remark 6.1. *A straightforward adaptation of the proof shows that for every $0 < \gamma < 1$, $\mathcal{A}$ is $(\gamma, \frac{\gamma}{2-\gamma})$ fault free iff there exists a word w such that $\mathbf{P}_A(w) \geq \frac{1}{2}$.*

We end with the α-resilient active diagnosis problem. The construction of the reduction is a bit simpler. This is due to the fact that the system is FF-diagnosable for any arbitrary strategy. In other words, the reduction only relies on the α-resilient property to establish undecidability.

Proposition 6.3. *The α-resilient active diagnosis problem is undecidable.*

This time, given a probabilistic automaton A with alphabet Σ, one transforms A into a CLTS, augmented by two states and some transitions. This CLTS is called $\mathcal{C}$ and its initial state is the initial state of A. At each step, when reading an event of Σ, with probability $1/2$ we exit A and will commit a fault in the next step. When a $\sharp$ is read after a word $w_1\sharp\dots\sharp w_k$ with for all $i \leq k$ w_i does not contain $\sharp$, either we go back to the initial state of A or we will trigger a fault on the next turn depending on the probability to accept w_k. If a strategy can regularly trigger a word with acceptance probability greater than $1/2$, it can slow the speed at which the runs become faulty.

Proof. We proceed here by reduction from the problem of the existence of a word w such that $\mathbf{P}_A(w) > \frac{1}{2}$. We consider a probabilistic automaton $A = \langle Q, q_0, \Sigma, (\mathbf{P}_a)_{a\in\Sigma}, F\rangle$ for which we assume w.l.o.g. that: (1) $\Sigma \cap \{u, \mathbf{f}, \sharp, \natural\} = \emptyset$ and (2) the probabilities are fractions $\frac{n}{d}$ with $d \in \mathbb{N}$ fixed. One builds the CLTS $\mathcal{C} = \langle Q', q_0, \Sigma', T\rangle$ represented in Figure 6.6 (with some shortcuts to ease readability) and defined by:

- $Q' = Q \cup \{q_1, f_1\}$;
- $\Sigma' = \Sigma \cup \{\mathbf{f}, \sharp, \natural\}$, $\Sigma_u = \{\mathbf{f}\}$ et $\Sigma_c = \Sigma \cup \{\sharp\}$;
- the transition function T is defined by:
  1. $T(q_1, \mathbf{f}, f_1) = T(f_1, \natural, f_1) = 1$;
  2. for every $s, s' \in Q, a \in \Sigma, T(s, a, s') = d \cdot \mathbf{P}_a(s, s')$ and $T(s, a, q_1) = d$;
  3. for every $s \in F$, $T(s, \sharp, s_0) = 1$ and for every $s \in S \setminus F$, $T(s, \sharp, q_1) = 1$;
  4. for every other triplet, T is equal to 0.

Once again, the probabilities in A are multiplied by the constant d, which we abbreviate in the figure by $d \cdot A$.

As every fault is followed by a $\natural$, whatever the strategy π, $\mathcal{C}_\pi$ is FF-diagnosable.

• Assume there exists $w = w_1 \dots w_k \in \Sigma^*$ such that $\mathbf{P}_A(w) > \frac{1}{2}$. We denote $v = \mathbf{P}_A(w)$. We define the deterministic strategy π by:

- $\pi((w\sharp)^*w) = \{\mathbf{f}, \natural, \sharp\}$;
- for all $0 \leq i < k$, $\pi((w\sharp)^*w_1 \dots w_i) = \{\mathbf{f}, \natural, w_{i+1}\}$;
- $\pi(w') = \Sigma'$ for any other word w'.

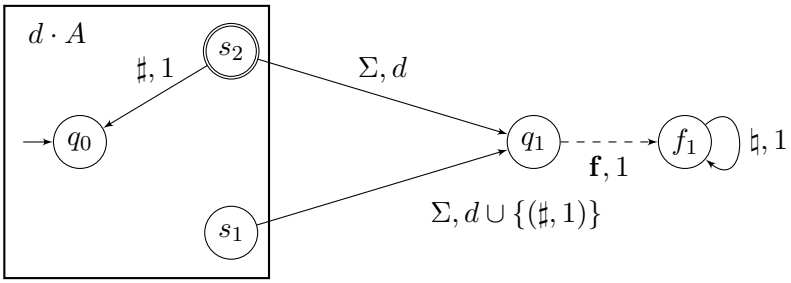


Figure 6.6: Reduction to α-resilient active diagnosability.

Under strategy π, the observed sequence of a correct run ρ is some $(w\sharp)^m w_1 \dots w_i$ with $0 \leq i \leq k$.
$\circ$ If $\mathcal{P}(\rho) = (w\sharp)^m w_1 \dots w_i$ with $0 < i$ then with conditional equiprobability, $\mathsf{last}(\rho) \in Q$ or $\mathsf{last}(\rho) = q_1$. Thus with probability $\frac{1}{2}$, the run will be correct after the next observation.
$\circ$ If $\mathcal{P}(\rho) = (w\sharp)^m$ then with conditional probability v, $\mathsf{last}(\rho) = q_0$ and with probability $1 - v$, $\mathsf{last}(\rho) = q_1$. Thus with probability v, the run will be correct after the next observation.

Consider an arbitrary n and write the Euclidian division of $n-1$ by $k+1$ as $n-1 = m(k+1) + i$ with $i \leq k$. One has $2^{-(n-1)}\mathbb{P}_\pi(\mathsf{C}_n) = (2v)^m$. Hence $\frac{2^{(n-1)}}{\mathbb{P}_\pi(\mathsf{C}_n)} = \left(\frac{1}{2v}\right)^{\lfloor \frac{n-1}{k+1} \rfloor}$ implying $\lim_{n\to\infty} \frac{2^{-n}}{\mathbb{P}_\pi(\mathsf{C}_n)} = 0$. So $\mathcal{C}_\pi$ is $\frac{1}{2}$-resilient.

• Assume now that for every word $w \in \Sigma^*$, $\mathbf{P}_A(w) \leq \frac{1}{2}$. Let π be an arbitrary strategy. The observed sequence of a correct run ρ is some $u_1\sharp \dots \sharp u_m$ such that for all i, $u_i \in \Sigma^*$.
$\circ$ If $u_m \neq \varepsilon$ with $0 < i$ then with conditional equiprobability, $\mathsf{last}(\rho) \in Q$ or $\mathsf{last}(\rho) = q_1$. Thus with probability $\frac{1}{2}$, the run will be correct after the next observation.
$\circ$ If $u_m = \varepsilon$ then with conditional probability $\mathbf{P}_A(u_{m-1})$, $\mathsf{last}(\rho) = q_0$ and with probability $1 - \mathbf{P}_A(u_{m-1})$, $\mathsf{last}(\rho) = q_1$. Thus with probability $\mathbf{P}_A(u_{m-1}) \leq 1/2$, the run will be correct after the next observation.

Summarising, one has: $\mathbb{P}_\pi(\mathsf{C}_n) \leq 2^{-(n-1)}$ implying $\limsup_{n\to\infty} \frac{2^{-n}}{\mathbb{P}_\pi(\mathsf{C}_n)} \geq \frac{1}{2}$.
So $\mathcal{C}_\pi$ is not $\frac{1}{2}$-resilient. □

2.2 Decidability of the Qualitative Problems

In contrast to the quantitative notions, and to the notable exception of the safe active diagnosis problem, all the qualitative problems of diagnosability under degradation constraints we introduced are decidable and EXPTIME-complete. The simplest case is the one of weak resilient active diagnosability. The proof idea is common to all cases: starting from a construction that gives an efficient characterisation of active diagnosability (inspired from [BFH+14] and detailed below), we establish a necessary and sufficient condition for the existence of a control strategy that ensures the given notion of diagnosability under a degradation constraint.

Let us start by defining a construction inspired from [BFH+14] [2]. This construction is the adaptation to the active setting of the FF-automaton used in Subsection 1.1.1 of Chapter 4, page 93. This construction uses the notion of *belief*. The initial belief is $\{q_0\}$, and given a current belief B and an observed event b, the belief obtained after b has been observed is defined by:

$$\Delta(B,b) = \{q \in Q \mid \exists q' \in B, \rho \in \mathsf{SR}_1, q' \overset{\rho}{\Rightarrow} q \wedge \mathcal{P}(\rho) = b\} \ .$$

$\Delta(B,b)$ is thus the set of states a partially observable systems may be in, given that the previous belief was B and observation O occurred. Importantly, it does not depend on the strategy as every controllable event is observable. The set of beliefs of a CLTS $\mathcal{C}$ is denoted $\mathcal{B}l_{\mathcal{C}}$ and we drop the subscript when there is no risk of confusion. Beliefs are of importance since they formalize the discrete information an observer has on the current state of the system. Thus, to decide FF-diagnosability of a CLTS, the states of the CLTS are enriched with two sets U and V that correspond, respectively, to the subset of correct, or faulty, states, that are reachable by a signalling run corresponding to the current observed sequence, *i.e.* to the set of correct, or faulty, states of the current belief. Such a pair of sets (U,V) is therefore called a *separated belief*. As we study FF-diagnosability here, one could wonder why we do not only use a set U as we did in Chapter 4. In fact, in some of the constructions that we make later, we need to know the full belief. For example, forgetting a faulty state could result in a controller making a choice that creates a deadlock in this faulty state. By using U and V, we have the information pertaining to FF-diagnosis (U) and to the current belief ($U \cup V$).

Formally, from a CLTS $\mathcal{C} = \langle Q, q_0, \Sigma, T\rangle$, we define its belief version on the same event alphabet $\mathcal{C}^B = \langle Q^B, q_0^B, \Sigma, T^B\rangle$ by:

- $Q^B = Q \times 2^Q \times 2^Q$ and $q_0^B = (q_0, \{q_0\}, \emptyset)$;
- for every $(q,U,V) \in Q \times 2^Q \times 2^Q$, for every $a \in \Sigma$, and every $q' \in Q$
  - if $a \notin \Sigma_o$, $T^B((q,U,V),a,(q',U,V)) = T(q,a,q')$;
  - if $a \in \Sigma_o$, letting $U' = \Delta(U,a) \cap Q_c$ and $V' = \Delta(U \cup V, a) \cap Q_f$, then $T^B((q,U,V),a,(q',U',V')) = T(q,a,q')$.
  - for every other triplet $((q,U,V),a,(q',U',V'))$, T is equal to 0.

The size of the belief CLTS $\mathcal{C}^B$ is exponential in the size of $\mathcal{C}$. For the properties we are interested in, they have the same behaviour. We introduce Θ, a discrete version of T^B, extended to observed sequences. For $w \in \Sigma_o^*$, $(q',U',V') \in \Theta((q,U,V),w)$ as soon as there exists a run ρ such that $\mathcal{P}(\rho) = w$ and $(q,U,V) \overset{\rho}{\Rightarrow} (q',U',V')$.

We now construct Win the set of all separated beliefs (U,V) such that, starting from any (q,U,V) with $q \in U \cup V$, $\mathcal{C}^B$ is actively diagnosable. This set is computed as a greatest fixpoint. We let $\mathsf{Win}_0 = 2^{Q_c} \times 2^{Q_f}$ and for $n \in \mathbb{N}$, Win_{n+1} is the set of the

[2] The difference with the construction of [BFH+14] is that we focus here on FF-diagnosability rather than IA-diagnosability which simplifies the writing of the proofs. The results of this chapter and of [BFH+14] however hold for both notions of diagnosability.

separated beliefs (U, V) of Win_n such that for all state $q \in U \cup V$, there exists a sequence of sets of allowed events $(\Sigma_i^\bullet)_{1 \leq i \leq k}$ and an observed sequence $w = o_1 \dots o_k$ with $o_i \in \Sigma_i^\bullet$ verifying:

- there exists a run ρ starting in (q, U, V) with $\mathcal{P}(\rho) = w$ and reaching (q^*, U^*, V^*) with $q^* \in Q_c$ (*i.e.* the current state is correct) or $U^* = 0$ (the fault is claimed);
- Consider a state q_i reached from $q' \in U \cup V$ by a run with observed sequence $o_1 \dots o_i$ with $0 \leq i < k$, *i.e.* $(q_i, U_i, V_i) \in \Theta((q', U, V), o_1 \dots o_i)$ for a separated belief (U_i, V_i). then:
  1. the control induced by $\Sigma_{i+1}^\bullet$ does not create any deadlock: $G^{\Sigma_{i+1}^\bullet}(q_i) \neq 0$;
  2. Every new separated belief obtained by an observable step $o \in \Sigma_{i+1}^\bullet$ starting in q_i belongs to Win_n: $\forall o \in \Sigma_{i+1}^\bullet, \forall (q_o, U_o, V_o) \in \Theta((q_i, U_i, V_i), o), (U_o, V_o) \in \mathsf{Win}_n$.

The computation of Win is in polynomial time in the size of $\mathcal{C}^B$, given that in every non-terminal iteration at least one separated belief is removed. The correctness of Win is established in the richer context of IA-diagnosability in [BFH+14], and π^* a (deterministic finite-memory) strategy ensuring diagnosability consists in, given a separated belief $(U, V) \in \mathsf{Win}$ choosing the greatest set $\Sigma^\bullet$ such that every possible separated belief reached on the next step still belongs to Win. Thus, π^* is the most permissive strategy ensuring active diagnosability.

To decide weakly (resp. strongly) resilient active diagnosability, and lasting fault free active diagnosability, we build on the belief CLTS construction.

Theorem 6.2. *Weakly resilient active diagnosability is* $\mathsf{EXPTIME}$*-complete.*

Analysing the set of separated beliefs Win gave a condition for the active diagnosability and in the positive case a deterministic finite-memory strategy π^* ensuring it. We show in this proof, that in order for a CLTS to be weakly resilient active diagnosable, it needs (1) to be actively diagnosable and (2) $\mathcal{C}^B$ must contain a reachable cycle of correct states associated with separated beliefs of Win. The idea is that if such a cycle exists, playing a strategy permissive enough (for example π^*), there is a fixed probability to stay within this cycle and this probability can be used to establish a lower bound to the speed at which the system becomes faulty.

The lower bound is straightforward considering that active diagnosability was already proven to be $\mathsf{EXPTIME}$-hard [BFH+14].

Proof. We first establish the membership in $\mathsf{EXPTIME}$. Given a CLTS $\mathcal{C}$, its belief CLTS $\mathcal{C}^B$, and the strategy π^*, we derive a pLTS $\mathcal{A}$. It is obtained from $\mathcal{C}^B$ by restricting it to the states with separated belief in Win and controlled by π^*. We claim that $\mathcal{C}$ is actively diagnosable with guarantee of weak resiliency iff there exists in $\mathcal{A}$ a reachable cycle such that the first component of every state along the cycle is a correct state of $\mathcal{C}$. • Suppose first that such a cycle exists in $\mathcal{A}$. We let $\alpha > 0$ be the probability of this cycle, n_1 its length, n_0 the observed length of the shortest run reaching a state of the

cycle and μ the probability of this run. For all $n \geq n_0$, $\mathbb{P}_{\mathcal{A}}(\mathsf{C}_n) \geq \mu\alpha^{\lceil \frac{n-n_0}{n_1} \rceil}$. As a consequence, $\mathcal{A}$ is α'-resilient for all $\alpha' < \alpha$. $\mathcal{A}$ is thus weakly resilient. Therefore, $\mathcal{C}_{\pi^*}$, which has the same probabilistic behaviour as $\mathcal{A}$ is weakly resilient too.
• Conversely, suppose that there is no such cycle in $\mathcal{A}$. Let π' be a (live) strategy such that $\mathcal{C}_{\pi'}$ is FF-diagnosable. This strategy can be mimicked in $\mathcal{C}^B$, ignoring the separated belief information. The reachable states of $\mathcal{C}^B_{\pi'}$ are associated with separated beliefs of Win (due to the characterisation recalled above). As π^* is the most permissive strategy ensuring to stay in Win, there does not exist any such cycle in $\mathcal{C}^B_{\pi'}$ either. Consequently, there exists $n_f \in \mathbb{N}$ such that every run ρ in $\mathcal{C}^B_{\pi'}$ with $|\rho| \geq n_f$ ends in a state which first component is faulty. Thus $\mathbb{P}_{\mathcal{C}_{\pi'}}(\mathsf{C}_{n_f}) = \mathbb{P}_{\mathcal{C}^B_{\pi'}}(\mathsf{C}_{n_f}) = 0$, which means that $\mathcal{C}_{\pi'}$ is not weakly resilient.

The complexity lower-bound is obtained by reduction from the active diagnosability problem for CLTS, which is known to be $\mathsf{EXPTIME}$-hard [BFH+14]. From $\mathcal{C} = \langle Q, q_0, \Sigma, T\rangle$ a CLTS, we define the CLTS $\mathcal{C}' = \langle Q \cup \{q'_0, q_s\}, q'_0, \Sigma \cup \{\sharp\}, T'\rangle$ with $\sharp$ a fresh observable event, and such that $T'(q'_0, \sharp, q_0) = T'(q'_0, \sharp, q_s) = T'(q_s, \sharp, q_s) = 1$, for every $q, q' \in Q$ and $a \in \Sigma, T'(q, a, q') = T(q, a, q')$ and for every other triplet $T'(q, a, q') = 0$. Clearly enough, $\mathcal{C}'$ is actively diagnosable iff $\mathcal{C}$ is actively diagnosable. Moreover, $\mathcal{C}'$ is safe by construction, and thanks to Theorem 6.1(a), it is strongly resilient, and thus weakly resilient. □

The proof of the next theorem also relies on the set of separated beliefs Win. We build a subset of Win, called WinK. A separated belief (U, V) of Win belongs to WinK if there exists a strategy π such that from every distribution with support $U \cup V$, π guarantees to stay in Win, and to give a positive probability to the set of infinite correct runs. The CLTS is actively diagnosable with guarantee of strong resiliency iff from the initial belief one can reach a belief of WinK while staying in Win. The strategy π defined with the construction of WinK does not necessarily allows to diagnose the system. So the winning strategy consists in cleverly combining the strategy used to make the system FF-diagnosable and π.

Theorem 6.3. *Strongly resilient active diagnosability is* $\mathsf{EXPTIME}$*-complete.*

Proof. Let $\mathcal{C}$ be a CLTS. As in the construction preliminary to Theorem 6.2, we build $\mathcal{C}^B$, Win and π^*. We then define $\mathsf{WinK}_U \subseteq 2^Q \times \mathsf{Win}$ by a greatest fix point computation. For $(U', (U, V)) \in \mathsf{WinK}_U$, (U, V) is a separated belief for which there exists a strategy allowing to a set of runs starting in $U \cup V$ to stay in the states of $\mathcal{C}^B$ associated with a belief of Win, and if the run started in U', it stayed correct. WinK_U is obtained as the limit of a non-increasing sequence $(\mathsf{WinK}_n)_{n\in\mathbb{N}}$ defined inductively by: $\mathsf{WinK}_0 = \{(U', (U, V)) \mid (U, V) \in \mathsf{Win} \wedge \emptyset \neq U' \subseteq U\}$ and for $n \in \mathbb{N}$, WinK_{n+1} is the set of elements $(U', (U, V))$ of WinK_n such that there exist a set of allowed events $\Sigma^\bullet$ verifying:

- $\Sigma^\bullet$ does not create a deadlock: $\forall q \in U \cup V, G^{\Sigma^\bullet}(q) \neq 0$;
- under the control $\Sigma^\bullet$ no run starting in a state of U' will make a fault before the next observation: $\forall q_c \in U', \forall \rho \in \mathsf{SR}_1, q_c \stackrel{\rho}{\Rightarrow} q_\wedge \mathcal{P}(\rho) \in \Sigma^\bullet \Rightarrow q \in Q_c$;

- every triplet reached by an observable step $o \in \Sigma^\bullet$ belongs to WinK_n: $(\tilde{U}', (\tilde{U}, \tilde{V})) \in \mathsf{WinK}_n$ with:

  1. $\tilde{U}' = \{q'_c \in Q_c \mid \exists q_c \in U'_1, \exists \rho \in \mathsf{SR}_1, q_c \stackrel{\rho}{\Rightarrow} q'_c \wedge \mathcal{P}(\rho) = a\}$;
  2. $\tilde{U} = \Delta(U, o) \cap Q_c$ and $\tilde{V} = \Delta(U \cup V, o) \cap Q_f$.

From WinK_U, we define the set $\mathsf{WinK} \subseteq \mathsf{Win}$ by keeping only the second component of WinK_U: $\mathsf{WinK} = \{(U, V) \in \mathsf{Win} \mid \exists U', (U', (U, V)) \in \mathsf{WinK}_U\}$. Let us state some of the properties of this construction.

- By induction, if $(U', (U, V)) \notin \mathsf{WinK}_n$ then for every (live) strategy and $q \in U$, there exists a faulty run starting in q of observable length n;
- If $\emptyset \neq U'' \subseteq U'$ then $(U', (U, V)) \in \mathsf{WinK}_U$ implies $(U'', (U, V)) \in \mathsf{WinK}_U$. Thus, if $(U, V) \notin \mathsf{WinK}$, for all $q \in U$, $(\{q\}, (U, V)) \notin \mathsf{WinK}_U$.

We also define PreWin the set of states of $\mathcal{C}^B$ of the form $Q \times \mathsf{Win}$ from which a state (q, U, V) with $(U, V) \in \mathsf{WinK}$ is reachable. Let us show that $\mathcal{C}$ is diagnosable and strongly resilient iff the initial state of $\mathcal{C}^B$ belongs to PreWin.

• Suppose that the initial state belongs to PreWin. Let $(U', (U, V))$ be an element of WinK_U. We define $\pi_{(U',(U,V))}$ the strategy that ensures to stay in WinK_U. This strategy immediately derives from the fixpoint definition of WinK_U. For $(U, V) \in \mathsf{WinK}$, we also define $\pi_{(U,V)} = \pi_{(U',(U,V))}$ for an arbitrary U' such that $(U', (U, V)) \in \mathsf{WinK}_U$. Finally, we let π_0 be the following strategy working in three successive phases which may not all be triggered.

1. First π_0 mimics π^* until a separated belief $(U, V) \in \mathsf{WinK}$ is reached;
2. Then, at every observed sequence w, π_0 chooses to apply $\pi_{(U,V)}$ with probability $p_w = \frac{|w|}{|w|+1}$, and to switch to the third phase with probability $1 - p_w$;
3. Finally, π_0 behaves forever as π^*.

We observe that $\mathcal{C}_{\pi_0}$ is FF-diagnosable. Indeed, on the one hand, the events allowed by π_0 are included in those allowed by the maximally permissive strategy π^*, and on the other hand almost-surely, π^* is applied from some moment on. Therefore every fault will almost surely be detected.

Moreover, let us prove that it is strongly resilient. Indeed, by definition of PreWin, there exists a run ρ starting in the initial state and reaching a state (q, U, V) such that (U, V) belongs to WinK. Let $U' \subseteq U$ the one chosen arbitrarily when defining $\pi_{(U,V)}$. Without loss of generality, we suppose that ρ reaches a state of U'. As a fault can only be created after ρ if π_0 switches to its third phase, for $n \geq |\rho|_o$ we have

$$\mathbb{P}_{\pi_0}(\tilde{\rho} \in \mathsf{C}_n \mid \rho \preceq \tilde{\rho}) \geq \mathbb{P}_{\pi_0}(\rho) \prod_{i=|\rho|}^{n} \frac{i}{i+1} = \mathbb{P}_{\pi_0}(\rho) \frac{|\rho|}{n+1} \ .$$

Thus, for every $0 < \alpha < 1$, similarly to $n\alpha^n$, $\frac{\alpha^n}{\mathbb{P}_{\pi_0}(\mathsf{C}_n)}$ converges to 0.

• Conversely, suppose that the initial state does not belong to PreWin. Let π be a strategy ensuring diagnosability. For every state (q, U, V) with $q \in U$ reachable by a run ρ_0 with π, $(U, V) \notin \mathsf{WinK}$ and due to one of our observations $(\{q\}, (U, V)) \notin \mathsf{WinK}_U$. Let K be the number of iterations in the fixpoint computation of WinK. Then, for every sequence of K random choices under π, there exists a faulty run $\rho \in \mathsf{F}$, compatible with these choices, starting in (q, U, V) and of observable length smaller than K. Adding up the probabilities of runs corresponding to every sequence of choices of π we obtain

$$\mathbb{P}_\pi(\rho \in \mathsf{F}_{|\rho_0|_o+K} \mid \rho_0 \preceq \rho) \geq \lambda^{K|Q|}\mathbb{P}_\pi(\rho_0)$$

where $\lambda = \min_{q' \in Q} \frac{1}{G^\Sigma(q')}$. Thus, for every $n \in \mathbb{N}, \mathbb{P}_\pi(\mathsf{C}_{n+K}) \leq \mathbb{P}_\pi(\mathsf{C}_n)(1 - \lambda^{K|Q|})$. Letting $\alpha = (1 - \lambda^{K.|Q|})^{\frac{1}{K}}$, we obtain $\lim_{n\to\infty} \frac{\alpha^n}{\mathbb{P}_\pi(\mathsf{C}_n)} > 0$, so that $\mathcal{C}_{\pi_0}$ is not strongly resilient.

To conclude the proof, we observe that the $\mathsf{EXPTIME}$-hardness derives from the same reduction as in the proof of Theorem 6.2. □

It turns out that this same combination of strategies can be used to ensure lasting fault freeness and FF-diagnosability. In fact, the following theorem establishes that the characterisation of the strongly resilient active diagnosability also applies to the lasting fault free active diagnosability.

Theorem 6.4. *Lasting fault free active diagnosability is equivalent to strongly resilient active diagnosability.*

We show here that the characterisation given in the proof of Theorem 6.3 for a CLTS to be actively diagnosable with guarantee of strong resiliency also characterises the fact that the CLTS is actively diagnosable with guarantee of lasting fault freeness. This shows the equivalence of the two notions in the active case.

Proof. We reuse the definitions from the proof of Theorem 6.3. Let us show that $\mathcal{C}$ is actively diagnosable with guarantee of lasting fault freeness iff the initial state of $\mathcal{C}^B$ belongs to PreWin.

• Suppose that the initial state belongs to PreWin. Then, as discussed in the proof of Theorem 6.3, $\mathcal{C}_{\pi_0}$ is diagnosable and there exists a finite run ρ such that $\mathbb{P}(\tilde{\rho} \in \mathsf{C}_n \mid \rho \preceq \tilde{\rho}) \geq \mathbb{P}(\rho)\frac{|\rho|}{n+1}$. Thus:

$$\sum_{n=1}^{\infty} \mathbb{P}(\mathsf{C}_n) \geq \sum_{n=|\rho|}^{\infty} \mathbb{P}(\tilde{\rho} \in \mathsf{C}_n \mid \rho \preceq \tilde{\rho}) \geq \mathbb{P}(\rho)|\rho| \sum_{n=|\rho|}^{\infty} \frac{1}{n+1} = \infty.$$

• Conversely, if the initial state does not belong to PreWin. Let π be a strategy ensuring diagnosability. For every $n \in \mathbb{N}, \mathbb{P}(\mathsf{C}_{n+K}) \leq \mathbb{P}(\mathsf{C}_n)(1 - \lambda^{K.|Q|})$. Thus:

$$\sum_{n=1}^{\infty} \mathbb{P}(\mathsf{C}_n) \leq K \sum_{n=1}^{\infty} (1 - \lambda^{K.|Q|})^n \leq K.|Q_B|.\frac{1}{\lambda^{K.|Q|}} < \infty.$$

□

Given the equivalence of strong resiliency and lasting fault freeness, from Theorem 6.3 we derive:

Corollary 6.1. *Lasting fault free active diagnosability is* EXPTIME*-complete.*

2.3 Safe active diagnosis problem under finite-memory strategies

Contrary to the other qualitative problems, safe active diagnosability is known to be undecidable [BFH$^+$14]. In order to regain decidability, one can restrict the strategies so that they only use finite memory. Note first that decidability is not immediate even if the strategies are assumed to be finite-memory, since no *a priori* bound on the memory is known[3]. This restriction was studied in [BFH$^+$14] where the authors give an NEXPTIME algorithm. However, the known lower bound is only EXPTIME, leaving a gap. We refine here this complexity result by proving that safe active diagnosis can be solved in EXPTIME when restricting to finite-memory strategies.

To do so, we prove a more general result in the context of a well-known model, quite popular in artificial intelligence and more recently in formal methods, that combines partial observation, probabilities and control, namely *Partially Observable Markov Decision Processes* (POMDP) [Å65, KLC98]. We establish that the existence of finite-memory schedulers that ensure a Büchi objective with probability 1 and a safety objective with positive probability in a POMDP is decidable in EXPTIME. We then reduce the safe active diagnosis of a CLTS $\mathcal{C}$ restricted to finite-memory strategies to the existence of a finite-memory scheduler in a POMDP $\mathsf{M}_{\mathcal{C}}$ ensuring at the same time a Büchi objective with probability 1 and a safety objective with positive probability.

Definition 6.8. *A* partially observable Markov decision process *(POMDP) is a tuple* $\mathsf{M} = \langle Q, q_0, \mathsf{Obs}, \mathsf{Act}, T\rangle$ *where*

- Q *is a finite set of states with* q_0 *the initial state;*
- $\mathsf{Obs} : Q \to \mathcal{O} \cup \{\epsilon\}$ *assigns an observation* $\mathsf{O} \in \mathcal{O}$ *to each state.*
- Act *is a finite set of actions;*
- $T : Q \times \mathsf{Act} \to \mathsf{Dist}(Q)$ *is a partial transition function. Letting* $\mathsf{Ena}(q) = \{a \in \mathsf{Act} \mid T(q,a)$ *is defined*$\}$ *the set of enabled actions in state* q*, we assume that:*
  - *for all* $q \in Q$, $\mathsf{Ena}(q) \neq \emptyset$*, and*
  - *whenever* $\mathsf{Obs}(q) = \mathsf{Obs}(q')$*, then* $\mathsf{Ena}(q) = \mathsf{Ena}(q')$ *and slightly abusing our notation, we denote by* $\mathsf{Ena}(\mathsf{O})$ *the set of events enabled in every state with observation* O*.*

A *decision rule* of a POMDP is a distribution from $\mathsf{Dist}(\mathsf{Act})$ that resolves one nondeterminism choice by randomization. A *scheduler* for a POMDP maps histories of observations to decision rules. Formally, a scheduler is a function $\tau : \mathcal{O}^+ \to \mathsf{Dist}(\mathsf{Act})$

[3]In the case of Proposition 4.9, page 120, the restriction in fact made the problem more difficult.

such that for every $\mathsf{O}_1 \cdots \mathsf{O}_i$, $\mathsf{Supp}(\tau(\mathsf{O}_1 \cdots \mathsf{O}_i)) \subseteq \mathsf{Ena}(O_i)$. Given a scheduler τ, a POMDP M yields a stochastic process. This stochastic process can be represented by an infinite state pLTS, denoted $\mathsf{M}(\tau)$ in which states are histories of observations. One denotes by $\mathbb{P}^{\mathsf{M}}_{\tau}(\mathsf{Ev})$ the probability that an infinite observed sequence of Ev is realized in this pLTS.

Similarly to what was said for strategies, one can define finite-memory schedulers for POMDP. The notion of *belief* can be adapted to POMDP. As for CLTS, it is a non-empty set of states that represents the current state estimate, *i.e.* the set of states the system may be in, given the actions (which affect the reachable set of states contrary to what is done for CLTS) and observations so far. The initial belief is $\{q_0\}$, and given a current belief B, a decision rule δ and an observation O, the belief obtained after δ has been applied and O has been observed is defined by:

$$\Delta(B, (\delta, \mathsf{O})) = \bigcup_{q \in B,\ a \in \mathsf{Supp}(\delta)} \mathsf{Supp}(T(q,a)) \cap \mathsf{Obs}^{-1}(\mathsf{O}) \ .$$

Aiming at providing a POMDP $\mathsf{M}_{\mathcal{C}}$ for the safe active diagnosis problems of a CLTS $\mathcal{C}$, we face several difficulties. First, in a CLTS the observations are related to events while in a POMDP they are related to states. As a consequence, we need to label the states by the latest observation made by the system. Secondly, our objectives are not based on states but on observed sequences. Fortunately, the relevant information pertaining to the observations, namely the information about ambiguity of observed sequences, is available in the belief. Thus (with two exceptions) the states are triples formed of a state q, an event 'a' and a belief B of the CLTS. A third adaptation concerns the control mechanism. In $\mathcal{C}$, the control is performed by choosing (possibly randomly) a subset of allowed controllable events. Thus actions of $\mathsf{M}_{\mathcal{C}}$ are subsets of events that include the uncontrollable events. Given some control decision $\Sigma^{\bullet}$, to define the transition probability of $\mathsf{M}_{\mathcal{C}}$ from (q, a, B) to (q', a', B'), one must consider all runs in $\mathcal{C}$ labelled by events of $\Sigma^{\bullet}$ from q to q' such that the last event, labelled by 'a'', is the only observable one. The probability of any such run is obtained by the product of the individual step probabilities. The latter are then defined by the normalization of weights w.r.t. $\Sigma^{\bullet}$. Finally, there cannot be infinite runs of unobservable events due to the convergence of $\mathcal{C}$. However some runs can reach, via unobservable events, a state from which no event of $\Sigma^{\bullet}$ is enabled. In other words, the control $\Sigma^{\bullet}$ applied in (q, a, B) may have a positive probability to reach a deadlock (*i.e.* the chosen decision rule leads to a strategy for the CLTS which is not live). In order to capture this behaviour and to obtain a non defective probability distribution, we add an additional state `lost`, that corresponds to such deadlocks. The next definition formalizes our approach.

Definition 6.9. *The POMDP* $\mathsf{M}_{\mathcal{C}} = \langle Q^{\mathsf{M}_{\mathcal{C}}}, q_0^{\mathsf{M}_{\mathcal{C}}}, \mathsf{Obs}, \mathsf{Act}, T^{\mathsf{M}_{\mathcal{C}}} \rangle$ *derived from a CLTS* $\mathcal{C} = \langle Q, q_0, \Sigma, T \rangle$ *is defined by:*

- $Q^{\mathsf{M}_{\mathcal{C}}} = Q \times \Sigma_o \times \mathcal{B}l_{\mathcal{C}} \uplus \{(q_0, \varepsilon, \{q_0\}), \mathbf{l}ost\}$ *with* $q_0^{\mathsf{M}_{\mathcal{C}}} = (q_0, \varepsilon, \{q_0\})$*;*
- *the set of observations is* $\mathcal{O} = \Sigma_o \cup \{\mathbf{l}ost\}$*, with* $\mathsf{Obs}(\mathbf{l}ost) = \mathbf{l}ost$ *and for* $(q, a, B) \in Q^{\mathsf{M}_{\mathcal{C}}}$*,* $\mathsf{Obs}((q, a, B)) = a$*;*

- $\mathsf{Act} = \{\Sigma^\bullet \subseteq \Sigma \mid \Sigma^\bullet \supseteq \Sigma \setminus \Sigma_c\}$;
- *for every* $(q_1, a, B) \in Q^{\mathsf{M}_\mathcal{C}}$ *and* $\Sigma^\bullet \in \mathsf{Act}$, $T^{\mathsf{M}_\mathcal{C}}((q_1, a, B), \Sigma^\bullet) = \mu \in \mathsf{Dist}(Q^{\mathsf{M}})$ *where for* $b \in \Sigma^\bullet \cap \Sigma_o$*:*
  - $\mu((q', b, \Delta(B, b))) =$

$$\sum_{\substack{q_1 \xrightarrow{a_1} q_2 \cdots \xrightarrow{a_n} q_{n+1} \xrightarrow{b} q' \\ a_1 \cdots a_n \in \Sigma^\bullet \cap \Sigma_u}} \Big(\prod_{i=1}^{n} T^{\Sigma^\bullet}(q_i, a_i, q_{i+1}) \Big) \cdot T^{\Sigma^\bullet}(q_{n+1}, b, q');$$

  - $\mu(\mathtt{lost}) = \displaystyle\sum_{\substack{q_1 \xrightarrow{a_1} q_2 \cdots \xrightarrow{a_n} q_{n+1} \\ a_1 \cdots a_n \in \Sigma^\bullet \cap \Sigma_u \\ G^{\Sigma^\bullet}(q_{n+1}) = 0}} \prod_{i=1}^{n} T^{\Sigma^\bullet}(q_i, a_i, q_{i+1});$
- *for every* $\Sigma^\bullet \in \mathsf{Act}$, $T^{\mathsf{M}_\mathcal{C}}(\mathtt{lost}, \Sigma^\bullet) = \mathbf{1}_{\mathtt{lost}}$.

Given $\mathcal{C}$, the construction of $\mathsf{M}_\mathcal{C}$, which is of size in $2^{O(|Q|+|\Sigma|)}$, can be done in exponential time. Also, the probability distributions over next states (μ in Definition 6.9) are presented as sums over runs of $\mathcal{C}$, but they can be computed in polynomial time by matrix operations.

A CLTS $\mathcal{C}$ and its associated POMDP $\mathsf{M}_\mathcal{C}$ are closely related. In particular, strategies in $\mathcal{C}$ and schedulers in $\mathsf{M}_\mathcal{C}$ are in a one-to-one correspondence. First, let us explain how to naturally derive a strategy π for $\mathcal{C}$ from a scheduler τ in $\mathsf{M}_\mathcal{C}$. For an observed sequence $a_1 \cdots a_n \in \Sigma_o^*$, we set $\pi(a_1 \cdots a_n) = \tau(a_1 \cdots a_n)$ Notice that the strategy π obtained that way is not necessarily live: for example, if after $a_1 \cdots a_n$ the choice of τ leads with positive probability to `lost`, then π is not live. However, as soon as τ ensures to avoid state `lost`, then the corresponding strategy π is live. Similarly, to a live strategy π for $\mathcal{C}$, we can associate a scheduler τ in $\mathsf{M}_\mathcal{C}$ that always avoids `lost`: given a sequence of observations that does not contain `lost`, thus of the form $a_1 \cdots a_n$, with $a_i \in \Sigma_o$ for all i, we set $\tau(a_1 \cdots a_n) = \pi(a_1 \cdots a_n)$.

Moreover, if (π, τ) is a pair of live strategy and corresponding scheduler (that always avoids `lost`), the probability measures $\mathbb{P}_{\mathcal{C}_\pi}$ and $\mathbb{P}_\tau^{\mathsf{M}_\mathcal{C}}$ are essentially equivalent. More precisely, the product in $\mathsf{M}_\mathcal{C}$ with the observation and the belief does not change the probability measure defined by $\mathcal{C}_\pi$.

We now show how to decide for POMDP the existence of a finite-memory scheduler that ensures a Büchi objective with probability one and a safety objective with positive probability. We use LTL notations to denote sets of runs in a POMDP, such as $\Diamond$, $\Box$ and $\Box\Diamond$ for eventually, always and infinitely often respectively (given a state q, $\Box\Diamond q$ thus represents the set of runs containing q infinitely often).

Theorem 6.5. *The problem whether, given a POMDP* M *with subsets of states* F *and* I*, there exists a finite-memory scheduler* τ *such that* $\mathbb{P}_\tau^{\mathsf{M}}(\Box\Diamond F) = 1$ *and* $\mathbb{P}_\tau^{\mathsf{M}}(\Box I) > 0$ *is* EXPTIME*-complete.*

Theorem 6.5 derives from Propositions 6.4 and 6.5 below, that state, respectively, the upper bound in the general case, and the lower bound in a particular case, namely for the safe active diagnosability under finite-memory strategies.

Proposition 6.4. *Given a POMDP* M *with subsets of states* F *and* I*, one can decide in* EXPTIME *whether there exists a finite-memory scheduler* τ *such that* $\mathbb{P}^{\mathsf{M}}_{\tau}(\Box\Diamond F) = 1$ *and* $\mathbb{P}^{\mathsf{M}}_{\tau}(\Box I) > 0$.

Due to the complexity of this proof, we decompose it using two lemmas. The idea is the following. We first define a set $\mathsf{Win}_{=1}$ of pairs of beliefs (B, B') with $B \subseteq B'$ such that there exists a scheduler that ensures with probability 1 to stay in I from any state of B and to reach F from any state of B'. As $B \subseteq B'$, this implies that if one starts with a distribution which support is B', there is a scheduler satisfying both the Büchi objective with probability one and the safety objective with positive probability. Such a belief B' is a "winning" belief. However, there are "winning" beliefs that cannot be obtained directly from $\mathsf{Win}_{=1}$. In Lemma 6.1, we show how to compute efficiently $\mathsf{Win}_{=1}$ through a greatest fixed point algorithm. Using $\mathsf{Win}_{=1}$, we then build a set of beliefs Win, which contains intuitively the beliefs from which there exists a scheduler that can reach a belief that corresponds to the second component of a pair in $\mathsf{Win}_{=1}$ with positive probability while never reaching a "losing" belief (a belief from which we cannot satisfy the Büchi requirement). Finally, in Lemma 6.2 we show that Win contains exactly the set of "winning" beliefs. Thus, there exists a scheduler satisfying the two objectives iff the initial belief $\{q_0\}$ belongs to Win.

Proof. In this proof, the POMDP $\mathsf{M} = \langle Q, q_0, \mathsf{Obs}, \mathsf{Act}, T\rangle$ is fixed, and we use notation $\mathbb{P}^{\delta_0}_{\tau}(\mathsf{Ev})$ to denote the probability of Ev under scheduler τ assuming that instead of q_0, the initial state in M is given by the distribution $\delta_0 \in \mathsf{Dist}(Q)$.

Let us first explain how to compute the following set of pairs of beliefs:

$$\begin{aligned}\mathsf{Win}_{=1} = \{(B', B) \mid B' \subseteq I,\ B' \subseteq B \text{ s.t. } \exists \tau \text{ s.t.}\\ \forall \delta_0 \text{ with } \mathsf{Supp}(\delta_0) = B,\ \mathbb{P}^{\delta_0}_{\tau}(\Box\Diamond F) = 1, \text{ and}\\ \forall \delta'_0 \text{ with } \mathsf{Supp}(\delta'_0) = B',\ \mathbb{P}^{\delta'_0}_{\tau}(\Box I) = 1\}\ .\end{aligned}$$

Intuitively, $\mathsf{Win}_{=1}$ denotes pairs of beliefs such that there exists a scheduler that ensures a Büchi objective almost-surely from the larger belief, and a safety objective almost-surely from the smaller one. Note that, (1) in the definition of $\mathsf{Win}_{=1}$, we do not require the scheduler τ to be finite-memory and (2) as schedulers associate a decision rule to every sequence of observation, the same choices are taken after the same sequence of observations for the Büchi and the safety objective although the initial distribution differs. Given that we consider pairs of beliefs, we introduce the following notation: $\Delta((B', B), \mathsf{O}_1) = \big(\Delta(B', \mathsf{O}_1), \Delta(B, \mathsf{O}_1)\big)$, and similarly for sequences of actions and observations. Also, for $X \subseteq Q$ a subset of states, we denote by $\mathcal{B}l_{\subseteq X} = \{B \in \mathcal{B}l \mid B \subseteq X\}$ the set of beliefs contained in X.

We now show how to efficiently compute $\mathsf{Win}_{=1}$.

Lemma 6.1. *Let* Win_∞ *be the greatest fixed point starting from* $\{(q, B', B) \in Q \times \mathcal{B}l \times \mathcal{B}l \mid q \in B, B' \subseteq B,\ B' \subseteq I\}$ *of the following operator:*

$$
\begin{aligned}
W \mapsto \{(q, B'_1, B_1) \mid \exists n \geq 1,\ \exists q_0 \dots q_n \in Q,\ \exists \alpha_1, \cdots \alpha_n \exists \mathsf{O}_1 \cdots \mathsf{O}_n,\\
(B'_2, B_2) = \Delta\big((B'_1, B_1), (\alpha_1, \mathsf{O}_1) \cdots (\alpha_n, \mathsf{O}_n)\big), \forall q' \in B_2, (q', B'_2, B_2) \in W,\\
q_0 = q,\ q_n \in F,\ \forall i < n, T(q_i, \alpha_{i+1})(q_{i+1}) > 0, \forall 1 \leq j \leq n, \mathsf{Obs}(q_j) = \mathsf{O}_j,\\
\forall i \leq n,\ \forall \mathsf{O}'_i,\ \ \textit{for } (B'_3, B_3) = \Delta\big((B', B), (\alpha_1, \mathsf{O}_1) \cdots (\alpha_{i-1}, \mathsf{O}_{i-1})(\alpha_i, \mathsf{O}'_i)\big)\\
\textit{we have } \forall q' \in B_3,\ (q, B'_3, B_3) \subseteq W \cap Q \times \mathcal{B}l_{\subseteq I} \times \mathcal{B}l\} \ .
\end{aligned}
$$

We have $\mathsf{Win}_{=1} = \{(B', B) \mid \forall q \in B,\ (q, B', B) \in \mathsf{Win}_\infty\}$.

Proof of Lemma 6.1. To establish that $\mathsf{Win}_{=1}$ corresponds to the projection on the pair of beliefs of Win_∞, we first assume that for all $q \in B$, (q, B', B) belongs to Win_∞, and exhibit a scheduler τ that witnesses $(B', B) \in \mathsf{Win}_{=1}$. Let us define τ as follows. The scheduler τ has finite memory $\mathcal{B}l \times \mathcal{B}l$. From memory state (B', B), τ dictates to play uniformly all actions α such that for every observation O and every $q \in \Delta(B, \alpha, \mathsf{O})$, we have $(q, \Delta((B', B), \alpha, \mathsf{O})) \in \mathsf{Win}_\infty$. Note that this set of "safe" actions is necessarily non empty because $(q, B', B) \in \mathsf{Win}_\infty$. If α is played, and O is observed, the memory state of τ is updated to $\Delta((B', B), \alpha, \mathsf{O})$, which is still in Win_∞, by assumption on α. The scheduler τ then continues similarly with memory state $\Delta((B', B), \alpha, \mathsf{O})$. So defined, let us show that τ witnesses $(B', B) \in \mathsf{Win}_{=1}$. First, let δ_0 be a distribution with support B. The scheduler τ ensures to stay (surely) in Win_∞. Moreover, for every $q \in B$, with a positive probability, say $p_{(q,B',B)} > 0$, the sequence $(\alpha_1, \mathsf{O}_1) \cdots (\alpha_n, \mathsf{O}_n)$ of actions and observations leading to F that derives from the fixpoint definition, happens from q. There are finitely many $p_{(q,B',B)}$, all are positive, so they are lower bounded by some positive value p. Playing τ forever thus ensures visiting F almost surely, and iterating this reasoning, even visiting F infinitely often with probability 1. Now, assuming $B' \neq \emptyset$ let δ'_0 be a distribution with support B'. Any action picked by τ ensures that, whatever the observation, the first belief-component remains in I. Therefore, surely, from distribution δ'_0 the plays stay in the invariant I.

Let us now assume that the triplet (q, B', B) is removed during the iterative computation of the fixed point W_∞. We prove, by induction on k, that if (q, B', B) is removed at iteration k, then, $(B', B) \notin \mathsf{Win}_{=1}$. If $k = 0$, the pair is removed at initialization, hence $B' \not\subseteq I$ or $B' \not\subseteq B$, and obviously $(B', B) \notin \mathsf{Win}_{=1}$. Otherwise it happens at the k-th iteration, for some $k \geq 1$. Assume, towards a contradiction, that there exists a scheduler τ, witnessing that $(B', B) \in \mathsf{Win}_{=1}$. In particular, there exists a sequence of pairs of actions and observations allowed by the scheduler $(\alpha_1, \mathsf{O}_1) \cdots (\alpha_n, \mathsf{O}_n)$ so that there exists $q_0 \dots q_n \in Q$ with $q_0 = q$, $q_n \in F$, $\forall i < n, T(q_i, \alpha_{i+1})(q_{i+1}) > 0, \forall 1 \leq j \leq n$ and $\mathsf{Obs}(q_j) = \mathsf{O}_j$. Because the triple (q, B', B) was removed at iteration k, it must be that, either (1) for $(B'_2, B_2) = \Delta\big((B', B), (\alpha_1, \mathsf{O}_1) \cdots (\alpha_n, \mathsf{O}_n)\big)$, there exists $q_2 \in B_2$ such that $(q_2, B', B) \notin W_{k-1}$, (2) no run corresponding to a sequence $(\alpha_1, \mathsf{O}_1) \cdots (\alpha_n, \mathsf{O}_n)$ satisfying (1) and starting in q ends in F or (3) there exists an index i and an observation O'_i such that for $(B'_3, B_3) = \Delta\big((B', B), (\alpha_1, \mathsf{O}_1) \cdots (\alpha_{i-1}, \mathsf{O}_{i-1})(\alpha_i, \mathsf{O}'_i)\big)$ there exists $q \in B_3$, $(q, B'_3, B_3) \notin W_{k-1} \cap Q \times \mathcal{B}l_{\subseteq I} \times \mathcal{B}l$. In the first case, it means that

there is a positive probability, under τ to reach a pair of beliefs out of W_{k-1}, and thus out of $\mathsf{Win}_{=1}$ by induction hypothesis. As the sequence of action and observations was chosen so that one can reach F from q, the second case implies that the first case holds with our selected sequence of actions and observations. For the third case, let $(B'_3, B_3) = \Delta\big((B', B), (\alpha_1, \mathsf{O}_1) \cdots (\alpha_{i-1}, \mathsf{O}_{i-1})(\alpha_i, \mathsf{O}'_i)\big)$. Either there exists $q' \in B_3$ such that $(q, B'_3, B_3) \notin W_{k-1}$, then it is treated similarly to the first case. Else $B'_3 \notin \mathcal{B}l_{\subseteq I}$. Observe that, in this case, the second requirement on τ is not satisfied since $\mathbb{P}^{\delta'_0}_{\pi}(\Box I) < 1$. □

Thanks to Lemma 6.1, $\mathsf{Win}_{=1}$ can be computed in EXPTIME. Let us now define Lose as the set of beliefs that are clearly losing:

$$\mathsf{Lose} = \{B \in \mathcal{B}l \mid \neg\exists\tau\, \forall\delta_0 \text{ with } \mathsf{Supp}(\delta_0) = B, \mathbb{P}^{\delta_0}_{\tau}(\Box\Diamond F) = 1\}\ .$$

As established *e.g.* in [BGG09] in the more general framework of 2-player stochastic games with signals, Lose can also be computed in EXPTIME.

Informally, we now consider the set of beliefs from which one can reach, while staying in I, and not risking to fall in Lose, some belief B such that there exists $B' \neq \emptyset$ with $(B', B) \in \mathsf{Win}_{=1}$. In order to easily represent what staying in I means, we assume without loss of generality that the set of states $Q \setminus I$ is absorbing[4]. Formally, let Win be the following set of beliefs:

$$\begin{aligned}\mathsf{Win} = \{B_0 \in \mathcal{B}l \mid \exists (B', B) \in \mathsf{Win}_{=1} \text{ s.t. } B' \neq \emptyset \text{ and} \\ \exists\alpha_1 \cdots \alpha_n,\ \exists\mathsf{O}_1 \cdots \mathsf{O}_n,\ \Delta\big(B_0, (\alpha_1, \mathsf{O}_1) \cdots (\alpha_n, \mathsf{O}_n)\big) = B \\ \forall i \leq n,\ \forall\mathsf{O}'_i,\ \Delta\big(B_0, (\alpha_1, \mathsf{O}_1) \cdots (\alpha_{i-1}, \mathsf{O}_{i-1})(\alpha_i, \mathsf{O}'_i)\big) \notin \mathsf{Lose}\}.\end{aligned}$$

The set Win characterizes winning beliefs, that is, beliefs from which there exists a finite-memory scheduler (called a winning scheduler) ensuring at the same time, the Büchi objective $\Box\Diamond F$ almost-surely, and the safety objective $\Box I$ with positive probability. Formally:

Lemma 6.2. *$B_0 \in \mathsf{Win}$ if and only if for every δ_0 with $\mathsf{Supp}(\delta_0) = B_0$, there exists a finite-memory scheduler τ such that $\mathbb{P}^{\delta_0}_{\tau}(\Box\Diamond F) = 1$ and $\mathbb{P}^{\delta_0}_{\tau}(\Box I) > 0$.*

Proof of Lemma 6.2. Assume first that $B_0 \in \mathsf{Win}$. We design a finite memory scheduler τ that is winning from any initial distribution δ_0 with support B_0. In a first mode, τ aims at reaching a pair of beliefs $(B', B) \in \mathsf{Win}_{=1}$ from B_0. More precisely, τ plays the sequence of actions that leads with positive probability from B_0 to some $B \in \mathcal{B}l$ such that there exists $B' \neq \emptyset$ with $(B', B) \in \mathsf{Win}_{=1}$. If this succeeds, τ then switches to another mode, where it behaves as the winning scheduler that starts from (B', B) in Lemma 6.1. If it fails, the play ends in a belief $B_1 \notin \mathsf{Lose}$ (by definition of Win), and from there τ plays to ensure visiting F infinitely often with probability 1. All in all, τ ensures almost surely visiting F infinitely often, and with positive probability (the

[4]This can be ensured similarly to what was done for the set Q_f in Subsection 1.4 of Chapter 2, page 43.

probability of the prefix leading to B, times the probability that the play is in B' at that time point) to stay in I. Note that the size of the memory τ uses is in $O(|\mathcal{B}l|^2)$. Indeed, in its first phase, it tries to reach a belief by using a set of actions of length smaller than $|\mathcal{B}l|$ as it does not need to visit the same belief twice. If it fails to reach the target belief, then ensuring the Büchi requirement can be done with a belief-based scheduler, *i.e.* a scheduler that only remembers the current belief, thus with memory of size $|\mathcal{B}l|$. If it reaches its target however, it needs to remember pairs of beliefs as done in Lemma 6.1, thus requires a memory of size $|\mathcal{B}l|^2$.

Let now δ_0 be an initial distribution with support B_0, and assume that there exists a finite-memory scheduler τ such that $\mathbb{P}^{\delta_0}_\tau(\Box\Diamond F) = 1$ and $\mathbb{P}^{\delta_0}_\tau(\Box I) > 0$. We consider $\mathcal{M}(\tau)$ the pLTS generated by τ, with finite state space $Q \times \mathsf{Mem}$, where Mem is a finite set of memory states. Without loss of generality, we iteratively tag each state of $\mathcal{M}(\tau)$ with its associated belief. Since τ is winning and almost surely a run reach a BSCC, there must exist a BSCC $\mathcal{C}$ in $\mathcal{M}_\tau$, reachable from some (q_0, m_0) via an I-run ρ (a run where all state are included in I), and such that all states $(q, m) \in \mathcal{C}$ satisfy $q \in I$, and there exists a state $(q_f, m_f) \in \mathcal{C}$ such that $q_f \in F$. Let $(q, m) \in \mathcal{C}$ be the state reached by run ρ, B be the belief obtained after observing ρ. From (q, m), under scheduler τ, all plays stay in I. Moreover, for any $q' \in B$, from (q', m), under scheduler τ, almost all runs visit F infinitely often. As a consequence, by the definition of $\mathsf{Win}_{=1}$, $(\{q\}, B) \in \mathsf{Win}_{=1}$. Then, we conclude that $B_0 \in \mathsf{Win}$, exploiting the I-run ρ, and the fact that τ ensures $\Box\Diamond F$ almost-surely, and thus always avoids Lose. □

Win characterizes the winning beliefs, and can be computed in $\mathsf{EXPTIME}$. We thus showed the computability in $\mathsf{EXPTIME}$ of the set of supports B from which for every distribution δ_0 with $\mathsf{Supp}(\delta_0) = B$ there exists a finite-memory scheduler τ such that $\mathbb{P}^{\delta_0}_\tau(\Box\Diamond F) = 1$ and $\mathbb{P}^{\delta_0}_\tau(\Box I) > 0$. □

Now the safe active diagnosis restricted to finite-memory strategies can be reduced to the existence for POMDP of a finite-memory scheduler that ensures a Büchi objective almost surely, and a safety objective with positive probability. As $\mathsf{M}_\mathcal{C}$ is exponential in the size of $\mathcal{C}$ and the algorithm on the POMDP is in $\mathsf{EXPTIME}$, we obtain a 2$\mathsf{EXPTIME}$ complexity upper-bound. Fortunately, in order to avoid a doubly exponential blowup and to establish the $\mathsf{EXPTIME}$ complexity, we observe that the exponential comes in both cases from the computation of beliefs depending *only* on the original CLTS. This implies that the safe active probabilistic diagnosis problem is in $\mathsf{EXPTIME}$ when restricted to finite-memory strategies.

Corollary 6.2. *The safe active diagnosis problem restricted to finite-memory strategies is decidable in* $\mathsf{EXPTIME}$.

Proof. Given a CLTS $\mathcal{C}$, we build $\mathsf{M}_\mathcal{C}$ and decide if there exists a scheduler τ ensuring $\mathbb{P}^{\delta_0}_\tau(\Box\Diamond F) = 1$ and $\mathbb{P}^{\delta_0}_\tau(\Box I) > 0$ with $I = \{(q, a, B) \mid q \in Q_c\}$ and $F = \{(q, a, B) \mid B \subseteq Q_f \vee q \in Q_c\}$ and δ_0 is the Dirac distribution of support $\{q_0^{\mathsf{M}_\mathcal{C}}\}$. Due to the link between $\mathsf{M}_\mathcal{C}$ and $\mathcal{C}$, this choice of F corresponds to runs that are either correct or surely faulty in $\mathcal{C}$ and this choice of I corresponds to runs that are correct. Thus there exists a

finite-memory scheduler τ as defined above iff the corresponding strategy π in $\mathcal{C}$ ensures safe active diagnosis. Moreover, as explained above the corollary, deciding the existence of this scheduler can be done in EXPTIME. □

A matching lower-bound is already known from the literature:

Proposition 6.5 ([BFH+14])**.** *The safe active diagnosis problem restricted to finite-memory strategies is* EXPTIME*-hard.*

Obviously, this lower bound also holds for the more general problem: on POMDP, whether there exists a finite-memory strategy ensuring a Büchi objective almost-surely and a safety objective with positive probability.

3 Conclusion

Degradation of a controllable probabilistic system combines two objectives. The system must satisfy at the same time a diagnosability and a degradation condition. Interestingly and as shown first in [BFH+14], having to satisfy both conditions at the same time increases the difficulty: safe active diagnosability combines two decidable problems yet ends up being undecidable. In order to regain decidability, we introduced two new degradation notions both in a qualitative and a quantitative ways. While the quantitative versions are undecidable, the qualitative ones brings interesting results. Indeed, on the one hand, they are close to safe active diagnosability as two of the notions are equivalent to it for finite pLTS. On the other hand, they are decidable in EXPTIME. As EXPTIME is the lower bound of the complexity of active diagnosability, it is unsurprisingly also a lower bound of the complexity of the combination of active diagnosability and of a degradation condition. Therefore we can test the combination of active diagnosability with a degradation condition without reaching a new complexity class.

This analysis however can result in diagnosers requiring infinite memory. When restricted to finite-memory controllers, many differences appear. First, as the pLTS obtained by controlling a CLTS with a finite-memory strategy is finite, then according to Theorem 6.1, safety, strong resiliency and lasting fault freeness are equivalent. Studying the safe active diagnosability, we showed it to be EXPTIME-complete. Thus, the restriction to finite memory helped regain decidability. For weakly resilient active diagnosability, the restriction to finite memory is not necessary as, as shown in Theorem 6.2, if the system is weakly resilient active diagnosable, there exists a strategy with finite memory.

The notions of degradation introduced here were inspired from the notion of safe active diagnosability. One could be interested in other notions of degradations representing different forms of failures within the system. For example, in our framework, the notion of faulty run is a Boolean one; once a fault occurred, the run is faulty. The fault is thus seen as a definitive and complete damage of the system. But a fault could only represent a small degradation of the system which would still be partially available. In this alternative framework, the degradation to be evaluated would be the evolution

of the number of faults in a run w.r.t. its length. Another possible direction of research would be to take a more general approach to the notion of combined objectives. It would be interesting to determine which pairs of objectives can be studied separately and which ones lead to undecidability.

Chapter 7

Opacity

This thesis consists in a study of the control of the information in probabilistic systems, mainly focusing on diagnosis. While the goal of diagnosis is to analyse an observation in order to reveal a hidden information (the fault), one could be interested in asking a dual question: can we limit the amount of hidden information that is revealed by the system. This question belongs to an important domain of partial observation issues called opacity. While the two notions can appear similar, the motivations behind them are different. Diagnosability is a notion of safety of a system while opacity is one of security. This difference in motivations implies that the questions asked for opacity are not be the same as the ones asked for diagnosis. Moreover, the model itself can differ in order to possess different properties. In this introduction, we first present informally opacity and some questions related to this domain of research: in the passive framework then in the active one. Finally, we discuss the form of control used in the active opacity framework, emphasising the differences with active diagnosability.

Opacity problems for passive systems. Given a set of secret runs, a run discloses the secret if every run with the same observed sequence is secret. With this definition, the disclosure of the secret is akin to exact diagnosability. One can also define a notion of disclosure that would resemble approximate diagnosability: for $\varepsilon > 0$, a secret run ε-discloses the secret if the probability of runs with the same observed sequence that are not secret conditioned on the probability of the observed sequence is at most ε. For non-probabilistic systems, opacity boils down to detecting if there exist a run disclosing the secret to an observer. For probabilistic systems, we are interested in quantifying the opacity of the system [BMS15, SH14, BKM12]. For instance we would want to determine if the measure of disclosing runs of a given length is positive, if it is above a certain threshold, and how this measure evolves with the observed length of the runs. This precise quantification of the measure of disclosing runs, called *disclosure*, does not appear in the diagnosability notions we studied. Indeed, a fault was considered a dangerous event which cannot, in any case, be missed. For the disclosing of the secret however, it is more usual to tolerate that part of the secret may be leaked, as long as it is a limited amount.

Opacity problems for active systems. The focus on disclosure takes a whole new meaning when considering active systems. Indeed, a controller will have an effect on the system, and depending on its choices, this disclosure will increase or decrease. Existing works study the case where the controller maximises the disclosure [BCS15, BKMS18]. This corresponds to a worst-case analysis of the system, *i.e.* for the worst possible control. This kind of analysis aims at representing the case where the control is done by an attacker that is observing the system. This control can be obtained by the attacker by using a virus for example. The opposite direction, where the control tries to minimise the disclosure is also worthy of analysis. Indeed, for example, if a system has been designed in order to satisfy a specification, yet there are still some liberties within the system, some choices that are possible and that do not affect the specification, then these choices can be made in order to optimise opacity of the system. In this case, the control is realised during the design of the system and is made in order to minimise the disclosure of the system. Thus, both maximisation and minimisation corresponds to real issues.

Formalisation of the control. When we studied active diagnosis in Chapter 6, the control was exterior to the system: from the observations it received, it was able to stop some controllable actions from occurring. The controller and the observer had the same information and thus could be thought as the same mechanism. In the examples given earlier (the virus for maximisation and the system design for the minimisation), the control comes from within the system. There is thus a clear separation between the controller and the observer/attacker. To formalise this opposition, the control is realised with a full knowledge of the system: it knows what is the exact run that is followed and especially what is the current state and make its decision based on this. However, as the attacker is not himself within the system a run only discloses the secret with its observation. In other words, the controller will try to minimise or maximise a set of runs satisfying a condition that is based on their observations. Moreover, we assume the attacker is aware of how the controller makes its choices. Indeed, the security of the system should not be based on the black box hypothesis (that an attacker is lacking information). Especially when considering cases such as the virus example: the virus could very well have been implanted in the system by the attacker, ensuring he is aware of how the virus works.

In Section 1, we establish the specifications and important questions of opacity that we consider throughout the chapter. These definitions present two different horizons over which to consider opacity: a given fixed horizon and an unbounded yet finite horizon. In Section 2 and Section 3, we study opacity over finite horizon for maximisation and minimisation respectively. These two sections echo one another emphasising the differences between the two. Finally in Section 4 we detail our results for opacity over fixed horizon for both maximisation and minimisation.

This chapter develops and extends some of the results from [BHL17a].

1 Specification for Opacity

The notion of opacity is very similar to the one of diagnosis. They both consider the information revealed by runs of partially observable systems. The difference is that the goal of diagnosis is to reveal an information about the current state of the system, while opacity tries to hide an information. Despite these similarities, the framework of opacity contains important differences with the one of diagnosis. We first define formally the framework of opacity for passive systems (Subsection 1.1), then extend the definitions to allow a control of the system (Subsection 1.2).

1.1 Opacity for Markov chains

Labelled Markov chains, as introduced in Chapter 4, are pLTS where every event is observable. We now define another kind of Markov chains called *observable Markov chains* which are pLTS where the observation is associated with the state instead of the transition. This labelling of states thus describes what an external observer can see and is given by an *observation function*[1]. We use this new framework as, while diagnosis aimed to detect an event (the fault), opacity consists in hiding that the system is currently in a secret behaviour represented by its state. This could however easily be translated as the detection of a transition triggered when entering a secret state. In fact, the equivalence of associating events with transitions or with states is a folk result. Figure 7.1 gives the informal idea of how to push events from states to transitions and Theorem 7.1 starts by a modification of the system that is close to the usual transformation allowing to associate the labels with the states rather than with the transitions.

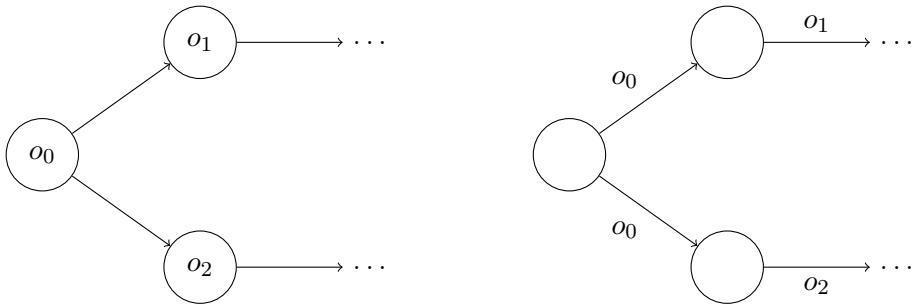


Figure 7.1: Pushing observations from states to transitions.

Definition 7.1. *An* observable Markov chain *(OMC) over alphabet Σ_o is a tuple $\mathcal{M} = (S, p, \mathsf{O})$ where S is a countable set of states, $p : S \to \mathsf{Dist}(S)$ is the transition function, and $\mathsf{O} : S \to \Sigma_o \cup \{\varepsilon\}$ is the observation function.*

We write $p(s'|s)$ instead of $p(s)(s')$ to emphasise the fact that the probability of going to state s' is conditioned by being in state s. Given a distribution μ_0 on S, we

[1]The equivalent of the observation function for pLTS was called mask function.

denote by $\mathcal{M}(\mu_0)$ the Markov chain with initial distribution μ_0. The definitions of runs, observed sequences, probability measure, ... can easily be adapted to OMC. To give an example, an infinite run of $\mathcal{M}(\mu_0)$ is a sequence of states $\rho = s_0 s_1 \ldots \in S^\omega$ such that $\mu_0(s_0) > 0$ and for each $i \geq 0$, $p(s_{i+1}|s_i) > 0$. The observed sequence of this infinite run is $\mathsf{O}(\rho) = \mathsf{O}(s_0)\mathsf{O}(s_1)... \in \Sigma_o^\infty$. The observation function is called *non erasing* if $\mathsf{O}(S) \subseteq \Sigma$ (all states are visible).

As in opacity one aims to hide a secret behaviour of the system, a way to represent what is secret is needed. There are various ways to define it depending on if we want the secret to be permanent, intermittent, described within the system... We consider here the case where the secret is permanent and given by a subset of states Sec $\subseteq S$ of the model: a (finite of infinite) run $s_0 s_1 \ldots$ is *secret* if $s_i \in$ Sec for some i, otherwise it is *public*. Under this choice, the secret itself behaves very similarly to the fault. As for pLTS where we could make the partition between faulty and correct states without loss of generality, we assume here that the set of secret states Sec is absorbing. To show this can be done without loss of generality, a new Markov chain $\mathcal{M}' = (S', p', \mathsf{O}')$ is defined from $\mathcal{M}$ by: $S' = (S \times \{0,1\})$, where $(s,0)$ represents state s where the secret has not been visited while $(s,1)$ represents the opposite situation. The transitions are then duplicated accordingly: (1) $p'((s',i)|(s,i)) = p(s'|s)$ for all $s \in S, s' \in S \setminus$ Sec, and $i \in \{0,1\}$, (2) $p'((s',1)|(s,i)) = p(s'|s)$ for all $s \in S, s' \in$ Sec, and $i \in \{0,1\}$. The new observation function is defined by $\mathsf{O}'((s,i)) = \mathsf{O}(s)$ for all $s \in S$ and $i = 0,1$ and the new set of secrets is $S \times \{1\}$. There is a one-to-one probability-preserving correspondence between the runs in $\mathcal{M}$ and the ones in $\mathcal{M}'$.

Example 7.1. *Consider the OMC of Figure 7.2 with initial distribution* $\mathbf{1}_{q_0}$*. The observation associated with a state by the observation function is displayed next to it. The secret state is shaded. Assuming* $o_1 \neq \varepsilon$ *and* $o_2 \neq \varepsilon$*, every state is associated with an observation different than* ε*, therefore the observation function is non-erasing.*

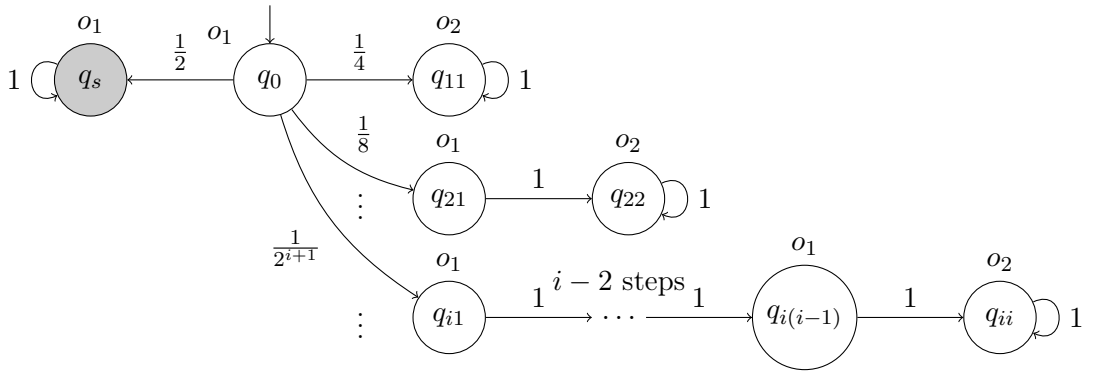


Figure 7.2: An infinitely-branching OMC with Sec $= \{q_s\}$.

One quantitative way to define the disclosure of a system is to consider that a

run discloses the secret if the probability that the current run belongs to the secret, conditioned over the observation of the run, is greater than some given threshold $\varepsilon > 0$.

Definition 7.2. *Given an OMC $\mathcal{M} = (S, p, \mathsf{O})$, an initial distribution μ_0,* Sec $\subseteq S$ *and an observation $w \in \Sigma^*$, the* proportion of secret runs *with observation w is:*

$$\mathsf{Psec}_{\mathcal{M}(\mu_0)}(w) = \frac{\mathbb{P}_{\mathcal{M}(\mu_0)}(\{\rho \in \mathsf{O}^{-1}(w) \mid \rho \text{ is secret}\})}{\mathbb{P}_{\mathcal{M}(\mu_0)}(w)}.$$

For $\varepsilon > 0$, w is ε-min-disclosing *if $\mathsf{Psec}_{\mathcal{M}(\mu_0)}(w) > 1 - \varepsilon$ and no prefix of w satisfies this inequality. Writing $D^{\varepsilon}_{\min}$ for the set of ε-min-disclosing observations, the* ε-disclosure *is defined by* $\mathrm{Disc}^{\varepsilon}(\mathcal{M}(\mu_0)) = \sum_{w \in D^{\varepsilon}_{\min}} \mathbb{P}_{\mathcal{M}(\mu_0)}(w)$. *The* positive ε-disclosure problem *consists in deciding whether* $\mathrm{Disc}^{\varepsilon}(\mathcal{M}(\mu_0)) > 0$.

To establish a parallel with diagnosability, positive ε-disclosure is similar to $\varepsilon\mathsf{FF}$-diagnosability. Indeed, in $\varepsilon\mathsf{FF}$-diagnosability (resp. the positive ε-disclosure problem) one considers the fault (resp. secret) revealed if the likelihood of the fault (resp. secret) conditioned on the observation is above $1 - \varepsilon$. The difference is that for opacity, we ask whether the measure of the set of runs disclosing the secret is positive while for diagnosability, we require this probability to be equal to the probability of faulty runs. In other words, considering a run to be faulty iff it is secret, the system is $\varepsilon\mathsf{FF}$-diagnosable iff $\mathrm{Disc}^{\varepsilon}(\mathcal{M}(\mu_0)) = \mathbb{P}(\mathsf{F}_{\infty})$.

In this chapter, we aim at studying active notions of opacity. While being the most realistic notion of probabilistic disclosure, ε-disclosure is unfortunately a too complex notion. Indeed, the problem is already undecidable for OMC:

Theorem 7.1. *The positive ε-disclosure problem is undecidable for OMC.*

To establish this undecidability result, we reduce the emptiness problem for probabilistic automata. The reduction itself is pretty straightforward. Note however that the reduction requires to first translate the PA into an OMC such that the probability of acceptance of a word w in the PA is equal to the probability to end in a secret state in the Markov chain knowing that the run has observed sequence w.

The emptiness problem and the value 1 problem[2] are undecidable for PA already with a two-letter alphabet [Paz71, GO10]. Hence in the various reductions we use the alphabet $\{a, b\}$.

Proof. Given a PA $A = (Q, q_0, \{a, b\}, (\mathbf{P}_a, \mathbf{P}_b), F)$ that we suppose complete without loss of generality, we first transform A into an incomplete OMC $\hat{A}$ where $\{a, b, \flat\}$ is the observation alphabet (an illustration is given in Figure 7.3). The set of states is $\hat{Q} = Q \cup \{q_{\text{tag}} \mid q \in Q \wedge \text{tag} \in \{a, b\}\}$, with initial distribution $\mathbf{1}_{q_0}$. The observation function $\hat{\mathsf{O}}$ is defined by $\hat{\mathsf{O}}(q) = \flat$ and $\hat{\mathsf{O}}(q_c) = c$ for $q \in Q$ and $c \in \{a, b\}$. The transition function $\hat{p}$ is defined for $q, q' \in Q$ and $c \in \{a, b\}$ by $\hat{p}(q' \mid q_c) = \mathbf{P}_c(q, q')$ and $\hat{p}(q_c \mid q) = \frac{1}{4}$. This OMC is incomplete as for every state $q \in Q$, the sum of the probabilities exiting q in $\hat{A}$ is 1/2.

[2]These notions were defined before Theorem 4.3, page 115

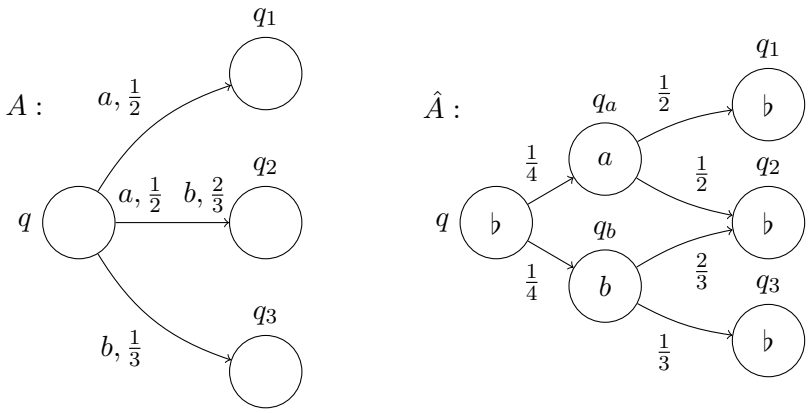


Figure 7.3: From PA A to incomplete OMC $\hat{A}$.

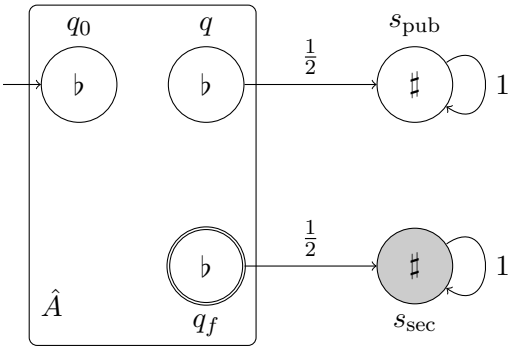


Figure 7.4: Reduction to the positive ε-disclosure problem.

We now build the OMC $\mathcal{M}_A = (S, p, \mathsf{O})$ over alphabet $\{a, b, \flat, \sharp\}$ by adding two states to complete $\hat{A}$ (see Figure 7.4 where the doubly circled state q_f is a final state of A):

- $S = \{s_{\text{pub}}, s_{\text{sec}}\} \cup \hat{Q}$, with $\text{Sec} = \{s_{\text{sec}}\}$;
- The function p is obtained from $\hat{p}$ by adding the transitions: For every $q \in F$, $p(s_{\text{sec}} \mid q) = \frac{1}{2}$, for every $q \in Q \setminus F$, $p(s_{\text{pub}} \mid q) = \frac{1}{2}$, and $p(s_{\text{pub}} \mid s_{\text{pub}}) = p(s_{\text{sec}} \mid s_{\text{sec}}) = 1$;
- O extends $\hat{\mathsf{O}}$ by $\mathsf{O}(s_{\text{sec}}) = \mathsf{O}(s_{\text{pub}}) = \sharp$.

We now prove that, given $\varepsilon \in]0, 1[$, A accepts a word with probability strictly greater than $1 - \varepsilon$ iff $\text{Disc}^{\varepsilon}(\mathcal{M}(\mu_0)) > 0$. First assume that there exists a word $w = a_1 \dots a_n \in \{a, b\}^*$ with $\mathbf{P}_A(w) > 1 - \varepsilon$. Then w corresponds to a non secret run with observed sequence $\hat{w} = \flat a_1 \flat \dots a_n \flat$ in $\mathcal{M}_A$ and $\mathsf{Psec}_{\mathcal{M}(\mu_0)}(\hat{w}\sharp) = \mathbf{P}_A(w) > 1 - \varepsilon$, which implies $\text{Disc}^{\varepsilon}(\mathcal{M}(\mu_0)) > 0$.

Conversely, if $\mathrm{Disc}^{\varepsilon}(\mathcal{M}(\mu_0)) > 0$, then there exists an observation w' in $(\{a, b, \flat, \sharp\})^*$ such that $\mathsf{Psec}_{\mathcal{M}(\mu_0)}(w') > 1-\varepsilon$. In this case, w' is of the form $\flat a_1 \flat \dots a_n \flat \sharp\sharp^*$ where, letting $w = a_1 \dots a_n$, we have $\mathsf{Psec}_{\mathcal{M}(\mu_0)}(w') = \mathsf{Psec}_{\mathcal{M}(\mu_0)}(\flat a_1 \flat \dots a_n \flat \sharp) = \mathbf{P}_A(w)$. Therefore $\mathcal{L}_{>1-\varepsilon}(A)$ is not empty. □

This undecidability result leads us to consider the simpler case where the disclosure is the probability of the set of runs surely leaking the secret, *i.e.*, such that *all* runs with the same observation are secret. One such disclosure notion, the ω-disclosure (used in [BCS15, BMS15, BKMS16]), was defined for a Markov chain $\mathcal{M} = (S, p, \mathsf{O})$ with initial distribution μ_0 by considering a measurable set of secret runs $\mathsf{SecRuns} \subseteq \Omega^{\mathcal{M}(\mu_0)}$. In our context, as mentioned earlier, $\mathsf{SecRuns}$ is $\mathsf{Reach}(\mathrm{Sec})$, the set of infinite runs visiting a state from Sec. Moreover an infinite observation $w \in \Sigma^\omega$ discloses the secret if all runs $\rho \in \mathsf{O}^{-1}(w)$ are secret. Setting $\overline{\mathsf{SecRuns}} = \Omega^{\mathcal{M}(\mu_0)} \setminus \mathsf{SecRuns}$, we define:

Definition 7.3. *For an OMC $\mathcal{M} = (S, p, \mathsf{O})$, an initial distribution μ_0 and a subset* $\mathrm{Sec} \subseteq S$*, with* $\mathsf{SecRuns} = \mathsf{Reach}(\mathrm{Sec})$*, the* ω-disclosure *of* Sec *in $\mathcal{M}$ is:*
$\mathrm{Disc}_\omega(\mathcal{M}(\mu_0)) = \mathbb{P}_{\mathcal{M}(\mu_0)}(\mathsf{SecRuns} \setminus \mathsf{O}^{-1}(\mathsf{O}(\overline{\mathsf{SecRuns}})))$.

The downside of this definition is that it only considers infinite observed sequences. In reality, an attacker will only have access to finite observed sequences before having to deduce if the system is in a secret state. To obtain measures directly related to the finite observation of a potential attacker, we assume that $\mathcal{M} = (S, p, \mathsf{O})$ is *convergent*: each infinite run ρ has an infinite observation $\mathsf{O}(\rho) \in \Sigma^\omega$. Two measures can then be defined: using fixed or finite horizon. In the fixed-horizon case, the attacker observes the system for a fixed amount of time and has to make his deduction at the end of this observation. In order to link the amount of time the attacker observes the system and the number of observations they receive, in this case, we only consider non-erasing observation functions O. In the finite-horizon case, the attacker can wait as long as they want, as long as it is a finite amount of time.

Definition 7.4. *Let $\mathcal{M} = (S, p, \mathsf{O})$ be an OMC, μ_0 an initial distribution and* $\mathrm{Sec} \subseteq S$*. A finite observation $w \in \Sigma^*$* discloses the secret *if all runs $\rho \in \mathsf{O}^{-1}(w)$ are secret. It is* min-disclosing *if it discloses the secret and no strict prefix of w does.*

n-disclosure *: When O is non-erasing, we denote by D_n, for $n \in \mathbb{N}$, the set of disclosing observations of length n. The* n-disclosure *(disclosure with fixed horizon n) is* $\mathrm{Disc}_n(\mathcal{M}(\mu_0)) = \sum_{w \in D_n} \mathbb{P}_{\mathcal{M}(\mu_0)}(w)$*;*

Disclosure *: Writing $D_{\min}$ for the set of* min*-disclosing observations, the* disclosure *(w.r.t. finite horizon) is defined by* $\mathrm{Disc}(\mathcal{M}(\mu_0)) = \sum_{w \in D_{\min}} \mathbb{P}_{\mathcal{M}(\mu_0)}(w)$.

Note that if D is the set of disclosing observations, and $\mathcal{V}(\mu_0) = \cup_{w \in D} \cup_{\rho \in \mathsf{O}^{-1}(w)} \mathsf{Cyl}(\rho)$ the set of runs disclosing the secret, then $\mathrm{Disc}(\mathcal{M}(\mu_0))$ equals $\mathbb{P}_{\mathcal{M}(\mu_0)}(\mathcal{V}(\mu_0))$. As waiting longer gives more information, the disclosure with finite horizon is always at least as large as the disclosure with fixed horizon. In fact, $(\mathrm{Disc}_n(\mathcal{M}(\mu_0)))_{n \in \mathbb{N}}$ is a non-decreasing sequence with limit $\mathrm{Disc}(\mathcal{M}(\mu_0))$.

Example 7.2. *Consider the infinitely-branching OMC of Figure 7.2. The single secret run is* $\mathsf{SecRuns} = \{q_0 q_s^\omega\}$ *and its observation is* o_1^ω*. Moreover, the observations of the public (as opposed to secret) infinite runs is* $\mathsf{O}(\overline{\mathsf{SecRuns}}) = o_1^+ o_2^\omega$*. As a consequence, the infinite observed sequence of the secret run discloses the secret,* $\mathrm{Disc}_\omega = \frac{1}{2}$*. However, no finite observation is disclosing:* $\forall n \in \mathbb{N}, \mathrm{Disc}_n = \mathrm{Disc} = 0$*.*

This shows that disclosure and ω*-disclosure differ.*

However, both notions coincide for convergent finitely-branching OMC.

Lemma 7.1. *Let* $\mathcal{M} = (S, p, \mathsf{O})$ *be a Markov chain,* μ_0 *an initial distribution and* $\mathrm{Sec} \subseteq S$*. For* $\mathsf{SecRuns} = \mathsf{Reach}(\mathrm{Sec})$*,* $\mathrm{Disc}(\mathcal{M}(\mu_0)) \leq \mathrm{Disc}_\omega(\mathcal{M}(\mu_0))$ *with equality when* $\mathcal{M}$ *is convergent and finitely branching.*

Proof. We first establish the following claim: if $\mathcal{M}$ is convergent and finitely branching, then the set of runs ρ such that $\mathsf{O}(\rho)$ has length n is finite for any $n > 0$.

We first prove the claim for signalling runs. We proceed by induction on the observable length. There exist finitely many signalling runs of length 0: by convention they are the runs that (1) do not contain any event and (2) start in an unobservable state of $\mathsf{Supp}(\mu_0) \cap \mathsf{O}^{-1}(\varepsilon)$. Let us assume the hypothesis holds for $n \in \mathbb{N}$. For every signalling run ρ_0 with $|\rho_0|_o = n$ we consider the tree formed by the set $O_{n+1} = \{\rho \in \mathsf{SR} \mid \rho_0 \preceq \rho \wedge |\mathsf{O}(\rho)| = n+1\}$ by sharing common prefixes. Internal nodes of this tree correspond to unobservable states while all leaves are observable. Since the OMC is finitely branching, the tree is of bounded degree. By contradiction, assume that the tree is infinite. König's lemma yields an infinite branch containing only unobservable states, which contradicts the convergence hypothesis. Therefore there exist only finitely many signalling runs of observable length $n+1$ extending ρ_0. As there exist finitely many signalling runs of observable length n according to the induction hypothesis, one deduces that there exist finitely many signalling runs of observable length $n+1$[3]. This concludes the induction. The result can then be extended to every runs as, from the convergence hypothesis, for every $n \in \mathbb{N}$ and every run ρ of observable length n, there exists a signalling run $\rho' \in \mathsf{SR}_{n+1}$ such that $\rho \preceq \rho'$. As $|\mathsf{SR}_{n+1}| < \infty$, there are finitely many runs of observable length n.

We now prove that the set of infinite runs $\mathcal{V} = \cup_{w \in D} \cup_{\rho \in \mathsf{O}^{-1}(w)} \mathsf{Cyl}(\rho)$ is contained in $\mathsf{SecRuns} \setminus \mathsf{O}^{-1}(\mathsf{O}(\overline{\mathsf{SecRuns}}))$. Let ρ_1 be an infinite run in $\mathcal{V}$. Then there is a disclosing observation $w_1 \in \Sigma^*$ and a signalling prefix ρ_1' of ρ_1 such that $\mathsf{O}(\rho_1') = w_1$ and ρ_1' is secret. For any infinite run ρ_2 such that $\mathsf{O}(\rho_1) = \mathsf{O}(\rho_2)$, the observation w_1 is also a prefix of $\mathsf{O}(\rho_2)$, hence there is a finite signalling prefix ρ_2' of ρ_2 such that $\mathsf{O}(\rho_2') = w_1$. Since w_1 is disclosing, ρ_2' is also secret, hence ρ_1 belongs to $\mathsf{SecRuns} \setminus \mathsf{O}^{-1}(\mathsf{O}(\overline{\mathsf{SecRuns}}))$ and $\mathrm{Disc}(\mathcal{M}(\mu_0)) \leq \mathrm{Disc}_\omega(\mathcal{M}(\mu_0))$.

For the converse inclusion, let ρ be an infinite run in $\mathsf{SecRuns} \setminus \mathsf{O}^{-1}(\mathsf{O}(\overline{\mathsf{SecRuns}}))$ with observation $\mathsf{O}(\rho) = w = o_1 o_2 \ldots \in \Sigma^\omega$. We prove by contradiction that there is a finite disclosing prefix $\hat{w}$ of w and a signalling prefix $\hat{\rho}$ of ρ such that $\rho \in \mathsf{Cyl}(\hat{\rho})$ and $\mathsf{O}(\hat{\rho}) = \hat{w}$.

[3] For the case $n = 0$ one must also consider the signalling runs obtained from runs starting in states of $\mathsf{Supp}(\mu_0) \setminus \mathsf{O}^{-1}(\varepsilon)$. There are finitely many such runs too.

Otherwise, for any $n \geq 1$, $w_n = o_1 \dots o_n$ is not disclosing and there exists a signalling run ρ_n such that $\mathsf{O}(\rho_n) = w_n$ but ρ_n is not secret. The set $\mathcal{T} = \{\rho' \in \mathsf{SR} \mid \exists n\ \rho' \leq \rho_n\}$ of all signalling prefixes of the ρ_n's form a tree: the root of the tree is ε and the nodes at level k are the prefixes with observation w_k, $\{\rho' \in \mathcal{T} \mid |\mathsf{O}(\rho')| = k\}$. A node ρ'' is a child of ρ' if $|\mathsf{O}(\rho')| = w_k$, $|\mathsf{O}(\rho'')| = w_{k+1}$ for some k and $\rho' \preceq \rho''$. From the claim, we know that $\mathcal{T}$ is of bounded degree. Assuming that it is infinite, König's lemma again yields an infinite branch ρ_∞ such that each prefix of length k is not secret and has observation w_k. Hence ρ_∞ is not secret and has observation $\mathsf{O}(\rho_\infty) = w$, which is a contradiction. □

In the following we only consider finitely-branching convergent systems. As a consequence, we will only focus on disclosure over fixed or finite horizon.

1.2 Opacity for Markov Decision Processes

We now want to add control to the system. The form of control we define here differs from the one used in Chapter 6. Indeed, in the context of diagnosability, the controller observed the system and from this observation, they chose some controllable actions that they blocked until the next observation. Thus the control had to make decisions without an exact knowledge of the state of the system. For opacity we are interested by a control acting with full knowledge. Therefore, the control can be more accurate. We use Markov decision processes where in each state there is a set of possible actions and the controller chooses one of them. This action induces a probability distribution on the next state reached.

Definition 7.5. *An observable Markov Decision Process (OMDP) over alphabet Σ is a tuple $\mathsf{M} = (S, \mathsf{Act}, p, \mathsf{O})$ where S is a finite set of states, $\mathsf{Act} = \cup_{s \in S} A(s)$ where $A(s)$ is a finite non-empty set of actions for each state $s \in S$, $p : S \times \mathsf{Act} \to \mathsf{Dist}(S)$ is a (partial) transition function defined for (s, a) when $a \in A(s)$ and $\mathsf{O} : S \to \Sigma \cup \{\varepsilon\}$ is the observation function.*

The difference with POMDP described in Definition 6.8, page 6.8, beside some syntactic modifications is on how the control is defined. As for OMC, we write $p(s'|s, a)$ instead of $p(s, a)(s')$. We use the same kind of definitions as usual for runs, observed sequences, ... For example, given an initial distribution μ_0, an infinite run of M is a sequence $\rho = s_0 a_0 s_1 a_1 \dots$ where $\mu_0(s_0) > 0$ and $p(s_{i+1}|s_i, a_i) > 0$, for $s_i \in S$, $a_i \in A(s_i)$, for all $i \geq 0$. We denote by $\mathsf{M}(\mu_0)$ the OMDP M with initial distribution μ_0. For decidability and complexity results, we assume that all probabilities occurring in the model (transition probabilities and initial distribution) are rational.

Example 7.3. *Consider the OMDP of Figure 7.5. From the initial state q_0, two actions are possible. If action a is chosen, the system moves to q_1 with probability $1/2$ and to q_2 with probability $1/2$. If b is chosen, every state has a probability $1/3$ to be reached.*

The OMDP model uses both non-deterministic choice (the choice of the action) and probabilistic choice (the induced distribution). The non-determinism is where the

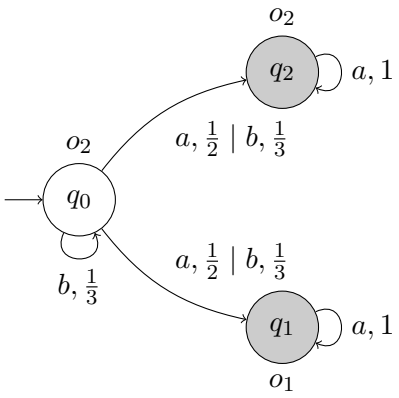


Figure 7.5: An example of OMDP where the initial distribution is a Dirac distribution on q_0. Transitions are labelled by a set of pairs of actions and of the probability to take this transition under that action.

control can be operated. It is resolved by a strategy which associates with every run a distribution on the actions enabled at the last state of the run. Given a finite run ρ with $\mathsf{last}(\rho) = s$, a *decision rule* of an OMDP for ρ is a distribution $\delta \in \mathsf{Dist}(A(s))$ representing the action chosen after ρ. For such a decision rule δ, we write $p(s'|s, \delta) = \sum_{a \in A(s)} \delta(a) p(s'|s, a)$.

Definition 7.6. *A* strategy *for the OMDP* $\mathsf{M} = (S, \mathsf{Act}, p, \mathsf{O})$ *is a mapping* σ *associating to every finite run* ρ *a decision rule* $\sigma(\rho)$.

Given a strategy σ, a run $\rho = s_0 a_0 s_1 a_1 \dots$ of M is σ-compatible if for all i, $a_i \in \mathsf{Supp}(\sigma(s_0 a_0 s_1 a_1 \dots s_i))$.

In order to apply the strategies as defined here one requires to remember the whole run that occurred. Moreover, the strategies are allowed to choose randomly between the different allowed actions. All of this may not always be necessary however. We are thus interested specifically in strategies satisfying specific properties. A strategy σ is *deterministic* if $\sigma(\rho)$ is a Dirac distribution for each finite run ρ. In this case, we denote by $\sigma(\rho)$ the single action $a \in A(\mathsf{last}(\rho))$ such that $\sigma(\rho) = \mathbf{1}_a$. A strategy σ is *observation-based* if for any finite run ρ, $\sigma(\rho)$ only depends on (1) the observed sequence $\mathsf{O}(\rho)$ and (2) the current state $\mathsf{last}(\rho)$, *i.e.* given ρ' such that $\mathsf{O}(\rho) = \mathsf{O}(\rho')$ and $\mathsf{last}(\rho) = \mathsf{last}(\rho')$, we have $\sigma(\rho) = \sigma(\rho')$. We then write $\sigma(\mathsf{O}(\rho), \mathsf{last}(\rho))$ for $\sigma(\rho)$.

Let σ be a strategy and ρ be a σ-compatible run. We define B^σ_ρ the *belief* of ρ w.r.t. σ about states as follows:

$$B^\sigma_\rho = \{s \in S \mid \exists \rho' \ \sigma\text{-compatible}, \ \mathsf{O}(\rho') = \mathsf{O}(\rho) \wedge s = \mathsf{last}(\rho') \wedge \mathsf{O}(s) \neq \varepsilon\}.$$

The belief B^σ_ρ contains the set of states that can be reached under the strategy σ and with observation $\mathsf{O}(\rho)$. A strategy σ is *belief-based* if for all finite run ρ, $\sigma(\rho)$ only depends on the belief B^σ_ρ and the current state $\mathsf{last}(\rho)$, *i.e.* given ρ' such that $B^\sigma_\rho = B^\sigma_{\rho'}$

and $\mathsf{last}(\rho) = \mathsf{last}(\rho')$, we have $\sigma(\rho) = \sigma(\rho')$. Observe that a belief-based strategy is observation-based since B^σ_ρ only depends on $w = \mathsf{O}(\rho)$. So we also write B^σ_w for B^σ_ρ. A strategy σ is *memoryless* if $\sigma(\rho)$ only depends on $\mathsf{last}(\rho)$ for all ρ.

The semantics of a OMDP M with initial distribution μ_0 under the strategy σ is a (possibly infinite)observable Markov chain $\mathsf{M}_\sigma(\mu_0)$ where each state is associated with a finite σ-compatible run of $\mathsf{M}(\mu_0)$, that can be equipped with the observation function mapping $\mathsf{O}(\mathsf{last}(\rho))$ to the state associated with the finite run ρ. The transition function p_σ is defined for ρ a finite run and $\rho' = \rho a s'$ by $p_\sigma(\rho'|\rho) = \sigma(\rho)(a)p(s'|s,a)$ and we denote by $\mathbb{P}_{\mathsf{M}_\sigma(\mu_0)}$ (or $\mathbb{P}_\sigma$ for short when there is no ambiguity) the associated probability measure. Writing $\mathcal{V}_\sigma(\mu_0)$ for the set of runs disclosing the secret in $\mathsf{M}_\sigma(\mu_0)$, we have $\mathrm{Disc}(\mathsf{M}_\sigma(\mu_0)) = \mathbb{P}_{\mathsf{M}_\sigma(\mu_0)}(\mathcal{V}_\sigma(\mu_0))$. We assume all OMDP considered are convergent (there is no cycle of unobservable states), which implies the convergence of all OMC induced by strategies.

Example 7.4. *Consider the OMC of Figure 7.6. It represents the semantics of the OMDP of Figure 7.5 with the strategy σ choosing the action b initially, then always choosing action a. σ is observation-based as the only run for which it does not select a is the empty run, which is the only run with observed sequence o_2. It is also belief-based as the empty run is the only run with belief $\{q_0\}$. Indeed, after some observations, the current belief is either $\{q_0, q_2\}$ or $\{q_1\}$. It is however not memoryless as the empty run and q_0bq_0 both ends in q_0 but the same action is not chosen in both cases. After three observations, under σ, the system cannot be in q_0 any more, it is thus necessarily in a secret state. Therefore,* $\mathrm{Disc}(\mathsf{M}_\sigma(\mu_0)) = 1$.

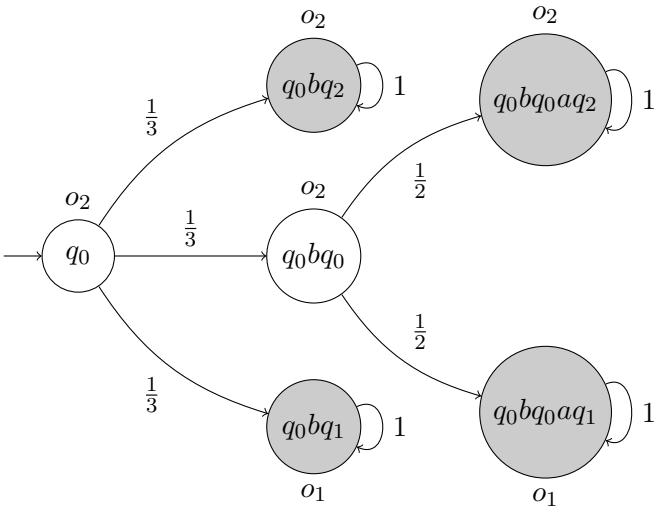


Figure 7.6: The OMC induced by the strategy choosing the action b initially, then always choosing action a for the OMDP of Figure 7.5.

The control can be either adversarial or cooperative with respect to the system: it

can try to either maximise or minimise the opacity. We therefore define the disclosure value of an OMDP according to the type of the strategies. We only consider ε-disclosure for fixed horizon in light of the undecidability result of Theorem 7.1.

Definition 7.7. *Given an OMDP* $\mathsf{M} = (S, \mathsf{Act}, p, \mathsf{O})$*, an initial distribution* μ_0 *and a secret* $\mathrm{Sec} \subseteq S$*, for* $\mathrm{disc} \in \{\mathrm{Disc}, \mathrm{Disc}_n, \mathrm{Disc}_n^\varepsilon\}$*,* $n \in \mathbb{N}$ *and* $0 < \varepsilon < 1$*, the maximal disclosure of* Sec *in* M *is* $\mathrm{disc}_{\max}(\mathsf{M}(\mu_0)) = \sup_\sigma \mathrm{disc}(\mathsf{M}_\sigma(\mu_0))$ *and the minimal disclosure of* Sec *is* $\mathrm{disc}_{\min}(\mathsf{M}(\mu_0)) = \inf_\sigma \mathrm{disc}(\mathsf{M}_\sigma(\mu_0))$*.*

Note that the construction ensuring that once a secret state is visited, the run remains in a secret state forever, extends naturally from OMC to OMDP. We only consider OMDP of this form in the rest of this chapter.

Example 7.5. *Consider the OMDP of Figure 7.5. As soon as the strategy selects action a, the system enters a secret state and discloses the secret with probability* 1*. Therefore* $\mathrm{Disc}_{\max}(\mathsf{M}(\mu_0)) = 1$*. If the strategy only selects action b however, observing a '*o_1*' clearly shows the system is in* q_1*, thus disclosing the secret, while after observing at least two '*o_2*', the belief is* $\{q_0, q_2\}$ *which does not disclose the secret. The probability to observe at some point '*o_1*' being equal to* $1/2$*,* $\mathrm{Disc}_{\min}(\mathsf{M}(\mu_0)) = 1$*.*

We study the following problems for OMDP over finite or fixed horizon:

- **Computation problems.**
  - The *value problem*: compute the disclosure;
  - The *strategy problem*: compute an optimal strategy whenever it exists.
- **Quantitative decision problems.** Let $\bowtie = \geq$ for maximisation and $\bowtie = \leq$ for minimisation.
  - The *disclosure problem*: Given M and a threshold $\theta \in [0, 1]$, decide if $\mathrm{disc}(\mathsf{M}) \bowtie \theta$;
  - The *strategy decision problem*: decide if there exist a strategy σ such that $\mathrm{disc}(\mathsf{M}_\sigma) \bowtie \theta$.
- **Qualitative decision problems.**
  - The *limit-sure disclosure problem*: the disclosure problem when $\theta = 1$ for maximisation and $\theta = 0$ for minimisation;
  - The *almost-sure disclosure problem*: the strategy decision problem when $\theta = 1$ for maximisation and $\theta = 0$ for minimisation.

For the complexity results regarding a fixed horizon n, we assume that n is written in unary representation or bounded by a polynomial in the size of the model where the polynomial is independent of the model as done in classical studies (see for instance [PT87]).

As said earlier, the whole power of the strategies we defined may not be necessary to answer the above problems. Restricting ourself to a subset of strategies that gives the same disclosure values can help simplify the proofs and the representation in practice of these strategies. Moreover, it helps understanding what is important in the control of an OMDP to optimise the disclosure. We thus show that for disclosure problems we can restrict strategies to observation-based ones.

Proposition 7.1. *Given an OMDP, a secret and a strategy σ, there exists an observation-based strategy σ' such that for* $\mathrm{disc} \in \{\mathrm{Disc}, \mathrm{Disc}_n, \mathrm{Disc}_n^{\varepsilon}\}$, $\mathrm{disc}(\mathsf{M}_\sigma(\mu_0)) = \mathrm{disc}(\mathsf{M}_{\sigma'}(\mu_0))$.

For this proof, from an arbitrary strategy σ, we build an observation-based strategy σ' with the same disclosure value. The strategy σ' is randomised and is obtained by choosing, after an observed sequence w, a distribution on the different choices made by σ on runs with observed sequence w. This is done so that the probability of choosing an action after observing w is the same for both strategies.

We then prove that σ' meets the same disclosure value as σ. More precisely, we establish that the probability to reach a state with a given observation is the same for both strategies. This is done by induction on the length of the observed sequence and on an ordering of the unobservable states. Two cases have to be considered, depending on if the last state is observable or not. However, each case is dealt with in the same way (both for the initialisation and for the induction step). Thus we only detail the first one.

Proof. Let $\mathsf{M} = (S, \mathsf{Act}, p, \mathsf{O})$ be an OMDP with initial distribution μ_0, and let σ be a strategy. For an observation $w \in \Sigma^*$ and a state $s \in S$, we define the sets (note that these are finite sets given the claim in Lemma 7.1) $R(w,s) = \{\rho \text{ finite run of } \mathsf{M}_\sigma(\mu_0) \mid \mathsf{O}(\rho) = w \wedge \mathsf{last}(\rho) = s\}$.

We now define a mapping $\hat{\sigma}$ from $\Sigma^* \times S$ to $\mathsf{Dist}(\mathsf{Act})$ by

$$\hat{\sigma}(w,s) = \frac{1}{\sum_{\rho \in R(w,s)} \mathbb{P}_\sigma(\rho)} \sum_{\rho \in R(w,s)} \mathbb{P}_\sigma(\rho)\sigma(\rho).$$

$\hat{\sigma}(w,s)$ corresponds to the average choice made by σ after a run with observed sequence w and ending in s. Using $\hat{\sigma}$, we define the new strategy σ' for a finite run ρ by $\sigma'(\rho) = \hat{\sigma}(\mathsf{O}(\rho), \mathsf{last}(\rho))$. We claim that $\mathbb{P}_{\sigma'}(R(w,s)) = \mathbb{P}_\sigma(R(w,s))$ for any observation w and any state s, which entails equality of disclosure.

Partitioning the set of states into $S = S_o \uplus S_u$ where $S_u = \mathsf{O}^{-1}(\varepsilon)$, we can assume a topological sort on the subgraph obtained by removing all edges in $S \times S_o$ (this subgraph is acyclic due to the hypothesis of convergence). This means that there exists a numbering η of the states so that if $\eta(s') > \eta(s)$, there is no transition from s to s'. We proceed to prove the above claim by a joint induction on the pairs (w,s) using $|w|$ and $\eta(s)$.

For the base cases, we need to establish the property for $w = \varepsilon$ with $s \in S_u$, and for $w \in \Sigma$ with $s \in S_o$, where $\mu_0(s) > 0$ in both cases.

Case 1. By induction on $\eta(s)$, we consider a state $s \in S_u$ such that $\eta(s) = \min_{s' \in S_u}(\eta(s'))$. Then $\mathbb{P}_{\sigma'}(R(\varepsilon, s)) = \mu_0(s) = \mathbb{P}_{\sigma}(R(\varepsilon, s))$. Assuming the property holds for (ε, s) with $\eta(s) \leq n$, we prove it for s' with $\eta(s') = n + 1$. We have:

$$\mathbb{P}_{\sigma'}(R(\varepsilon, s')) = \mu_0(s') + \sum_{s \in S_u, \eta(s) < \eta(s')} \sum_{a \in A(s)} p(s'|s, a) \sum_{\rho \in R(\varepsilon, s)} \mathbb{P}_{\sigma'}(\rho)\sigma'(\rho)(a)$$

and using the definition of σ' yields:

$$\mathbb{P}_{\sigma'}(R(\varepsilon, s')) = \mu_0(s') + \sum_{s \in S_u, \eta(s) < \eta(s')} \sum_{a \in A(s)} p(s'|s, a)\hat{\sigma}(\varepsilon, s)(a) \sum_{\rho \in R(\varepsilon, s)} \mathbb{P}_{\sigma'}(\rho)$$

$$= \mu_0(s') + \sum_{s \in S_u, \eta(s) < \eta(s')} \sum_{a \in A(s)} p(s'|s, a) \frac{\sum_{\rho \in R(\varepsilon, s)} \mathbb{P}_{\sigma}(\rho)\sigma(\rho)(a)}{\sum_{\rho \in R(\varepsilon, s)} \mathbb{P}_{\sigma}(\rho)} \sum_{\rho \in R(\varepsilon, s)} \mathbb{P}_{\sigma'}(\rho).$$

Applying the induction hypothesis on (ε, s) yields $\sum_{\rho \in R(\varepsilon, s)} \mathbb{P}_{\sigma'}(\rho) = \mathbb{P}_{\sigma'}(R(\varepsilon, s)) = \mathbb{P}_{\sigma}(R(\varepsilon, s)) = \sum_{\rho \in R(\varepsilon, s)} \mathbb{P}_{\sigma}(\rho)$ thus:

$$\mathbb{P}_{\sigma'}(R(\varepsilon, s')) = \mu_0(s') + \sum_{s \in S_u, \eta(s) < \eta(s')} \sum_{a \in A(s)} p(s'|s, a) \sum_{\rho \in R(\varepsilon, s)} \mathbb{P}_{\sigma}(\rho)\sigma(\rho)(a) = \mathbb{P}_{\sigma}(R(\varepsilon, s')).$$

Case 2. We now consider $w = o \in \Sigma$ and $s' \in S_o$, hence $\mathsf{O}(s') = o$. Then:

$$\mathbb{P}_{\sigma'}(R(o, s')) = \mu_0(s') + \sum_{s \in S_u} \sum_{a \in A(s)} p(s'|s, a) \sum_{\rho \in R(\varepsilon, s)} \mathbb{P}_{\sigma'}(\rho)\sigma'(\rho)(a)$$

and a reasoning similar as above yields the result.

For the induction step, we first need to prove the property for (w, s') with $s' \in S_u$, assuming it holds for all (w, s) with $s \in S_o$ and for all (w, s) with $s \in S_u$ and $\eta(s) < \eta(s')$. Then we have:

$$\mathbb{P}_{\sigma'}(R(w, s')) = \sum_{\substack{s \in S_o \\ s \in S_u, \eta(s) < \eta(s')}} \sum_{a \in A(s)} p(s'|s, a) \sum_{\rho \in R(w, s)} \mathbb{P}_{\sigma'}(\rho)\sigma'(\rho)(a)$$

and we can conclude along the same lines as above.

Finally, we consider (w', s') with $w' = wo \in \Sigma^*\Sigma_o$ and $s \in S_o$, with:

$$\mathbb{P}_{\sigma'}(R(w', s')) = \sum_{s \in S} \sum_{a \in A(s)} p(s'|s, a) \sum_{\rho \in R(w, s)} \mathbb{P}_{\sigma'}(\rho)\sigma'(\rho)(a)$$

which again implies the desired result. □

As seen in the previous proof, erasing observations leads to technical and cumbersome developments. In order to avoid them in the design of procedures for the finite-horizon case, we apply the preliminary transformation that ensures the observation is non-erasing described in the next proposition. We precisely state the size of the obtained OMDP in view of complexity results.

Proposition 7.2. *Given an OMDP* $\mathsf{M} = (S, \mathsf{Act}, p, \mathsf{O})$*, an initial distribution* μ_0 *and a secret* Sec*, one can build in exponential time an OMDP* $\mathsf{M}' = (S', \mathsf{Act}', p', \mathsf{O}')$*, an initial distribution* μ_0' *and a secret* Sec' *where* O' *is non-erasing and for* $\mathrm{disc} \in \{\mathrm{Disc}_{min}, \mathrm{Disc}_{max}\}$ $\mathrm{disc}(\mathsf{M}(\mu_0)) = \mathrm{disc}(\mathsf{M}'(\mu_0'))$*. In addition, the size of* S'*,* p' *and* μ_0' *is polynomial w.r.t. the ones of* S*,* p *and* μ_0*. The size of* Act' *is polynomial w.r.t. the size of* Act *and exponential w.r.t. the size of* S*.*

The main idea of the construction is that every time a run visits an observable state, an observation-based strategy can fix a set of action for the current state and for every unobservable state. It will then keep this choice until the run visits a new observable state. Once such a set of actions is fixed, one can easily compute the probability distribution to reach the next observable state. Unobservable states can thus be removed from the system. We also add a new state to deal with the possibility that an unobservable state had a positive probability in the initial distribution.

Proof. We first build the new OMDP and then explain the correspondence between strategies in both models, which induces the relationship between disclosures.
Construction of the OMDP. We start from OMDP $\mathsf{M} = (S, \mathsf{Act}, p, \mathsf{O})$ with $\mathsf{Act} = \cup_{s \in S} A(s)$, observation alphabet Σ, and a set of secret states $\mathrm{Sec} \subseteq S$. Choosing a fresh observation symbol $\sharp$ and a fresh state $s_\sharp$, we build an OMDP $\mathsf{M}' = (S', \mathsf{Act}', p', \mathsf{O}')$ with set of states $S' = \{s_\sharp\} \cup (S \setminus \mathsf{O}^{-1}(\varepsilon))$, and observation alphabet $\Sigma \cup \{\sharp\}$, where the initial distribution is $\mathbf{1}_{s_\sharp}$. The observation function O' is defined by $\mathsf{O}'(s_\sharp) = \sharp$ and $\mathsf{O}'(s) = \mathsf{O}(s)$ otherwise. Note that all states have non-trivial observation. The set of actions of M' is $\mathsf{Act}' = \mathsf{DR}$ where DR is the set of vectors of deterministic decision rules $\vec{\delta}$ over S, *i.e.* such that $\vec{\delta}(s) \in A(s)$. The intuition of DR is that the actions associated with the current state and any unobservable state by the strategy after an observation is fixed until the next observable state, so we can gather this set of action into a single action.

We now define the transition probabilities, starting by the transitions exiting $s_\sharp$. For a run $\rho = s_0 a_1 \dots a_n s_n$, we write $\pi(\rho) = \prod_{i=1}^{n} p(s_i | s_{i-1}, a_i)$ and $\mathsf{first}(\rho) = s_0$. Given an observable state $s \in S$ and $\vec{\delta} \in \mathsf{DR}$, the set $\hat{E}(s_\sharp, \vec{\delta}, s)$ contains the finite runs $\rho = s_0 a_1 s_1 \dots a_n s_n$ starting from some $s_0 \in \mathsf{Supp}(\mu_0)$ and ending in $s_n = s$ such that $a_i = \vec{\delta}(s_{i-1})$ for all i, $1 \leq i \leq n$ and all states $s_0, \dots, s_{n-1}$ are unobservable. Observe that the intermediary states are all distinct due to the convergence of the OMDP. We set $p'(s \mid s_\sharp, \vec{\delta}) = \sum_{\rho \in \hat{E}(s_\sharp, \vec{\delta}, s)} \mu_0(\mathsf{first}(\rho)) \pi(\rho)$. If $s \in \mathsf{Supp}(\mu_0)$, the set $\hat{E}(s_\sharp, \vec{\delta}, s)$ contains the run reduced to $\rho = s$.

We turn to the transitions exiting the other states. It is easier as we do not need to take the initial distribution into account. Given a state $s \neq s_\sharp$, an action $\vec{\delta} \in \mathsf{DR}$ and an observable state s', we consider the finite set $\hat{E}(s, \vec{\delta}, s')$ of signalling runs of

M $\rho = s_0 a_1 \dots a_n s_n$ starting in $s_0 = s$ and ending in $s_n = s'$ such that for each i, $1 < i \leq n$, $a_i = \vec{\delta}(s_{i-1})$, and all intermediate states are unobservable. We set $p'(s'|s, \vec{\delta}) = \sum_{\rho \in \hat{E}(s,\vec{\delta},s')} \pi(\rho)$. Note that $\hat{E}(s, \vec{\delta}, s')$ may include runs like $\rho = s\ \vec{\delta}(s)s'$.

In order to efficiently compute the transition function of some $\vec{\delta}$, one uses a topological sort of the unobservable states thanks to the convergence hypothesis, and then compute the probability from observable states to reach first the unobservable states topologically sorted and then the observable states. This gives a polynomial time computation of the transition function of $\vec{\delta}$. Thus, the size of S', p' and μ'_0 is polynomial w.r.t. the ones of S, p and μ_0. Moreover, the size of Act' is polynomial w.r.t. the size of Act and exponential w.r.t. the size of S.

Correspondence between strategies. The above construction ensures that any run $\rho' = s_\sharp \vec{\delta}_1 s_1 \dots \vec{\delta}_k s_k$ of M' corresponds to the set of runs $\rho = \rho_1 s_1, \dots \rho_k s_k$ of M containing the sequence $s_1 \dots s_k$ of observable states, with $\rho_1 \in \hat{E}(s_\sharp, \vec{\delta}_1, s_1)$ and $\rho_i \in \hat{E}(s_{i-1}, \vec{\delta}_i, s_i)$ for $1 < i \leq k$. All runs in the set have the same observation $w = \mathsf{O}(s_1) \dots \mathsf{O}(s_k)$ with $\mathsf{O}'(\rho') = \sharp w$.

To show that disclosure over finite horizon is the same in both OMDP, we establish correspondences between the strategies of M and M' and the associated disclosure value. From Proposition 7.1, we can restrict to observation-based strategies.

• Let σ' be an observation-based strategy of M', defined on $\sharp\Sigma^* \times S'$. Given an observation $w \in \Sigma^*$ there exists $\vec{\delta}$ such that for every state $s \in S'$, we have $\sigma'(\sharp w, s) = \vec{\delta}$. We define $\sigma(w, s) = \vec{\delta}(s)$. Then, writing $\mathbb{P}_\sigma$ (resp. $\mathbb{P}_{\sigma'}$) instead of $\mathbb{P}_{\mathsf{M}_\sigma(\mu_0)}$ (resp. $\mathbb{P}_{\mathsf{M}'_{\sigma'}(\mu'_0)}$), and defining for $w \in \Sigma^*$ and $s \in S \setminus \mathsf{O}^{-1}(\varepsilon)$, $R(w, s) = \{\rho \in \mathsf{SR}^{\mathsf{M}_\sigma(\mu_0)} \mid \mathsf{O}(\rho) = w \wedge \mathsf{last}(\rho) = s\}$ and $R'(w, s) = \{\rho' \in \mathsf{SR}^{\mathsf{M}'_{\sigma'}(\mu'_0)} \mid \mathsf{O}(\rho') = \sharp w \wedge \mathsf{last}(\rho') = s\}$, we have $\mathbb{P}_\sigma(R(w, s)) = \mathbb{P}_{\sigma'}(R'(w, s))$.

• Conversely, given an observation-based strategy σ of M, we build an observation-based strategy σ' of M' as follows: Given $w \in \Sigma^*$, we define the mapping $\sigma'(\sharp w) : S \to \mathsf{Dist}(\mathsf{DR})$ by $\sigma'(\sharp w)(s) = \sigma(w, s)$ for any $s \in S$. Then, using the same notations as above, we have $\mathbb{P}_\sigma(R(w, s)) = \mathbb{P}_{\sigma'}(R'(w, s))$.

Therefore, defining the set of secret states of S' by $\mathrm{Sec}' = \mathrm{Sec} \cap S'$, as the set of secret states is absorbing, the disclosures over finite horizon are equal for σ and σ'. □

Thus for finite horizon, one can restrict oneself to non-erasing observation functions with some care on the complexity of the actions. Also, on fixed horizon, we assume the observation function is non-erasing. Thus in both cases we are able to use this assumption.

2 Maximisation with finite horizon

We start the study of the disclosure problem with the maximisation objective over finite horizon. In other words, here the strategy tries to maximise the disclosure of the secret after an arbitrarily long, yet finite, amount of time.

In Subsection 2.1, we show how to restrict the study to deterministic strategies without loss of generalities. We show that most of the notions are unfortunately un-

decidable in Subsection 2.2, and prove the decidability of the almost-sure disclosure in Subsection 2.3.

2.1 Deterministic strategies are sufficient

We showed in Proposition 7.1 that one can limit oneself to observation-based strategies. In fact, for maximisation problems, one can do even better. Indeed, the additional power given by randomisation is not useful and thus observation-based deterministic strategies are sufficient.

Proposition 7.3. *Given an OMDP* M*, a secret* Sec *and a disclosure notion* $\mathrm{disc} \in \{\mathrm{Disc}, \mathrm{Disc}_n, \mathrm{Disc}_n^\varepsilon\}$*, for any observation-based strategy* σ *there exists a deterministic observation-based strategy* σ' *such that* $\mathrm{disc}(\mathsf{M}_\sigma(\mu_0)) \leq \mathrm{disc}(\mathsf{M}_{\sigma'}(\mu_0))$.

This proof strongly uses Lemma 1 of [CDGH10] (or alternatively [GS14]) which establishes, in an active stochastic setting, that deterministic strategies are sufficient to optimise an objective defined by a set of infinite runs. This Lemma does not directly give the result we want as, contrary to the objectives used in their paper, the choice of the strategy modifies which runs are disclosing. However, as a disclosing run for a randomised strategy is also a disclosing run for a deterministic strategy that does not introduce new runs, we can use parts of their proof to show our result.

Proof. In the proof of Lemma 1 of [CDGH10], the authors show that a randomised observation-based strategy can be seen as a convex combination of a family of deterministic observation-based strategy. As a consequence, in our framework, given an observation-based strategy σ and a disclosure notion disc, there exists an observation-based deterministic strategy σ_{det} such that for every finite run ρ, $\mathsf{Supp}(\sigma_{det}(\rho)) \subseteq \mathsf{Supp}(\sigma(\rho))$ and $\mathbb{P}_{\mathsf{M}_{\sigma_{det}}(\mu_0)}(\mathcal{V}_\sigma(\mu_0)) \geq \mathbb{P}_{\mathsf{M}_\sigma(\mu_0)}(\mathcal{V}_\sigma(\mu_0))$.

The second property is not enough to conclude, as a disclosing run under σ is not necessarily a disclosing run under σ'. However, thanks to the first property we can obtain that $\mathcal{V}_\sigma(\mu_0) \cap \Omega^{\mathsf{M}_{\sigma_{det}}(\mu_0)} \subseteq \mathcal{V}_{\sigma_{det}}(\mu_0)$. Indeed, as σ is more permissive than σ_{det}, $\Omega\mathsf{M}_{\sigma_{det}}(\mu_0) \subseteq \Omega\mathsf{M}_\sigma(\mu_0)$. This implies that, given a run ρ, if $\mathsf{O}(\rho)$ discloses the secret with the strategy σ then either $\mathsf{O}(\rho)$ discloses the secret with the strategy σ_{det} or $\mathsf{O}(\rho)$ cannot be observed with σ_{det}.

This implies: $\mathbb{P}_{\sigma_{det}}(\mathcal{V}_\sigma(\mu_0)) = \mathbb{P}_{\sigma_{det}}(\mathcal{V}_\sigma(\mu_0) \cap \Omega^{\mathsf{M}_{\sigma_{det}}(\mu_0)}) \leq \mathbb{P}_{\sigma_{det}(\mu_0)}(\mathcal{V}_{\sigma_{det}}(\mu_0))$.

Therefore, $\mathrm{disc}(\mathsf{M}_{\sigma_{det}}(\mu_0)) = \mathbb{P}_{\sigma_{det}}(\mathcal{V}_{\sigma_{det}}(\mu_0)) \geq \mathbb{P}_{\sigma_{det}}(\mathcal{V}_\sigma(\mu_0)) \geq \mathbb{P}_\sigma(\mathcal{V}_\sigma(\mu_0))$ and the result holds since $\mathbb{P}_\sigma(\mathcal{V}_\sigma(\mu_0)) = \mathrm{disc}(\mathsf{M}_\sigma(\mu_0))$. □

Observe that this proof shows that the restriction to deterministic strategy does not decrease the disclosure. However, it does not necessarily keep the same disclosure as before contrary to the proof used to restrict to observation-based strategies. Therefore it cannot be applied for minimisation problems.

2.2 Undecidability of the disclosure and limit-sure disclosure problems

As mentioned in the previous proof, one of the difficulties of opacity is that the set of disclosing runs depends on the strategy: a transition can be completely blocked by some strategy, modifying the set of disclosing observations. This was illustrated in Figure 7.5, where choosing action a in state q_0 removes the edge to q_0. This situation was excluded in the computation of the disclosure presented in [BKMS16, BKMS18] where the authors study a restricted form of Interval Markov Chains [JL91]. The disclosure problem for the general class of OMDP was left open. We answer negatively to the general problem by proving undecidability of the disclosure problem, hence the disclosure cannot be computed in general. Undecidability also holds for limit-sure disclosure.

Writing $\mathbb{I}$ for the set of intervals in $[0,1]$, an interval Markov chain (IMC) over an alphabet Σ is a tuple $\mathsf{M} = (S, s_{init}, I, \mathsf{O})$ where S is the set of states, s_{init} is the initial state, $I : S \to \mathbb{I}^S$ associates with every state $s \in S$ a mapping from S to $\mathbb{I}$, and $\mathsf{O} : S \to \Sigma \cup \{\varepsilon\}$ is the observation function. We abuse notations by writing $\mu \in I(s)$ to denote any distribution $\mu : S \to [0,1]$ such that for all $s' \in S$, $\mu(s') \in I(s' \mid s)$. The notion of run ρ is the same as for an OMC but a transition from $s = \mathsf{last}(\rho)$ to some successor requires the choice of a distribution $\mu \in I(s)$. A strategy of IMC M is thus a mapping σ associating with each finite run ρ with $s = \mathsf{last}(\rho)$ a distribution $\sigma(\rho) \in I(s)$. In other words, an IMC is a OMDP where the chosen action represents a set of probabilities satisfying the interval conditions set by I and summing to 1. In fact, an IMC can be transformed into an (exponentially larger) OMDP where actions are the basic feasible solutions of the linear program specified by the constraints associated with intervals [SVA06, CSH08]. Thus undecidability results for IMC also hold for OMDP.

Example 7.6. *Consider the IMC of Figure 7.7. From the initial state, the strategy must attach a probability $p_1 \in [0, \frac{1}{2}]$ to the transition to s_1 and a probability $p_2 \in [\frac{1}{4}, 1]$ to the transition to s_2. As, in order to obtain a distribution, we require that $p_1 + p_2 = 1$, p_2 is* de facto *restricted to the interval $[\frac{1}{2}, 1]$. If $p_1 = \frac{1}{4}$ is selected, then $p_2 = \frac{3}{4}$ and the run has a probability p_1 to move to s_1 and p_2 to move to s_2. In these two states, the only exiting transition is labelled by the interval $\{1\}$ which we simplified by removing the braces in the figure. Thus, the run then loops indefinitely on the state.*

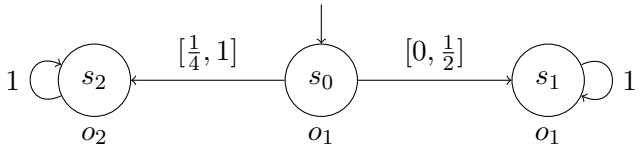


Figure 7.7: Example of IMC.

Theorem 7.2. *The maximal finite-horizon disclosure problem is undecidable for OMDP, even when the secret is reached with probability* 1 *and for a non-erasing observation function.*

The maximal finite-horizon disclosure problem when restricted to finite-memory strategies is undecidable (even with the same additional assumptions).

Starting from a PA A, we build an IMC $\mathsf{M}_A = (S, s_0, I, \mathsf{O})$ such that there exists a word $w \in \{a,b\}^*$ with $\mathbb{P}^F_A(w) > \frac{1}{2}$ if and only if $\mathrm{Disc}_{\max}(\mathsf{M}_A) > \frac{1}{4}$. The proof of Theorem 7.2 is more involved than the proof of Theorem 7.1 because the strategies must be taken into account. The goal is to have the strategy choose a single word then "plays" it in the IMC and the probability to disclose the secret is half of the probability of the selected word. However, just with this, nothing would prevent the strategy to switch words during the run if it realises that the current run will not disclose the secret. We add a second component that ensures that if the strategy deviated from the single selected word the current run will not disclose the secret. Doing so, the strategy loses the advantage of knowing the current state of the run.

Proof. We first give the construction in two steps. The first step is very similar to what was done in the proof of Theorem 7.1. Starting from a PA $A = (Q, q_0, \{a,b\}, T, F)$ that is supposed complete, we build an IMC $\hat{A} = (\hat{Q}, q_0, \hat{I}, \hat{\mathsf{O}})$ where $\Sigma = \{a,b\}$ is the observation alphabet. The set of states is $\hat{Q} = Q \cup \{q_c \mid q \in Q \wedge c \in \{a,b\}\}$, with initial state q_0. The observation function $\hat{\mathsf{O}}$ is defined by $\hat{\mathsf{O}}(q) = \varepsilon$ and $\hat{\mathsf{O}}(q_c) = c$ for $q \in Q$ and $c \in \{a,b\}$. The interval mapping $\hat{I} : \hat{Q} \to \mathbb{I}^{\hat{Q}}$ is defined for $q, q' \in Q$ and $c \in \{a,b\}$ by:

- $\hat{I}(q' \mid q_c) = T(q' \mid q, c)$ is a point interval;
- $\hat{I}(q_c \mid q) = [0,1]$.

Compared to the illustration given in Figure 7.3, this construction amounts to replacing all $\flat$ by ε (making the states non observable) and the probabilities $\frac{1}{4}$ from original states to new ones by the interval $[0,1]$.

However, the construction of the complete IMC $\mathsf{M}_A = (S, s_0, I, \mathsf{O})$ from A is more involved and requires to add a supplementary gadget limiting the power of the strategy. This is why we first use an observation function which can erase states and explain at the end how to relax this hypothesis. The construction is illustrated in Figure 7.8 with some conventions to avoid too many edges, a final state from A (*e.g.* like q_f) is doubly circled.

- $S = \{s_0, s_1, q^1_\sharp, q^2_\sharp, q_\flat, q_s\} \cup \hat{Q} \cup \{s_c \mid c \in \{a,b,\sharp\}\} \cup \{r_c \mid c \in \{a,b,\sharp,\flat\}\}$;
- $I(s_1 \mid s_0) = I(q_0 \mid s_0) = \frac{1}{2}$ and the restriction of I to $\hat{Q}$ is $\hat{I}$. For all $c \in \{1,a,b,\sharp\}$, $c' \in \{a,b,\sharp\}$, $I(s_{c'} \mid s_c) = \frac{1}{6}$ and $I(r_{c'} \mid s_c) = [0, \frac{1}{4}]$, for all $c, c' \in \{a,b,\sharp,\flat\}$, $I(r_{c'} \mid r_c) = \frac{1}{5}$ and $I(q_s \mid r_c) = \frac{1}{5}$. For all $q \in Q \setminus F$, $I(q^1_\sharp \mid q) = [0,1]$, for all $q \in F$, $I(q^2_\sharp \mid q) = [0,1]$, and $I(q_s \mid q^1_\sharp) = I(q_\flat \mid q^2_\sharp) = I(q_s \mid q_\flat) = I(q_s \mid q_s) = 1$.
- O extends $\hat{\mathsf{O}}$ by: $\mathsf{O}(s_0) = \mathsf{O}(s_1) = \varepsilon$, $\mathsf{O}(q^1_\sharp) = \mathsf{O}(q^2_\sharp) = \mathsf{O}(q_s) = \sharp$, $\mathsf{O}(q_\flat) = \flat$, for all $c \in \{a,b,\sharp,\flat\}, \mathsf{O}(r_c) = c$, and for all $c \in \{a,b,\sharp\}, \mathsf{O}(s_c) = c$.

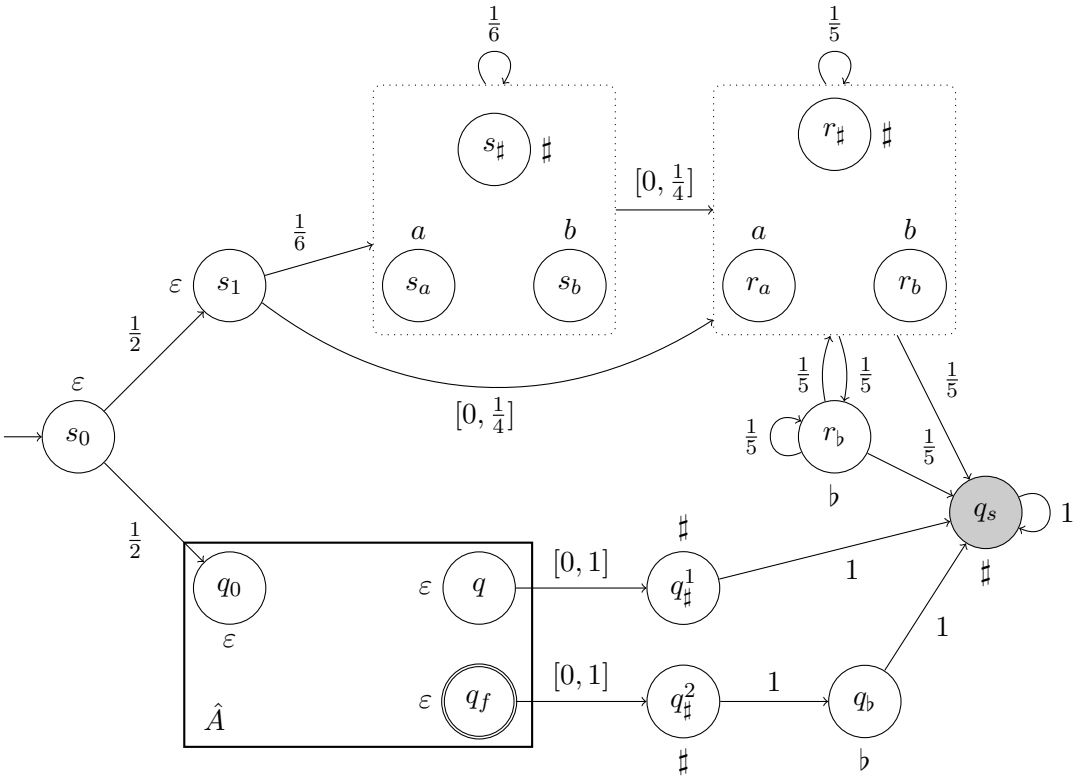


Figure 7.8: Reduction from the strict emptiness problem of PA to the maximal finite-horizon disclosure problem. An edge outgoing from a dotted box should be duplicated to originate from all states in the box and an edge entering a dotted box from a state s should be duplicated from s to any state in the box. Hence a loop on a dotted box represents a complete graph inside the box (including self-loops).

Informally, for $\mathrm{Sec} = \{q_s\}$, the upper gadget ensures that for any strategy σ there is at most one word $w \in \{a, b\}^*$ such that the observation $w\sharp\flat\sharp$ discloses the secret. The lower gadget allows one to generate secret runs of observation $w\sharp\flat\sharp$ with half the probability as the one assigned by the PA to w.

We now formally prove that there exists a word $w \in \{a, b\}^*$ with $\mathbb{P}_A^F(w) > \frac{1}{2}$ if and only if $\mathrm{Disc}_{\max}(\mathsf{M}_A) > \frac{1}{4}$.
First suppose there exists a word $w = a_1 \dots a_n \in \{a, b\}^*$ accepted with probability greater than $\frac{1}{2}$ in A. We define the strategy σ for a finite run ρ in both parts of M_A (when relevant) as follows:

- In the upper part, assume that ρ ends in a state s_c with $c \in \{1, a, b, \sharp\}$. If there exists $i < n$ such that $\mathsf{O}(\rho) = a_1 \dots a_i$ then $\sigma(\rho)(r_{a_{i+1}}) = 0$, leaving no choice for the rest of the distribution: In order for the sum of probabilities to be equal to 1 we have for $b \neq a_{i+1}$, $\sigma(\rho)(r_b) = \frac{1}{4}$. If $\mathsf{O}(\rho) = w$, then $\sigma(\rho)(r_\sharp) = 0$, which also

leaves no choice for the rest of the distribution.

- In the bottom part, we can assume that ρ ends in a state $q \in Q$. If there exists $i < n$ such that $\mathsf{O}(\rho) = a_1 \dots a_i$ then $\sigma(\rho)(q_{a_{i+1}}) = 1$. Finally, if $\mathsf{O}(\rho) = w$ then $\sigma(\rho)(q^2_\sharp) = 1$ if $q \in F$, and $\sigma(\rho)(q^1_\sharp) = 1$ otherwise.

At the beginning the system will move with probability $\frac{1}{2}$ to $\hat{A}$, where the strategy ensures that the word $w\sharp$ is observed. This leads to the state $q^1_\sharp$ with probability $\frac{1}{2}\mathbb{P}^F_A(w)$ and thus the next observations belong to $\flat\sharp^*$ and the runs with observations in $w\sharp\flat\sharp^+$ belong to the secret. On the other hand, the system can also go to s_1 with probability $\frac{1}{2}$ from where, due to the decisions of the strategy, a run with observation $w\sharp$ ends in $s_\sharp$ (the decisions of the strategy ensure that either the run does not have observation $w\sharp$, or it could not go in a r state). Moreover, from $s_\sharp$, $\flat$ cannot be observed. This implies that $w\sharp\flat\sharp$ is a min-disclosing observation in $\mathsf{M}_{A,\sigma}$, hence $\mathrm{Disc}(\mathsf{M}_{A,\sigma}) \geq \frac{1}{2}\mathbb{P}^F_A(w) > \frac{1}{4}$. Since $\mathrm{Disc}_{\max}(\mathsf{M}_A) = \sup_\sigma \mathrm{Disc}(\mathsf{M}_{A,\sigma})$, we can conclude that $\mathrm{Disc}_{\max}(\mathsf{M}_A) > \frac{1}{4}$.

Conversely, suppose that the disclosure is strictly greater than $\frac{1}{4}$ and let σ be a strategy such that $\mathrm{Disc}(\mathsf{M}_{A,\sigma}) > \frac{1}{4}$. Then, σ must forbid states in $\{r_c \mid c \in \{a, b, \sharp\}\}$, otherwise there would be no disclosing observation since every observation can be simulated once a state r_c is reached. Writing $\overline{\Sigma} = \{a, b, \sharp\}$, we inductively define the word $\overline{w} \in \overline{\Sigma}^\infty \cup \overline{\Sigma}^*$ by a sequence $(\overline{w_i})_{i \geq 0}$ of non-decreasing prefixes of $\overline{w}$:

- We start with $\overline{w_0} = \varepsilon$;
- Assume $\overline{w_i}$ is built and let ρ_i be a run ending in state s_x for some $x \in \{1, a, b, \sharp\}$, with $\mathsf{O}(\rho_i) = \overline{w_i}$. If $\sigma(\rho_i)(r_c) = 0$ for some $c \in \{a, b, \sharp\}$, then $\overline{w_{i+1}} = \overline{w_i}c$, otherwise $\overline{w_{i+1}} = \overline{w_i}$.

The set of ambiguous observations (*i.e.* corresponding to both secret and non-secret runs) are the ones reaching the set of states $\{r_c \mid c \in \{a, b, \sharp, \flat\}\}$:

$$\bigcup_{\substack{\overline{w_i}x \neq \overline{w_{i+1}} \\ x \neq \flat}} \overline{w_i}x(\overline{\Sigma} \cup \{\flat\})^*\sharp^\omega.$$

Hence, the set of disclosing observations is reduced to either $w\sharp\flat\sharp^\omega$, where w is the largest prefix of $\overline{w}$ in $\{a, b\}^*$ if $\sharp$ occurs in $\overline{w}$, and empty otherwise. Since the disclosure is greater than 0, we obtain $w\sharp\flat\sharp$ as the single min-disclosing observation with $\mathrm{Disc}(\mathsf{M}_{A,\sigma}) = \mathbb{P}_{\mathsf{M}_{A,\sigma}}(w\sharp\flat\sharp)$. Since $\mathbb{P}^F_A(w) \geq 2.\mathbb{P}_{\mathsf{M}_{A,\sigma}}(w\sharp\flat\sharp)$, we can conclude that $\mathbb{P}^F_A(w) > \frac{1}{2}$.

The proof can be extended with a non-erasing observation function by replacing ε with a fresh symbol (like in the proof of Theorem 7.1). This requires to slightly modify the parts of the IMC corresponding to the sets of states $\{s_c \mid c \in \{a, b, \sharp\}\}$ and $\{r_c \mid c \in \{a, b, \sharp\}\}$ in order to ensure alternation of letters from $\{a, b\}$ and this new symbol.

The undecidability result holds even when restricted to finite-memory strategy as the strategy defined in the first direction of the proof only uses finite memory. □

If we could compute the maximal finite-horizon disclosure, we could solve the associated decision problem. Thus, as a consequence of the previous theorem, we obtain:

Corollary 7.1. *The maximal finite-horizon disclosure of an OMDP cannot be computed.*

We now turn to the qualitative disclosure problems, and using a reduction from the value 1 problem in PA, we also have:

Theorem 7.3. *The maximal finite-horizon limit-sure disclosure problem is undecidable for OMDP.*

Proof. The reduction from the value 1 problem for PA done here is similar to the one of the proof of Theorem 7.2. The difference is that any run initially moving from s_0 to s_1 (thus moving to the part of the IMC which was used to limit the power of the strategy) will now almost-surely disclose the secret. More precisely, the construction of M_A depicted in Figure 7.8 for the proof of Theorem 7.2 is slightly modified as follows (see Figure 7.9): a new state $q_\natural$ with $\mathsf{O}(q_\natural) = \natural$ is added in the upper part just before reaching the secret state q_s. In this case, the runs reaching the secret in the upper part disclose the secret as they end with $\natural\sharp^\omega$.

The disclosure on the bottom part is performed as before. As a consequence, if a word w is "selected" by the strategy, the finite-horizon disclosure will be equal to $1/2.(1 + \mathbb{P}^F_A(w))$. This value can be arbitrarily close to 1 iff A accepts words with probabilities arbitrarily close to 1, which yields the result. □

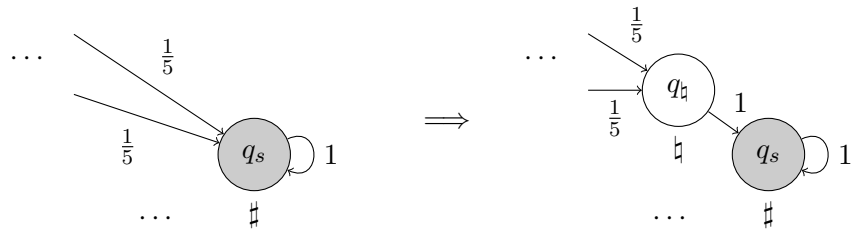


Figure 7.9: Modification of Figure 7.8 for limit disclosure.

2.3 Decidability of the almost-sure disclosure problem

Fortunately the maximal finite-horizon almost-sure disclosure problem is decidable. The proof relies on results for partially observable MDP (POMDP) described in Definition 6.8, page 186.

Theorem 7.4. *The maximal finite-horizon almost-sure disclosure problem in OMDP is* EXPTIME*-complete. Moreover, if the system is almost-surely disclosing, one can build a belief-based strategy with disclosure* 1*.*

We reduce the almost-sure disclosure problem for maximisation in OMDP to almost-sure reachability in POMDP. The POMDP we build is exponential in the size of the

original OMDP and the algorithm to solve almost-sure reachability is exponential in the size of the POMDP [CDH10]. A naive application of the two successive results would give a 2-EXPTIME algorithm. However, a finer analysis yields EXPTIME complexity of our algorithm as these two exponentials do not stack. The hardness is obtained by a reduction from the safety problem in games with imperfect information that was shown to be EXPTIME-complete in [BD08]. The reduction for the lower bound is similar to the one of Theorem 7.2.

Proof. We start by giving the construction of the POMDP, then we show that solving almost-sure reachability in the POMDP is equivalent to the finite-horizon almost-sure disclosure problem for maximisation. Finally we prove the hardness.

Construction of the POMDP. We start from an OMDP $\mathsf{M} = (S, \mathsf{Act}, p, \mathsf{O})$ with $\mathsf{Act} = \cup_{s\in S} A(s)$, observation alphabet Σ_o, and a set of secret states $\mathrm{Sec} \subseteq S$. Thanks to Proposition 7.2, we can assume O to be non-erasing (the potentially exponential blow up of the number of actions does not affect the complexity result as we will see later). Let μ_0 be an initial distribution. We assume w.l.o.g. that μ_0 is a Dirac distribution on some state $s_0 \in S$. We build a POMDP $\mathsf{M}' = \langle Q, q_0, \mathsf{Obs}, \mathsf{Act}, T\rangle$ with set of states $Q = S \times 2^S$, with $q_0 = (s_0, \{s_0\})$ and observation alphabet Σ_o. The observation function Obs is defined by $\mathsf{Obs}((s, B)) = \mathsf{O}(s)$. The set of actions of M' is $\mathsf{Act}' = \mathsf{DR}$ where DR is the set of vectors of deterministic decision rules $\vec{\delta}$ over S. Given a state $(s, B) \in S'$, an action $\vec{\delta} \in \mathsf{DR}$ and an observable state s', we have

$$T((s,B),\vec{\delta})(s',B') = \begin{cases} p(s'|s,\vec{\delta}(s)), \text{ for } B' = \{s'' \mid \mathsf{O}(s'') = \mathsf{O}(s') \wedge \\ \qquad\qquad \exists \hat{s} \in B, p(s''|\hat{s},\vec{\delta}(\hat{s})) > 0\} \\ 0, \text{ otherwise.} \end{cases}$$

Correspondence between strategies. To show that M is almost-surely disclosing for Sec iff $\mathrm{Sec} \times 2^{\mathrm{Sec}}$ can almost-surely be reached in M', we establish a correspondence between the strategies of M and the scheduler of M'. From Proposition 7.3, we can restrict to deterministic observation-based strategies for M, and from [CDH10], we also restrict to deterministic schedulers for M'.

In both direction of the equivalence, given a strategy σ and a scheduler τ, we write $\mathbb{P}_\sigma$ (resp. $\mathbb{P}_\tau$) instead of $\mathbb{P}_{\mathsf{M}_\sigma(\mu_0)}$ (resp. $\mathbb{P}_{\mathsf{M}'(\tau)}$), and define, for $w \in \Sigma_o^*$ and $s \in S$, the sets of finite runs $R(w, s) = \{\rho \text{ finite run of } \mathsf{M}_\sigma(\mu_0) \mid \mathsf{O}(\rho) = w \wedge \mathsf{last}(\rho) = s\}$ and $R'(w, s) = \{\rho' \text{ finite run of } \mathsf{M}'(\tau) \mid \mathsf{O}(\rho') = w \wedge \mathsf{last}(\rho') = (s, B^\sigma_w)\}$.

• Let τ be a scheduler of M', defined on Σ_o^*. We define σ for any observation $w \in \Sigma_o^*$ and state $s \in S$ by $\sigma(w, s) = \tau(w)(s, B^\sigma_w)$. Note that a τ-compatible run ρ' of M' ends in a state (s, B) where $B = B^\sigma_w$ (the belief w.r.t. σ) if $\mathsf{O}(\rho') = w$. We have $\mathbb{P}_\sigma(R(w, s)) = \mathbb{P}_\tau(R'(w, s))$.

Now let $\mathsf{Reach}(\mathrm{Sec} \times 2^{\mathrm{Sec}})$ be the set of runs reaching $\mathrm{Sec} \times 2^{\mathrm{Sec}}$ in $\mathsf{M}'(\tau)$. Then we claim that $\mathrm{Disc}_{\max}(\mathsf{M}_\sigma(\mu_0)) = \mathbb{P}_\tau(\mathsf{Reach}(\mathrm{Sec} \times 2^{\mathrm{Sec}}))$. Indeed, an observation w discloses the secret under strategy σ iff all observable states reachable with observed sequence w belong to the secret, *i.e.* iff $B^\sigma_w \subseteq \mathrm{Sec}$. Thus the runs ρ' in M' with a disclosing observation for M are the ones for which $\mathsf{last}(\rho') \in \mathrm{Sec} \times 2^{\mathrm{Sec}}$. Therefore,

thanks to the earlier remark, we have that the probability of reaching $\mathrm{Sec} \times 2^{\mathrm{Sec}}$ in M' under strategy τ is also the probability of disclosing Sec in M under strategy σ.

• Conversely, given an observation-based strategy σ of M, we build a scheduler τ of M' as follows: we define the mapping $\tau(w) : \Sigma_o^* \to \mathsf{Dist}(\mathsf{Act})$ by $\tau(w)(s, B_w^\sigma) = \sigma(w, s)$ for any $s \in S$. With the same reasoning as above, we immediately get that the probability of reaching $\mathrm{Sec} \times 2^{\mathrm{Sec}}$ in M' under strategy τ is also the probability of disclosing Sec in M under strategy σ.

We can conclude that M is almost-surely disclosing if and only if the runs of M' reach almost-surely the set $\mathrm{Sec} \times 2^{\mathrm{Sec}}$. Moreover, if M' almost-surely reaches the set $\mathrm{Sec} \times 2^{\mathrm{Sec}}$, we can build a scheduler τ doing so. Using the transformation described above and the results from [CDH10], we extract from τ a belief-based strategy σ of M that almost-surely discloses the secret.

Let us argue that the whole algorithm is in EXPTIME. The exponential in the algorithm of [CDH10] comes from a determinisation of the system, which is already done in our transformation from M to M', and thus not required a second time. Moreover, the non-erasing assumption on the observation could have created exponentially many new actions, which are exactly the ones built by the use of a vector of decision rules in our construction. Thus the exponentials do not stack.

The hardness is shown with a reduction from safety games with imperfect information.

Definition 7.8. *A* safety game with imperfect information *is defined by a tuple* $\mathcal{G} = (L, \ell_0, \Sigma, \Delta, O, F, \mathsf{obs})$ *where*

- L *is a finite set of locations with initial location* $\ell_0 \in L$*;*
- Σ *is a finite alphabet;*
- $\Delta \subseteq L \times \Sigma \times L$ *is the transition relation such that for all* $\ell \in L$ *and* $a \in \Sigma$ *there exists at least one* ℓ' *with* $(\ell, a, \ell') \in \Delta$*;*
- O *is a finite set of observations, and* $F \subseteq O$ *are the final observations;*
- $\mathsf{obs} : L \to O$ *is the observation mapping.*

A safety game with imperfect information $\mathcal{G}$ is a turn-based game played by two players Control and Environment. It starts in location ℓ_0 with Control to play. In the first round, Control chooses a letter $a_0 \in \Sigma$, then Environment chooses a location ℓ_1 such that $(\ell_0, a_0, \ell_1) \in \Delta$ and Control only observes $o_1 = \mathsf{obs}(\ell_1)$. The next rounds are played similarly and Control wins if for all $i, o_i \notin F$. The problem of existence of a winning strategy for Control is EXPTIME-complete [BD08]. We now describe a reduction from this problem to the almost-sure disclosure problem of OMDP.

The reduction is similar to the one in the proof of Theorem 7.2 except that we replace the probabilistic automaton by a safety game $\mathcal{G} = (L, \ell_0, \Sigma, \Delta, O, F, \mathsf{obs})$ with imperfect information and directly build an OMDP $\mathsf{M} = (S, \mathsf{Act}, p, \mathsf{O})$ over alphabet $(O \cup \{\sharp, \flat, \natural\}) \cup \Sigma \times (O \cup \{\sharp, \flat\})$, with:

- $S = \{s_0, \ell_0, q^1_\sharp, q^2_\sharp, q_\flat, q^1_s, q^2_s\} \cup \{\ell_c \mid \ell \in L, c \in \Sigma \times O\} \cup \{s_c \mid c \in \Sigma \times (O \cup \{\sharp\})\} \cup \{r_c \mid c \in \Sigma \times (O \cup \{\sharp, \flat\})\}$;
- $\mathsf{Act} = \Sigma$;
- For all $a \in \Sigma, o \in O, \ell_c \in S, \ell' \in \mathsf{obs}^{-1}(o)$, $p(\ell'_{a,o} \mid \ell_c, a) > 0$ iff $(\ell, a, \ell') \in \Delta$. If $\ell \in \mathsf{obs}^{-1}(F)$ then $p(q^1_\sharp \mid \ell_c, a) > 0$ and if $\ell \notin \mathsf{obs}^{-1}(F)$ then $p(q^2_\sharp \mid \ell_c, a) > 0$. For all $a \in \Sigma, c \in \{0\} \cup (\Sigma \times (O \cup \{\sharp\}))$, $(b', o') \in \Sigma \times (O \cup \{\sharp\})$, $p(s_{(b',o')} \mid s_c, a) > 0$ and if $b' \neq a$, $p(r_{(b',o')} \mid s_c, a) > 0$. For all $c, c' \in \Sigma \times (O \cup \{\sharp\})$, $p(r_{c'} \mid r_c, a) > 0$ and $p(q^2_s \mid r_c, a) > 0$. For all $a, a' \in \Sigma$, $p(q^1_s \mid q^1_\sharp, a) = p(q_\flat \mid q^2_\sharp, a) = p(q^1_s \mid q_\flat, a) = p(q^1_s \mid q^1_s, a) = p(q^2_s \mid q^2_s, a) = 1$.
- $\mathsf{O}(s_0) = \mathsf{O}(\ell_0) = \mathsf{obs}(\ell_0)$; For $z \in L, s, r$, $a \in \Sigma, o \in O$, $\mathsf{O}(z_{a,o}) = (a, o)$; For $o \in \{\sharp, \flat\}$, $\mathsf{O}(z_{a,o}) = o$, and $\mathsf{O}(q^1_\sharp) = \mathsf{O}(q^2_\sharp) = \sharp = \mathsf{O}(q^1_s)$, $\mathsf{O}(q_\flat) = \flat$, $\mathsf{O}(q^2_s) = \natural$.

The initial distribution is $\mu_0(s_0) = 1/2 = \mu_0(\ell_0)$ and the secret is $\mathrm{Sec} = \{q^1_s, q^2_s\}$.

This proof being similar to the one of Theorem 7.2, we only detail here the differences. A run starting in s_0 will almost surely trigger a $\natural$ and disclose the secret. A run starting in ℓ_0 will almost surely reach q^1_s as after any action in the copy of $\mathcal{G}$ there is a positive probability to reach $q^1_\sharp$ or $q^2_\sharp$. In order for a finite run starting in ℓ_0 to disclose the secret, it cannot go through $q^1_\sharp$ and should not share its observed sequence with a run ending in a state r_c. Given a strategy σ of M, if there exists a σ-compatible run ρ visiting a state ℓ_c with an observation $\mathsf{O}(\ell_c) \in \Sigma \times F$, then there is a σ-compatible path ρ' visiting $q^1_\sharp$, therefore a set of runs with positive probability do not visit the secret. Thus a deterministic strategy almost surely disclosing the secret in M never visits a state triggering an observation of the form $\Sigma \times F$. Moreover such a strategy does not take the current state into account. Indeed, let ρ and ρ' be two runs such that $\mathsf{O}(\rho) = \mathsf{O}(\rho')$ and ending in two states ℓ_c and s_c. If $\sigma(\rho) = a$ and $\sigma(\rho') = a'$ are two actions in Σ with $a \neq a'$ then there exists $o \in O$ such that $\rho a \ell_{a,o}$ is a σ-compatible run. Since $a \neq a'$, $\rho' a r_{a,o}$ is also a σ-compatible run with same observation than $\rho a \ell_{a,o}$. Hence no observation prefixed by $\mathsf{O}(\rho a \ell_{a,o})$ would disclose the secret.

Therefore, similarly as in Theorem 7.2, Control has a winning strategy iff there exists a deterministic strategy considering only the observed sequence that almost-surely discloses the secret. This implies EXPTIME-hardness of the maximal finite-horizon almost-sure disclosure problem. □

In this section on the maximisation of the disclosure over finite horizon, we saw that although we could restrict ourselves to deterministic strategies, most of the finite-horizon problems are very complicated (all but one are undecidable).

3 Minimisation with finite horizon

We now turn to minimisation over finite horizon where strategies try to hide the secret from an observer, and thus to minimise the disclosure.

This section shows a surprising result. Indeed, in Subsection 3.1 we show that for minimisation we cannot assume strategies to be deterministic any more, thus the problem seems more complex. However, the disclosure value can be computed as we establish in Subsection 3.2.

3.1 Deterministic strategies are not enough

After the proof of Proposition 7.3, we remarked that the proof showed that deterministic strategies do not decrease the disclosure, thus the proof could only work for maximisation. In fact, the result itself is limited to maximisation as randomisation may be necessary to minimise the disclosure. Let us see that on an example. Consider the OMDP depicted in Figure 7.10 with $\mathrm{Sec} = \{q_2, q_3\}$. There are two deterministic strategies, choosing respectively a or b in q_0. In both cases, the disclosure is $\frac{1}{2}$. On the other hand, for randomized strategies σ_p such that $\sigma_p(q_0) = pa + (1-p)b$ with $0 < p < 1$, there are no disclosing observations, hence the disclosure is 0.

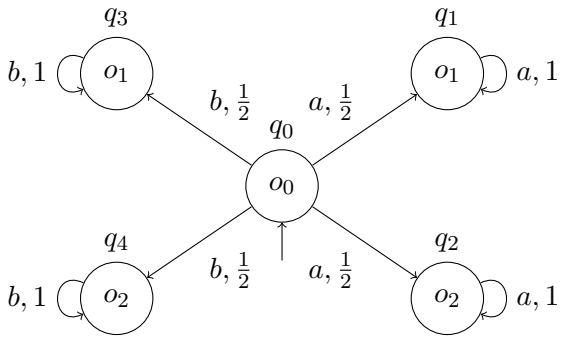


Figure 7.10: With $\mathrm{Sec} = \{q_2, q_3\}$, deterministic strategies are not sufficient for minimisation.

While we cannot restrict ourselves to deterministic strategies, we still use in the decision procedures a restricted class of strategies. These strategies are called *families of almost-deterministic strategies* and are based on *ε-decision rules*.

Definition 7.9. *Let δ be the deterministic decision rule for state s selecting action $a \in A(s)$. Then, for $\varepsilon > 0$, the* ε-decision rule $\delta_\varepsilon \in \mathsf{Dist}(A(s))$ *is defined by:*

1. *If $|A(s)| > 1$ then $\delta_\varepsilon(a) = 1 - \varepsilon$ and for all $b \in A(s) \setminus \{a\}$, $\delta_\varepsilon(b) = \frac{\varepsilon}{|A(s)|-1}$;*
2. *Else $\delta_\varepsilon = \mathbf{1}_a$.*

δ_ε is said to favour *a.*

ε-decision rules are used to define approximations of deterministic strategies.

Definition 7.10. *Let σ be an observation-based deterministic strategy. Then $\{\sigma_\varepsilon\}_{\varepsilon>0}$ is a family of* observation-based almost-deterministic strategies *defined for any state s and $w \in \Sigma^n$ by: $\sigma_\varepsilon(w, s) = \sigma(w, s)_{2^{-n}\varepsilon}$.*

In other words, given a strategy σ we define a family of strategies that have an increasingly high probability to play like σ as the run goes on, yet always allow the other actions with positive probability. We will see that strategies of this form are dominant in the sense that they are sufficient to compute the disclosure value. However, if there exists a strategy that minimises the disclosure, this strategy does not always belong to a family of almost-deterministic strategy.

3.2 The minimal disclosure value is computable

Using Proposition 7.2, we assume in the following that the observation function we consider is non-erasing. The complexity of the transformation does not affect the results since the polynomial complexity in the number of actions is dominated by the exponential complexity in the number of states.

In order to compute the minimal disclosure value, we build from an OMDP M, another OMDP $\mathsf{M}_{\min}$ which is a "correct abstraction" (as is stated by Proposition 7.4) for reducing minimal disclosure problems to minimal reachability problems. Intuitively, in $\mathsf{M}_{\min}$, the states are enlarged by the maximal belief that can occur independently of the action that has been selected.

Given a set of potential current states B and a new observation o, we define the maximal set of potential next states $\mathsf{NextMax}(B, o)$ over decision rules applied to B by:

$$\mathsf{NextMax}(B, o) = \{s' \in \mathsf{O}^{-1}(o) \mid \exists s \in B \ \exists a \in A(s) \ p(s'|s, a) > 0\}.$$

$\mathsf{NextMax}$ is intuitively the belief obtained with a strategy allocating some probabilities to every action. Observe that given a family of almost deterministic strategies $\{\sigma_\varepsilon\}$ and a run ρas of M with $\mathsf{O}(s) = o$, one has for all $\varepsilon > 0$, $B^{\sigma_\varepsilon}_{\rho as} = \mathsf{NextMax}(B^{\sigma_\varepsilon}_{\rho}, o)$. Then $\mathsf{M}_{\min}$ is formally defined as follows:

- $S_{\min}$, the set of states, is defined by: $S_{\min} = \{(s, B) \mid s \in B \subseteq \mathsf{O}^{-1}(\mathsf{O}(s))\}$;
- for every $(s, B) \in S_{\min}$, $A(s, B) = A(s)$;
- for every $(s, B), (s', B') \in S_{\min}$,

$$p((s', B')|(s, B), a) = \begin{cases} p(s'|s, a) \text{ if } B' = \mathsf{NextMax}(B, \mathsf{O}(s')), \\ 0 \text{ otherwise;} \end{cases}$$

- for every $(s, B) \in S_{\min}$, $\mathsf{O}(s, B) = \mathsf{O}(s)$.

Given μ_0 an initial distribution over S, the associated initial distribution $\mu_{\min}$ over $S_{\min}$ is defined by $\mu_{\min}(s, \mathsf{Supp}(\mu_0) \cap \mathsf{O}^{-1}(\mathsf{O}(s))) = \mu_0(s)$ and $\mu_{\min}(s, B) = 0$ for all other B. We define the subset $\mathsf{Avoid}(\mathrm{Sec}) \subseteq S_{\min}$ by $\mathsf{Avoid}(\mathrm{Sec}) = \{(s, B) | B \subseteq \mathrm{Sec}\}$.

Proposition 7.4. *The minimal disclosure value for* Sec *in* $\mathsf{M}(\mu_0)$ *is equal to the minimal probability to reach* $\mathsf{Avoid}(\mathrm{Sec})$ *in* $\mathsf{M}_{\min}(\mu_{\min})$. *Furthermore it is asymptotically reached by a family of belief-based almost deterministic strategies.*

We show this result in two steps. First we show that, using families of belief-based almost-deterministic strategies, one can obtain a disclosure value in M arbitrarily close to the minimal reachability probability in $\mathsf{M}_{\min}$. This ensures that the disclosure value is below this probability. Second, we show that the disclosure obtained by an arbitrary strategy is greater or equal to the probability to reach $\mathsf{Avoid}(\mathrm{Sec})$.

Proof. We know that the minimal reachability probability for $\mathsf{Avoid}(\mathrm{Sec})$ in $\mathsf{M}_{\min}(\mu_{\min})$ is obtained by a memoryless deterministic strategy $\sigma_{\min}$ that selects some action $a_{s,B}$ in state (s, B) (see *e.g.* [BK08]). Consider $\{\sigma_\varepsilon\}$ the family of belief-based almost-deterministic strategies defined by favouring $a_{s,B}$ in state s after a run ρ such that $B_\rho^{\sigma_\varepsilon} = B$. Given a run $\rho = s_0 a_0 \dots a_{n-1} s_n$ in $\mathsf{M}(\mu_0)$ we inductively define the run $b(\rho) = (s_0, S_0)a_0 \dots a_{n-1}(s_n, S_n)$ in $\mathsf{M}_{\min}(\mu_{\min})$ by: $S_0 = \mathsf{Supp}(\mu_0) \cap \mathsf{O}^{-1}(\mathsf{O}(s_0))$ and $S_{i+1} = \mathsf{NextMax}(S_i, \mathsf{O}(s_{i+1}))$. Due to the observation given when introducing $\mathsf{NextMax}$, with strategy σ_ε, the observation of run ρ discloses the secret iff $b(\rho)$ reaches $\mathsf{Avoid}(\mathrm{Sec})$. Consider, under strategy σ_ε, the probability to disclose the secret with runs ρ such that $b(\rho)$ includes at least once an action not selected by $\sigma_{\min}$. By construction, at each step i, the probability of not choosing the action favoured by σ_ε is $\frac{\varepsilon}{2^i}$, hence the probability of this set of runs is $\sum_{i\geq 0}(1-\varepsilon)^i \frac{\varepsilon}{2^i} \leq 2\varepsilon$. Consider now a finite run $s_0 a_0 \dots a_{n-1} s_n$ such that $b(\rho)$ is $\sigma_{\min}$-compatible. Then the probability of the original run is less than or equal to the probability of its corresponding run. So we deduce that the minimal disclosure value of $\mathsf{M}(\mu_0)$ is bounded above by $\nu + 2\varepsilon$ where ν is the minimal reachability probability for $\mathsf{Avoid}(\mathrm{Sec})$ in $\mathsf{M}_{\min}(\mu_{\min})$. Since this holds for all $\varepsilon > 0$, we obtain that the minimal disclosure value of $\mathsf{M}(\mu_0)$ is bounded above by the minimal reachability probability for $\mathsf{Avoid}(\mathrm{Sec})$ in $\mathsf{M}_{\min}(\mu_{\min})$.

Conversely consider an arbitrary strategy σ in $\mathsf{M}(\mu_0)$. This strategy may also be applied in $\mathsf{M}_{\min}(\mu_{\min})$ by forgetting the second component of the state, defining a strategy σ'. For any run $s_0 a_0 \dots s_n$ in $\mathsf{M}_\sigma(\mu_0)$, there is a single run $(s_0, S_0)a_0 \dots (s_n, S_n)$ in $\mathsf{M}_{\min}(\mu_{\min})$ under σ' with the same probability. Given the run $s_0 a_0 \dots s_n$, consider the successive associated subsets of beliefs according to σ, $B_0, \dots, B_n$. By induction (and definition of $\mathsf{M}_{\min}$) it is straightforward to show that $B_i \subseteq S_i$. So $s_0 a_0 \dots s_n$ does not disclose the secret in M under σ implies that $(s_0, S_0) \dots (s_n, S_n)$ does not reach $\mathsf{Avoid}(\mathrm{Sec})$. This entails that the reachability probability of $\mathsf{Avoid}(\mathrm{Sec})$ in $\mathsf{M}_{\min}(\mu_{\min})$ under σ' is less than or equal to the disclosure probability in $\mathsf{M}(\mu_0)$ under σ. □

Since minimal reachability probability in OMDP can be computed in polynomial time(see *e.g.* [BK08])[4] we immediately obtain the first part of the next theorem. We establish the second part (PSPACE-hardness) in the proof of Theorem 7.8 as the proof holds also for disclosure over fixed-horizon.

Theorem 7.5. *The minimal disclosure value of* $\mathsf{M}(\mu_0)$ *can be computed in* $\mathsf{EXPTIME}$. *The associated decision problem is* PSPACE*-hard.*

[4]Note that since observations are not useful for this reachability objective, observations could be removed from the OMDP $\mathsf{M}_{\min}$, yielding a Markov decision process, which is the model studied in [BK08].

We now turn to the existence of a strategy that achieves the minimal value. As remarked earlier, we cannot assume such a strategy belongs to an almost-deterministic family of strategies. We establish that it can be analysed without additional complexity.

Theorem 7.6. *The existence of a strategy that achieves the minimal disclosure value can be decided in* EXPTIME*. In the positive case, this strategy can be computed in* EXPTIME*.*

The main ingredient of the proof is an elimination algorithm that removes iteratively the beliefs from which no strategy can reach the maximal disclosure. Once al these beliefs were removed, if the initial belief was not deleted, then there exists a belief-based strategy minimising the disclosure and this strategy plays in order to stay within the beliefs kept by the algorithm.

Proof. Let us first introduce multiple notations that are used within the proof. We define $\mathrm{disc}^*(\mathsf{M}(s,B))$ as the minimal disclosure value when starting in M in state s with belief B. Given some belief B and some decision rule vector $\vec{\delta}$ over B we introduce the possible successors of B when applying $\vec{\delta}$: $\mathsf{Next}(B,\vec{\delta}) = \{s' \mid \exists s \in B\ \exists a \in \mathsf{Supp}(\vec{\delta}[s])\ p(s'|s,a) > 0\}$ and $\mathsf{Next}(B,\vec{\delta},o) = \mathsf{Next}(B,\vec{\delta}) \cap \mathsf{O}^{-1}(o)$.

The algorithm simultaneously solves the existence and the synthesis problem.
First, using Proposition 7.4, the algorithm computes for all $(s,B) \in S_{\min}$, $\mathrm{disc}^*(\mathsf{M}(s,B))$. Then it maintains a set Win of beliefs initially set to all beliefs from which it iteratively eliminates items and stops when no more elimination is possible. Given $B \in \mathsf{Win}$, it looks for a decision rule vector $\vec{\delta}$ over B such that:

- for all $o \in \mathsf{O}(\mathsf{Next}(B,\vec{\delta}))$, $\mathsf{Next}(B,\vec{\delta},o) \in \mathsf{Win}$;
- for all $s \in B$, $\mathrm{disc}^*(\mathsf{M}(s,B)) = \sum_{o\in\Sigma} \sum_{s'\in\mathsf{O}^{-1}(o)} p(s'|s,\vec{\delta}[s])\mathrm{disc}^*(\mathsf{M}(s',\mathsf{Next}(B,\vec{\delta},o)))$.

If such a $\vec{\delta}$ does not exist then B is eliminated from Win. In other words, a belief is eliminated if there does not exist a decision rule that meets the minimal disclosure value. Each iteration can be performed in polynomial time w.r.t. $|S_{\min}|$ and the number of iterations is at most $|S_{\min}|$. Observe that when a belief is eliminated, it should not be "reached" by a strategy that obtains the minimal disclosure value. So the elimination is sound.
When the elimination stops, the algorithm answers positively iff for all $o \in \mathsf{O}(\mathsf{Supp}(\mu_0))$, $\mathsf{Supp}(\mu_0) \cap \mathsf{O}^{-1}(o) \in \mathsf{Win}$. Thus, by the soundness of the elimination step, if the answer is negative there is no optimal strategy for minimal disclosure value.
If the answer is positive, let us consider the belief-based strategy σ defined by applying the decision rules obtained during the last iteration of the algorithm. On the one hand, under σ when visiting a state s with belief B such that $\mathrm{disc}^*(\mathsf{M}(s,B)) = 0$, one never leaves such kind of pairs of states and beliefs. So the secret is never disclosed, showing that the disclosure value obtained by σ for such (s,B) is null. Under σ, the disclosure value of all the other pairs of states and beliefs fulfil the equations of the elimination step. It is known that the single solution of this system is the vector of minimal reachability probabilities of Avoid in $\mathsf{M}_{\min}(\mu_{\min})$ (see [BK08] for instance) which yields the result. □

4 Fixed-horizon problems

We focus now on fixed-horizon problems for both maximal (Subsection 4.1) and minimal (Subsection 4.2) disclosure. In both cases, the algorithms and hardness results have similarities.

4.1 Maximal disclosure

In order to compute the value of the maximal disclosure within a fixed horizon, one could build the POMDP described in the proof of Theorem 7.4, then use pre-existing results on POMDPs. This would result in an EXPTIME algorithm, whereas we provide below an algorithm with a better complexity in PSPACE.

Theorem 7.7. *The fixed-horizon maximal value (when the horizon n is described in unary representation) is computable in* PSPACE *and the fixed-horizon maximal disclosure problem is* PSPACE*-complete.*

Due to the complexity of the proof, we separate the algorithm computing the value from the hardness of the decision problem in two separate proofs.

In order to compute the value, we first order the observation alphabet Σ. Then, a non-deterministic decision procedure operating in PSPACE enumerates every observed sequence of length n while maintaining the sets of states that were possible after every prefix of this observation, the actions that were chosen non-deterministically in these states and values used in the computation of the disclosure. The information kept is of polynomial size and when every observation has been read, one of the values computed are exactly the disclosure of the system at time n. This provides an NPSPACE algorithm which can be turned in to a PSPACE one using Savitch's Theorem [Sav70]. In order to get the value we observe that we can compute the polynomially sized denominator of this value and then we make iterative calls to the decision algorithm.

Proof. We first present a non-deterministic procedure that decides in NPSPACE the disclosure problem. It can then be determinised using Savitch's Theorem [Sav70].

From an arbitrarily ordered observation alphabet Σ, the procedure operates as follows for horizon n:

- It maintains a disclosure value v, a sequence of observations $o_0 \cdots o_i$ with $i \leq n$, a sequence of sets of states $B_1 \cdots B_i$ with $B_j \subseteq \mathsf{O}^{-1}(o_j)$ for all $j \leq i$, an action $a_{j,s} \in A(s)$ for all (j, s) with $j < i$ and $s \in S_j$, and for all (j, s) with $j \leq i$ and $s \in B_j$ the probability $p_{j,s}$ to reach s after the sequence of observations $o_0 \cdots o_i$;
- Initially $v = 0$, o_0 is the smallest observation in $\mathsf{O}(\mathsf{Supp}(\mu_0))$, where μ_0 is the initial distribution, $B_0 = \mathsf{Supp}(\mu_0) \cap \mathsf{O}^{-1}(o_0)$ and $p_{0,s} = \mu_0(s)$ for $s \in B_0$;
- If $i < n$ then for all $s \in B_i$, the procedure *guesses* an action $a_{i,s} \in A(s)$. Let o_{i+1} be the smallest observation such that there exists a state $s \in B_i$ and a state $s' \in \mathsf{O}^{-1}(o_{i+1})$ with $p(s'|s, a_{i,s}) > 0$. Then B_{i+1} is set to $\{s' \in \mathsf{O}^{-1}(o_{i+1}) \mid \exists s \in B_i \; p(s'|s, a_{i,s}) > 0\}$ and for all $s' \in B_{i+1}$, $p_{i+1,s'} = \sum_{s \in B_i} p_{i,s} p(s'|s, a_{i,s})$;

- If $i = n$, the procedure examines B_n. If $B_n \subseteq \mathrm{Sec}$ then $v = v + \sum_{s \in B_n} p_{n,s}$ otherwise v is unchanged. Afterwards it "backtracks" to the greatest $0 < i \leq n$ such that there exists $o'_i > o_i$ with some $s \in B_{i-1}$ and a state $s' \in \mathsf{O}^{-1}(o'_i)$ with $p(s'|s, a_{i-1,s}) > 0$. Then B_i and the $p_{i,s'}$'s are updated accordingly and the procedure carries on. If there is no such i, the procedure returns to $i = 0$ and similarly looks for some $o'_0 > o_0$, where the initialisation step is again performed except for the value of v which is unchanged. When the maximal observation in $\Sigma \cap \mathsf{O}(\mathsf{Supp}(\mu_0))$ is handled, the procedure terminates by comparing v to the threshold.

The correctness of the procedure follows from the fact that there exists an optimal deterministic strategy where the selection of the action for the current state depends only on the sequence of observations (and not on the sequence of visited states).

The space complexity of the procedure is in $O(n|S|(\log(|A|) + nb))$ where b is the maximal number of bits used to represent a transition probability of the OMDP.

Observe now that, since the maximal value is obtained by a deterministic strategy, one knows a denominator of this value: it is d^n where d is the least common multiple of the denominators of the probabilities occurring in the model. Its bit size is polynomial w.r.t. the size of the model. So by iteratively solving the disclosure problem for $\frac{i}{d^n}$ for increasing values of i, one computes the maximal value in PSPACE. □

As can be seen in the proof, the optimal strategy could be computed when solving the value problem. However the size of this strategy may be exponential due to the beliefs and thus this strategy is computable in EXPTIME.

For the hardness result, we reduce the validity of Quantified Boolean Formulae (QBF). Recall that QBF extends propositional formulas by allowing quantification over the Boolean variables. Syntactically, the formulae are described by the following grammar:

$$\phi ::= \psi \mid \exists x.\phi \mid \forall x.\phi$$
$$\psi ::= x \mid \psi \wedge \psi \mid \psi \vee \psi \mid \neg\psi \mid \texttt{true}$$

A QBF is *closed* if every Boolean variable is bound by a quantifier. Deciding if a closed QBF is *valid* (*i.e.* equivalent to `true`) is PSPACE-hard [Sip06].

The idea of the reduction is the following. Given ϕ a closed QBF (w.l.o.g. in 3CNF with n variables and m clauses), we build an OMDP M such that ϕ is true iff the disclosure of M is greater or equal to $\frac{1}{2^{2n}}$ in $2(n+m)+3$ steps. In fact, $\frac{1}{2^{2n}}$ is exactly the measure of runs reaching the secret in $2(n+m)+3$ steps, thus every path reaching the secret must be disclosing. Such a run discloses the secret iff some Boolean variable of ϕ and its negation (x and $\neg x$ for example) do not occur in its observation.

In M, during the first $2n$ steps, an assignment is 'given' to each Boolean variable: (i) for each existentially quantified Boolean variable x, the strategy chooses whether x or $\neg x$ occurs in the observation and (ii) for each universally quantified Boolean variable y, by a random choice with probability $\frac{1}{2}$. During the last $2m$ steps, the strategy must trigger a Boolean variable in every clause of ϕ so that if a clause is not satisfied by the

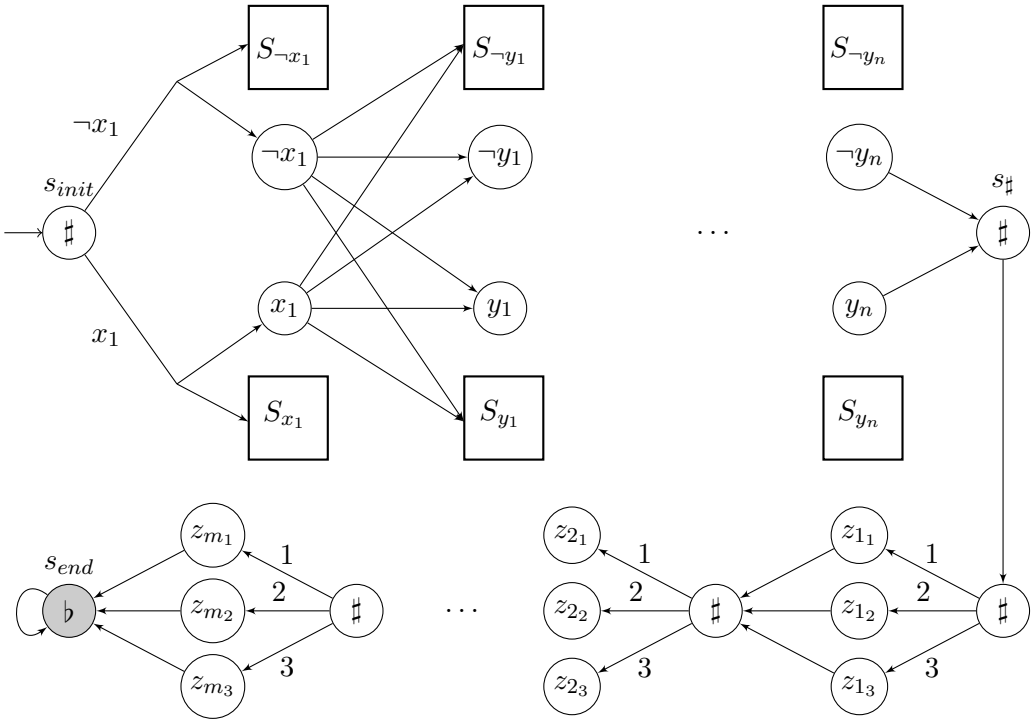


Figure 7.11: Reduction of the validity problem to the disclosure on a fixed horizon. The box S_{x_1} is represented in Figure 7.12.

current assignment, then a Boolean variable is observed as both true and false during the run. Thus the observation would not disclose the secret.

Proof. We reduce the validity of a quantified Boolean formula: Given a closed QBF in 3CNF $\phi = \exists x_1 \forall y_1 \exists x_2 \ldots \forall y_n \psi$ with $\psi = \bigwedge_{i=1\ldots m}(z_{i_1} \vee z_{i_2} \vee z_{i_3})$, we build an OMDP M such that ϕ is true if and only if $\mathrm{Disc}_{2(n+m)+3,\max}(\mathsf{M}) \geq \frac{1}{2^{2n}}$.

The OMDP $\mathsf{M} = (S, \mathsf{Act}, p, \mathsf{O})$ (depicted in Figure 7.11 with some conventions to avoid having too many edges) is defined by:

- $S = \{s_{init}, s_{end}, s_\sharp\} \cup \{s_{z_i} \mid z \in \{x, y\}, i = 1 \ldots n\} \cup \{s_{\neg z_i} \mid z \in \{x, y\}, i = 1 \ldots n\} \cup \{s_t^i \mid i \in \{1, \ldots, m\}\} \cup \{s^{z_{i_j}} \mid i \in \{1, \ldots, m\}, j \in \{1, 2, 3\}\} \cup_{i \in \{1,\ldots n\}} (S_{x_i} \cup S_{y_i} \cup S_{\neg x_i} \cup S_{\neg y_i})$ where for z_i one of the Boolean variable, $S_{z_i} = \{s_a^{z_i,1} \mid a \in \{\sharp, z_i, \neg z_i \mid z \in \{x, y\}, i \in \{1, \ldots, n\}\}\} \cup \{s_a^{z_i,2} \mid a \in \{\sharp, \flat, z_i, \neg z_i \mid z \in \{x, y\}, i \in \{1, \ldots, n\}\}\}$. Similar for the box $S_{\neg z_i}$ of the negation of a variable.

- $A(s_{init}) = \{x_1, \neg x_1\}$, and for all $i < n$, $A(s_{y_i}) = A(s_{\neg y_i}) = \{x_{i+1}, \neg x_{i+1}\}$. For $i \in \{1, \ldots, m\}$, $A(s_t^i) = \{1, 2, 3\}$ and $A(s) = \{next\}$ for all other states (even in boxes).

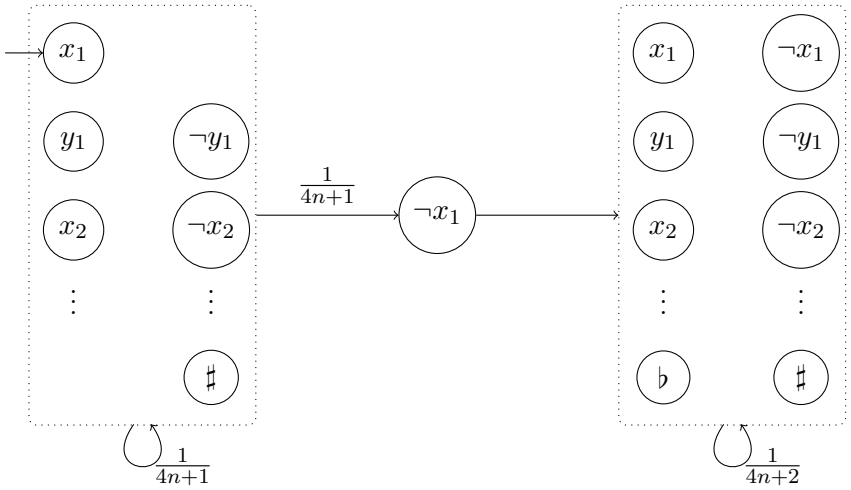


Figure 7.12: Representation of the box S_{x_1}. We use the convention of Figure 7.8.

- $p(s_a \mid s_{init}, a) = p(s_a^{a,1} \mid s_{init}, a) = 1/2$. For all $i < n$, $p(s_a \mid s_{y_i}, a) = p(s_a^{a,1} \mid s_{y_i}, a) = p(s_a \mid s_{\neg y_i}, a) = p(s_a^{a,1} \mid s_{\neg y_i}, a) = 1/2$. For all $i \leq n$, $p(s_{y_i} \mid s_{x_i}, next) = p(s_{y_i}^{y_i,1} \mid s_{x_i}, next) = p(s_{\neg y_i} \mid s_{x_i}, next) = p(s_{\neg y_i}^{\neg y_i,1} \mid s_{x_i}, next) = p(s_{y_i} \mid \neg s_{x_i}, next) = p(s_{y_i}^{y_i,1} \mid s_{\neg x_i}, next) = p(s_{\neg y_i} \mid s_{\neg x_i}, next) = p(s_{\neg y_i}^{\neg y_i,1} \mid s_{\neg x_i}, next) = 1/4$, and $p(s_t^1 \mid s_\sharp, next) = 1$. For all $i = 1 \ldots m, j \in \{1, 2, 3\}, p(s^{z_{i_j}} \mid s_t^i, j) = 1$, and if $i < m$, $p(s_t^{i+1} \mid s^{z_{i_j}}, next) = 1$. Finally, $p(s_{end} \mid s^{z_{m_j}}, next) = p(s_{end} \mid s_{end}, next) = 1$.

  We now describe p for the box S_{x_1} other boxes being similar. For all $a, b \in \{\sharp, x_1, z_i, \neg z_i \mid z \in \{x, y\}, i \in \{2, \ldots, n\}\}$, $p(s_b^{x_1,1} \mid s_a^{x_1,1}, next) = p(s_{\neg x_1}^{x_1,2} \mid s_a^{x_1,1}, next) = 1/(4n+1)$ and for all $c, d \in \{\sharp, \flat, z_i, \neg z_i \mid z \in \{x, y\}, i \in \{1, \ldots, n\}\}$, $p(s_d^{x_1,2} \mid s_c^{x_1,2}, next) = 1/(4n+2)$.

- $\mathsf{O}(s_{end}) = \flat, \mathsf{O}(s_a^b) = a$ and $\mathsf{O}(s^a) = a$ when a is a Boolean variable or its negation and for all other state s, $\mathsf{O}(s) = \sharp$.

The initial distribution μ_0 is $\mathbf{1}_{s_{init}}$ and the set of secret states is $\mathrm{Sec} = \{s_{end}\}$.

We show that ϕ is true iff the disclosure of M for observations of length $2(n+m)+3$ is greater than or equal to $\frac{1}{2^{2n}}$. First observe that for any strategy, the measure of runs reaching state s_{end} with observation of length $2(n+m)+3$ is exactly $\frac{1}{2^{2n}}$. Indeed, during each of the first $2n$ actions, whatever the choices of the strategy, there is a probability $\frac{1}{2}$ to go in one of the boxes and $\frac{1}{2}$ to advance to the next choice, thus a probability $\frac{1}{2^{2n}}$ to reach the state $s_\sharp$. From there every run reaches s_{end} in $2(m+1)$ steps. If the strategy is such that some variable and its negation are read on the way to s_{end}, then there exists a run with same observation reaching the second part of a box where every observation can be triggered, and thus the run reaching s_{end} will not disclose the secret.

Intuitively, during the first $2n$ steps, every Boolean variable is assigned a value:

either chosen by the strategy as it chooses whether x_i or $\neg x_i$ occurs in the observation for all $1 \leq i \leq n$, or randomly as y_i and $\neg y_i$ both have equal chance of being triggered. During the last $2m$ steps, the strategy must trigger a Boolean variable in every clause of the disjunction so that if a clause is not satisfied by the current assignment, then a Boolean variable is observed as both true and false during the run. Thus the observation would not disclose the secret. In order for a measure of $\frac{1}{2^{2n}}$ of runs to disclose the secret, for every assignment of the y_i the controller must force the run reaching s_{end} to disclose the secret.

Suppose that ϕ is equivalent to true. Thus there exist functions $(f_i)_{i=1\dots n}$ (expressing the choices for $x_1, \dots, x_n$) such that for every set of assignments $(a_1, \dots, a_n)$ of the variables $y_1, \dots, y_n$ the Boolean formula $\psi[f_1(), a_1, f_2(a_1), \dots, f_n(a_1, \dots, a_{n-1}), a_n]$ is true. We choose a strategy σ such that for every possible set of assignments $(a_1, \dots, a_n)$ for the variables $y_1, \dots, y_n$, for all $i \in \{0, \dots n-1\}$, $\sigma(\sharp f_1()a_1 f_2(a_1) \dots a_i) = f_{i+1}(a_1, \dots, a_i)$. Moreover for $i_1, \dots, i_k \in \{1, 2, 3\}$, there exists $z_{k+1_{i_{k+1}}} \in \{f_1(), a_1, f_2(a_1), \dots, a_n\}$ such that $\sigma(\sharp f_1()a_1 f_2(a_1) \dots a_n \sharp\sharp z_{1_{i_1}} z_{2_{i_2}} \dots z_{k_{i_k}}) = z_{k+1_{i_{k+1}}}$. The choice of the strategy is arbitrary in the other cases. Such a strategy can be defined since the formula $\psi[f_1(), a_1, f_2(a_1), \dots, f_n(a_1, \dots, a_{n-1}), a_n]$ is true and thus every clause is satisfied by this choice of assignments.

With this strategy, the fixed-horizon disclosure in $2(n+m+1)$ steps is $\frac{1}{2^{2n}}$. In other words, all the runs reaching the secret disclose it. Indeed let ρ be a secret run of length $2(n+m+1)$. There exists an assignment $a_1, \dots, a_n \in \{y_1, \neg y_1, \dots y_n, \neg y_n\}$ such that $\mathsf{O}(\rho) = \sharp f_1()a_1 f_2(a_1) \dots a_n \sharp\sharp z_{1_{i_1}} z_{2_{i_2}} \dots z_{m_{i_m}} \flat$. By choice of σ, if, for $z \in \{x, y\}$ and $i \in \{1, \dots n\}$, z_i appears in the observation of ρ, $\neg z_i$ does not appear, and vice versa. Therefore as $\flat$ can be read either in s_{end} or in a state reachable only by runs observing a Boolean variable and its negation, ρ discloses the secret.

Conversely, suppose that ϕ is not equivalent to true and let σ be a strategy, which can be assumed to be deterministic thanks to Proposition 7.3. We build partial functions $f_i : \Sigma^{2i} \mapsto \mathsf{Act}$ consistent with σ: for every observation $\sharp w \in \sharp\Sigma^{2i}$ of some run ρ, if σ chooses action $a \in A(\mathsf{last}(\rho))$ for ρ (i.e., $\sigma(\rho)(a) = 1$) then $f_i(w) = a$. As ϕ is not equivalent to true, there exists an assignment $(a_1, \dots, a_n)$ for the variables $y_1, \dots, y_n$ such that the Boolean formula $\psi[f_1(), a_1, f_2(a_1), \dots, f_n(a_1, \dots, a_{n-1}), a_n]$ is false. We now build a run with non null probability, reaching the secret but not disclosing it.

By construction, there exists ρ such that $\mathsf{O}(\rho) = \sharp f_1()a_1 f_2(a_1) \dots f_n(a_1 \dots a_{n-1})a_n \sharp$, with $\mathsf{last}(\rho) = s_\sharp$ and $\mathbb{P}_\sigma(\rho) > 0$ (where again $\mathbb{P}_\sigma$ stands for $\mathbb{P}_{\mathsf{M}_\sigma(\mu_0)}$). Let $i \in \mathbb{N}$ be an integer such that the clause $z_{i_1} \vee z_{i_2} \vee z_{i_3}$ is not true under the assignment $[f_1(), a_1, f_2(a_1), \dots, f_n(a_1, \dots, a_{n-1}), a_n]$, in other words such that the negations of z_{i_1}, z_{i_2} and z_{i_3} were chosen as assignment. Let ρ' be the run of length $2(n+m)+2$ extending ρ and ending in s_{end}. Then ρ' does not disclose the secret: indeed, there exists $j \in \{1, 2, 3\}$ such that z_{i_j} appears in its last $2m$ observations while its negation (written $\neg z_{i_j}$ here) appears in the first $2n+1$ observations. Thus there exists a run with same observation leading to the second part of the box $S_{\neg z_{i_j}}$ which is outside the secret and where every observation is possible. As the total measure of runs reaching the secret is $\frac{1}{2^{2n}}$ and at least a subset of measure $\mathbb{P}_\sigma(\rho)$ of the runs reaching the secret do not disclose it, the disclosure of M is strictly smaller than $\frac{1}{2^{2n}}$ in $2(n+m)+3$ observation

steps. □

The existence of an optimal strategy in the first part of the proof implies that the limit-sure and the almost-sure problem are equivalent. Moreover, the secret being revealed with probability 1 in a given number of steps, every run must reach the secret in this number of steps. Testing if there exists a strategy such that every run reaches a set of target states in a given number of steps in an OMDP can be solved in polynomial time.

Remark 7.1. *The proof of hardness can be adapted for maximal fixed-horizon ε-disclosure, but the algorithm for membership cannot be directly applied. The ε-disclosure could however be computed by maximising an exponential system of equations, resulting in an exponential time algorithm.*

4.2 Minimal disclosure

The proof of the next theorem is similar to the proof of Theorem 7.7 on the fixed-horizon maximal disclosure.

The hardness result is obtained once again using a reduction from the validity of a QBF. Many of the ideas used in the proof of Theorem 7.7 reappears here: the run is still composed of two parts, in the first one it gives an assignment to the Boolean variables and in the second one the strategies goes through the clauses of the formula and verify it can satisfy them. We however give the full proof due to non-negligible differences. Now the strategy must satisfy every clause of the formula in order for the run not to disclose the secret.

For the strategy decision problem, contrary to the maximisation case, due to the randomisation, there does not necessarily exists an optimal strategy. In order to get the same complexity for the strategy decision problem, we establish that when a randomised decision rule must be selected in the optimal strategy, it can always be uniformly distributed over its support.

Theorem 7.8. *The fixed-horizon minimal value is computable in* PSPACE. *The fixed-horizon minimal disclosure problem is* PSPACE*-complete. In addition, the strategy decision problem is also decidable in* PSPACE.

Proof. The procedures for the first two problems are very similar to the ones used in Theorem 7.7. There are only two differences. First, given B_i the current belief and o_{i+1} one computes $B_{i+1} = \mathsf{NextMax}(B_i, o_{i+1})$ (independently of the guessed actions $a_{i,s}$). Second, the computation procedure operates by decreasing values of i when the value is less or equal than $\frac{i}{d^n}$.

In order to decide whether a strategy exists that provides the minimal value, one guesses this strategy in PSPACE as before. However there is an additional difficulty since the (possible) optimal strategy may be randomised. Thus during the procedure, given some belief B and some state s, one guesses the support $A' \subseteq A_s$ of the decision rule and one defines the decision rule say δ as a uniform choice over A'. We claim that this restriction is sound. Assume another decision rule δ' with same support would provide

a smaller value. Then, since the support are unchanged, the decision rule informally described as $(1+\varepsilon)\delta' - \varepsilon\delta$ for small enough ε would still provide a better value, meaning that the support A' cannot be used to find an optimal strategy.

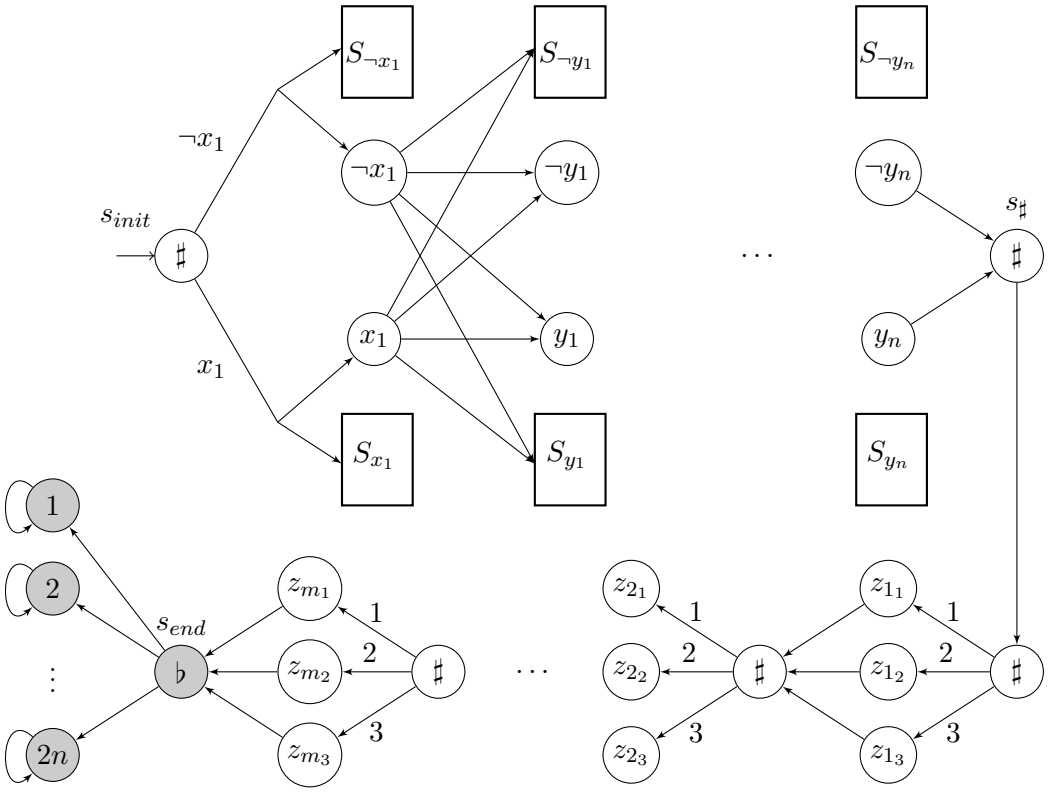


Figure 7.13: Reduction of the validity problem to the disclosure on a fixed horizon. The box S_{x_1} is represented in Figure 7.14

Like for the case of maximisation, the hardness of the fixed-horizon minimal disclosure problem is obtained by a reduction from the validity of a quantified Boolean formula. Let $\phi = \exists x_1 \forall y_1 \exists x_2 \dots \forall y_n \psi$ with $\psi = \bigwedge_{i=1\dots m}(z_{i_1} \vee z_{i_2} \vee z_{i_3})$ a closed QBF where we assume w.l.o.g. that in every clause the literals are distinct. We build the OMDP $\mathsf{M} = (S, \mathsf{Act}, p, \mathsf{O})$ (represented in Figure 7.13) where:

- $S = \{s_{init}, s_{end}, s_\sharp\} \cup \{s_{z_i} \mid z \in \{x,y\}, i = 1\dots n\} \cup \{s_{\neg z_i} \mid z \in \{x,y\}, i = 1\dots n\} \cup \{s_t^i \mid i \in \{1,\dots,m\}\} \cup \{s^{z_{i_j}} \mid i \in \{1,\dots,m\}, j \in \{1,2,3\}\} \cup \{s_{end}^{z_i} \mid z \in \{x,y\}, i = 1\dots n\} \cup_{i\in\{1,\dots n\}} (S_{x_i} \cup S_{y_i} \cup S_{\neg x_i} \cup S_{\neg y_i})$ where for z_i one of the Boolean variable, $S_{z_i} = \{s_a^{z_i} \mid a \in \{\sharp, z_i, \neg z_i \mid z \in \{x,y\}, i \in \{1,\dots,n\}\}\} \cup \{s_\flat^{z_i}, s_f^{z_i}\}$. Similar for the box $S_{\neg z_i}$ of the negation of a variable.
- $A(s_{init}) = \{x_1, \neg x_1\}$, and for all $i < n$, $A(s_{y_i}) = A(s_{\neg y_i}) = \{x_{i+1}, \neg x_{i+1}\}$. For $i \in \{1,\dots,m\}, A(s_t^i) = \{1,2,3\}$ and for every other state (even in boxes), $A(s) = \{\mathsf{next}\}$.

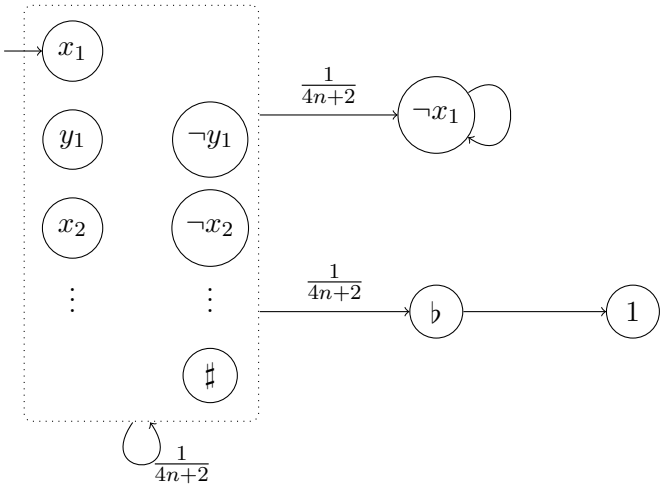


Figure 7.14: Representation of the box S_{x_1} (with the conventions of Figure 7.8).

- $p(s_a \mid s_{init}, a) = p(s_a^{a,1} \mid s_{init}, a) = 1/2$. For all $i < n$, $p(s_a \mid s_{y_i}, a) = p(s_a^{a,1} \mid s_{y_i}, a) = p(s_a \mid s_{\neg y_i}, a) = p(s_a^{a,1} \mid s_{\neg y_i}, a) = 1/2$. For all $i \leq n$, $p(s_{y_i} \mid s_{x_i}, \mathsf{next}) = p(s_{y_i}^{y_i,1} \mid s_{x_i}, \mathsf{next}) = p(s_{\neg y_i} \mid s_{x_i}, \mathsf{next}) = p(s_{\neg y_i}^{\neg y_i,1} \mid s_{x_i}, \mathsf{next}) = p(s_{s_{y_i}|\neg x_i}, y_i) = p(s_{y_i}^{y_i,1} \mid s_{\neg x_i}, y_i) = p(s_{\neg y_i} \mid s_{\neg x_i}, \mathsf{next}) = p(s_{\neg y_i}^{\neg y_i,1} \mid s_{\neg x_i}, \mathsf{next}) = 1/4$. $p(s_t^1 \mid s_\sharp, \mathsf{next}) = 1$. For all $i = 1 \ldots m$, $j \in \{1, 2, 3\}$, $p(s^{z_{i_j}} \mid s_t^i, j) = 1$, and if $i < m$, $p(s_t^{i+1} \mid s^{z_{i_j}}, \mathsf{next}) = 1$. Finally $p(s_{end} \mid s^{z_{m_j}}, \mathsf{next}) = 1$, and for all $z \in \{x, y\}$, $i = 1 \ldots n$, $p(s_{end}^{z_i} \mid s_{end}, \mathsf{next}) = 1/(2n)$.

  We now describe p for the box S_{x_1} other boxes being similar. For all $a \in \{\sharp, x_1, z_i, \neg z_i \mid z \in \{x, y\}, i \in \{2, \ldots, n\}\}, b \in \{\sharp, \flat, z_i, \neg z_i \mid z \in \{x, y\}, i \in \{1, \ldots, n\}\}$ $p(s_b^{x_1} \mid s_a^{x_1}, \mathsf{next}) = 1/(4n+2)$, and $p(s_{\neg x_1}^{x_1} \mid s_{\neg x_1}^{x_1}, \mathsf{next}) = p(s_f^{x_1} \mid s_\flat^{x_1}, \mathsf{next}) = p(s_f^{x_1} \mid s_f^{x_1}, \mathsf{next}) = 1$.

- $\mathsf{O}(s_{end}) = \flat$, $\mathsf{O}(s_a^b) = a$ and $\mathsf{O}(s^a) = a$ when a is a Boolean variable, its negation or $\flat$. For $i = 1 \ldots n$, $\mathsf{O}(s_{end}^{x_i}) = \mathsf{O}(s_f^{x_i}) = \mathsf{O}(s_f^{\neg x_i}) = 2i - 1$ and $\mathsf{O}(s_{end}^{y_i}) = \mathsf{O}(s_f^{y_i}) = \mathsf{O}(s_f^{\neg y_i}) = 2i$, and for any other state s, $\mathsf{O}(s) = \sharp$.

The initial distribution μ_0 is $\mathbf{1}_{s_{init}}$ and the secret runs are the ones visiting s_{end} (Sec $= \{s_{end}\} \cup \{s_{end}^{z_i} \mid z \in \{x, y\}, i = 1 \ldots n\}$).

In a similar fashion as what was done in the hardness part of the proof of Proposition 7.7, we show that ϕ is true iff the disclosure of M is equal to 0 in $2(n + m + 2)$ steps. First observe that a run ρ reaching s_{end} can be extended for all $j \in \{1, \ldots, 2n\}$ in a run ρ_j such that $\mathsf{O}(\rho_j) = \mathsf{O}(\rho)j$. Moreover, ρ_1 discloses the secret iff x_1 and $\neg x_1$ both occur in $\mathsf{O}(\rho_1)$ (and similarly for the other ρ_js). Indeed a run reaching S_{x_1} or $S_{\neg x_1}$ cannot have triggered both observations x_1 and $\neg x_1$ and also end with observation 1.

Intuitively, during the first $2n$ steps, all Boolean variables are be assigned a value: either chosen by the strategy as it chooses whether x_i or $\neg x_i$ occurs in the observation

for each $1 \leq i \leq n$, or randomly as y_i and $\neg y_i$ both have half a chance of being triggered. During the last $2m+1$ steps, the strategy must choose a Boolean formula in every clause so that if a clause is not satisfied by the current assignment, then a Boolean variable is observed as both true and false during the run. The last step then triggers randomly the observation j for $j \in \{1, \dots 2n\}$.

Suppose that ϕ is equivalent to true. Then there exist functions $(f_i)_{i=1\dots n}$ such that for every set of assignments $(a_1, \dots, a_n)$ for the variables $y_1, \dots, y_n$ the Boolean formula $\psi[f_1(), a_1, f_2(a_1), \dots, f_n(a_1, \dots, a_{n-1}), a_n)]$ is true. We choose a strategy σ such that for every possible set of assignments $(a_1, \dots, a_n)$ for the variables $y_1, \dots, y_n$, and for all i, $0 \leq i \leq n-1$, $\sigma(\sharp f_1()a_1 f_2(a_1) \dots a_i) = f_{i+1}(a_1, \dots, a_i)$. Moreover for $k \in \{1, \dots, m\}$ and $i_1, \dots, i_k \in \{1, 2, 3\}$, there exists $z_{k+1_{i_{k+1}}} \in \{f_1(), a_1, f_2(a_1), \dots, a_n\}$ such that $\sigma(\sharp f_1()a_1 f_2(a_1) \dots a_n \sharp\sharp z_{1_{i_1}} z_{2_{i_2}} \dots z_{k_{i_k}}) = z_{k+1_{i_{k+1}}}$. The choice of the strategy is arbitrary in the other cases. Since $\psi[f_1(), a_1, f_2(a_1), \dots, f_n(a_1, \dots, a_{n-1}), a_n)]$ is true, every clause is satisfied by this choice of assignments, hence it is possible to define such a strategy.

With this strategy, the fixed-horizon disclosure in $2(n + m + 2)$ steps is 0. In other words, none of the runs reaching the secret discloses it. Indeed let ρ be a secret run of length $2(n + m + 2)$, then there exist $a_1, \dots, a_n \in \{y_1, \neg y_1, \dots y_n, \neg y_n\}$ and $j \in \{1, \dots, 2n\}$ such that $\mathsf{O}(\rho) = \sharp f_1()a_1 f_2(a_1) \dots a_n \sharp\sharp z_{1_{i_1}} z_{2_{i_2}} \dots z_{m_{i_m}} \flat j$. By choice of σ, if, for $z \in \{x, y\}$ and $i \in \{1, \dots n\}$, z_i appears in the observation of ρ, $\neg z_i$ does not, and vice versa. Therefore as $\flat j$ can be read either from s_{end} or in a box state outside of the secret reachable only by runs that do not observe a Boolean variable and its negation, ρ does not disclose the secret.

Conversely, suppose that ϕ is not equivalent to true and let σ be an arbitrary strategy. We first build a deterministic strategy σ' with smaller or equal disclosure. The first choice concerns $\{x_1, \neg x_1\}$ and the next observation in a run corresponds to that choice. Consider σ_1 (resp. σ'_1) the strategy that selects x_1 (resp. $\neg x_1$) and then plays like σ. Due to the fact that observations are distinct, the disclosure value w.r.t. σ is a convex combination of the ones of σ_1 and σ'_1. So one substitutes σ by the one with smaller or equal disclosure. A similar pattern applies for every choice until reaching the horizon. Thus by iterating this transformation we obtain a deterministic strategy. So we assume now that σ is deterministic. Since there is a finite number of such strategies for fixed horizon, it only remains to prove that the disclosure value under σ is positive. We build partial functions $f_i : \Sigma^{2i} \mapsto \mathsf{Act}$ consistent with σ: for every observation $\sharp w \in \sharp\Sigma^{2i}$ of some run ρ, if σ chooses action $a \in A(\mathsf{last}(\rho))$ for ρ, then we set $f_i(w) = a$. As ϕ is not equivalent to true, there exists an assignment $(a_1, \dots, a_n)$ for the variables $y_1, \dots, y_n$ such that the Boolean formula $\psi[f_1(), a_1, f_2(a_1), \dots, f_n(a_1, \dots, a_{n-1}, a_n)]$ is false.

We now build a run disclosing the secret. By construction, there exists ρ such that $\mathsf{O}(\rho) = \sharp f_1()a_1 f_2(a_1) \dots f_n(a_1 \dots a_{n-1})a_n \sharp$, leading to $\mathsf{last}(\rho) = s_\sharp$ with $\mathbb{P}_\sigma(\rho) > 0$. Let $i \in \{1, \dots, m\}$ such that the negations of z_{i_1}, z_{i_2} and z_{i_3} were chosen as assignment hence $z_{i_1} \vee z_{i_2} \vee z_{i_3}$ is not true under the assignment $[f_1(), a_1, f_2(a_1), \dots, f_n(a_1, \dots, a_{n-1}), a_n)]$. Let ρ' be a run of length $2(n + m) + 3$ extending ρ and ending in s_{end}. Then ρ' does

not disclose the secret because there exists $j \in \{1,2,3\}$ such that z_{i_j} appears in the previous $2m$ observations while its negation (written $\neg z_{i_j}$ here) appears in the first $2n+1$ observations. Let ρ'' of length $2(n+m+2)$ extending ρ' by ending in $s_{end}^{z_{i_j}}$. There is no other run with the same observation and ρ'' is a secret run, thus ρ'' discloses the secret. Therefore the disclosure of M is positive.

Observe that this reduction also works for finite horizon since no further disclosure may occur after the first occurrence of a state in $\{s_{end}^{x_1}, \ldots, s_{end}^{x_n}, s_{end}^{y_1}, \ldots, s_{end}^{y_n}\}$. □

Contrary to the case of maximisation, the above proof implies PSPACE-completeness for the limit-sure and almost-sure problem for disclosure minimisation.

Remark 7.2. *As for maximisation, the proof of hardness can be adapted for ε-disclosure and the algorithm for membership cannot be directly applied. The minimal fixed horizon ε-disclosure could however be computed by minimising an exponential system of equations, resulting in an exponential time algorithm.*

5 Conclusion

To our knowledge, the opacity of probabilistic systems had only been studied in order to maximise the disclosure of the system. Moreover, these studies always restricted the framework so that the strategy that is chosen does not modify if an observed sequence is disclosing or not, leaving the general case open. In the context of the previous studies, only maximisation was considered, which is understandable as maximisation and minimisation of disclosure are similar: they both consists in the optimisation of a fixed event. We, however, focused on the general case, both for maximisation, and for minimisation. In our framework, maximisation and minimisation present a strong asymmetry. Indeed, when considering finite horizon, most maximisation problems are undecidable although deterministic strategies are optimal. In contrast, minimisations problems are decidable, but good strategies often require randomisation. Note that a complexity gap (PSPACE-hard versus in EXPTIME) remains to fill for the finite-horizon minimisation problem. For fixed horizon, there is still an asymmetry between maximisation and minimisation that clearly appears in some parts of the proofs. But it is not as strong as in finite horizon and algorithms with good complexities can be obtained for both.

Although we used a variant of Markov decision processes enriched with observation) to represent our models, opacity is not an usual MDP problem. Indeed, opacity is an hyper property as the disclosure depends on a set of paths linked by their observation. This gives a partial observation flavour to opacity. Opacity as seen here is therefore a problem in between OMDP and POMDP. For this kind of problems, as seen in this chapter, it is important to determine whether the problem can be translated to an MDP or a POMDP problem in order to use the results known on these models. Here, maximisation of the disclosure was closer to POMDP problems while minimisation was closer to OMDP problems.

A promising research direction to consider is the approximate notion of opacity. It is the most natural notion of opacity. Indeed, if an attacker knows there is a 99% chance to be in the secret, the secret could be considered to be disclosed. As we have shown in Section 1, the most naive definition of approximate opacity is undecidable even without control. For diagnosability, we showed in Chapter 4 that AFF-diagnosability, an elaborate notion of approximate diagnosability was decidable for passive systems. A natural question is thus to determine if we can define a similar notion of opacity. A notion that would measure the set of infinite secret runs which disclose with arbitrarily high probability the secret for example. As it is close to AFF-diagnosability, we conjecture that it should be decidable for observable Markov chains. However, in active system, the finite-horizon maximisation/minimisation disclosure problem are likely to be undecidable.

Chapter 8

Conclusion

Contributions

This thesis constitutes part of the work towards a theoretical analysis of partial observation problems in a stochastic framework. More specifically it focused on the problem of diagnosis. Diagnosis had already been studied for stochastic systems [TT05, CK13, BFH$^+$14], however the definitions used varied and many central issues had been left open. The first step to set solid foundations for the analysis of diagnosis in probabilistic systems was thus to define precise and realistic notions of diagnosis which would encompass the ones already established. This was done in Chapter 2. Before focusing on any specific framework, we performed in Chapter 3 a semantical analysis of the different notions of diagnosability. While some intuitions on the relations between the notions could be obtained directly from the definitions, the analysis allowed, among other things, to establish formally these links, with a few surprises due mostly to the distinction between finite systems, finitely-branching systems and infinitely-branching systems.

We then turned to multiple specific frameworks and developed methods to decide diagnosability with optimal complexity. First, we focused on passive systems. Moreover, in Chapter 4, we restricted ourselves to finite systems. This important restriction pushed us to make once again some semantical analysis in order to obtain refined results exploiting the finite number of states. This gave us precise characterisations of the decidable notions of diagnosis, allowing us to establish the exact complexities of the problems. We also showed how to automatically build diagnosers associated with each notion of diagnosability. In Chapter 5, we extended our analysis to infinite-state systems. This immediately raised one important issue that we did not have for finite systems: how to represent such systems. In consequence, we studied several possible representations, one of which yielded multiple decidability results. These decidability results were obtained in large part thanks to the analysis made in Chapter 3. This emphasizes the importance of a good understanding of a notion and how to characterise it as a preamble to study the problem.

We then considered active systems. As for stochastic infinite systems, many different

frameworks may be studied. But contrary to stochastic infinite systems, diagnosis had already been studied for stochastic active systems [BFH+14], providing a framework our developments. The latter revealed an issue with the control of a system: ensuring diagnosability could be at the expense of the correct performance of the system. We studied in Chapter 6 how to limit the degradation of the system while preserving diagnosability. More precisely, we defined notions of the degradation of a system and showed the decidability and precise complexity for some of them, and established the undecidability of the others.

In the last chapter, Chapter 7, we switched our focus to opacity, another partial observation problem. In active systems, we showed how, when possible, one could develop strategies maximising or minimising the opacity of a system. The main element that made this analysis successful is the understanding of the forms that the optimal strategies would take.

Our contributions, while providing a good foundation for the diagnosis of probabilistic systems and many interesting results, are far from giving the whole picture. In the next section we provide a list of remaining open questions and research directions extending the thesis.

Perspectives

The current thesis opens quite a few perspectives, some of which were already given in each chapter conclusion and are partially repeated here. We classify these ideas depending on whether they are short-term, mid-term or long-term objectives. This decomposition represents how direct the link between the current work and the perspective is. We start with the short-term perspectives, *i.e.* the problems immediately raised by the works presented here.

- The most immediate perspectives are the ones given by the gaps within our results: notions for which we could not establish the decidability status, complexities that are not tight, etc. For example, the algorithms given to decide the exact notions of diagnosability in probabilistic visibly pushdown automata are in EXPSPACE while the proven lower bound is only EXPTIME. Another open question is the exact complexity of computing the minimal disclosure of the opacity (PSPACE lower bound versus EXPTIME upper bound). The main open question however is the decidability status of the FA-diagnosability in pVPA. We showed that this notion was harder than the other notion of exact diagnosability by studying its membership in the Borel hierarchy, but could not give an algorithm nor an undecidability proof. We conjecture that this notion is decidable and, in fact, with an algorithm of the same complexity than the other. Indeed, the proofs of the the non-expressivity results that limited the study of FA-diagnosability uses systems that cannot be expressed by pVPA. There could exists a pathL formula that would characterise FA-diagnosability when restricted to pVPA. An ongoing work seems to confirm this to be true.

- Another immediate perspective is raised by the introduction of the pathL logic. It was used in Chapter 5 in order to decide some notions of exact diagnosability. This logic may be useful for example to test properties such as the uniformity of the speed of diagnosis, the boundedness of the mean detection time of a fault (or mean time before an information about the correctness of a run), etc. Moreover, if the pathL logic cannot be used directly, one could define an enriched version of this notion with greater expressive power. This enriched version must be carefully designed so that the generated formulae can be checked.

- The last immediate perspective relates to the active framework. Whether it be for diagnosis or opacity, we only focused on exact notions. It would be natural to tackle the, usually harder, approximate notions. This is in fact an ongoing work. The current results seem to point toward decidability for AFF-diagnosability in active systems while similar approximate notions are undecidable for opacity, both for maximisation and minimisation.

We now turn to mid-term perspectives. They correspond to problems that are strongly connected to this thesis, while not being immediate.

- In our active framework, the observations are clearly given by the model. This represents in reality sensors within the system. Using a sensor has a cost. Therefore, instead of having fixed sensors, one could have a list of potential sensors associated with costs. The goal would then be to obtain diagnosability while minimising the cost. A cost could also be given to having the sensor turned on, forcing the optimal strategy to decide when it needs to have the sensor operating. Some works were already done on this subject, see [CT08] and [TT07].

- Faults, as defined in this document, are a boolean property: a system is either faulty or correct. Moreover, they are permanent. Once a fault occurred, we did not consider as important to decide if more faults would be created later for example. No matter the number of faults, the run is deemed faulty. One could envision a different idea for the fault. A fault could represent a partial degradation of the system, the failure of one of its non-vital component or something that can be repaired (see [FHLM18] for a study of repairable faults in a non-stochastic framework). Seen this way, many new questions arise. In passive systems, this means defining measures of correctness for a system and testing properties, for example on the delay of the fault counter (one can test if this delay is bounded, if it is unbounded but in $o(n)$ where n is the length of the run, in $O(n)\dots$). In active systems, we would wish to find controllers that optimise the measures of interests or that ensure good delays of detection for the fault counter. This would obviously only be an interesting study for systems where many faults will be triggered. However, this is not an unrealistic assumption as every system is slowly degraded due to time elapsing.

- Our diagnosability algorithms currently only gives a Boolean answer. In order for system designers to modify its system so that it becomes diagnosable for example,

they need to know what is the cause of the missed fault. As a consequence, it would be interesting to design algorithm able to give a counter-example to the diagnosability of the system whenever it exists. This notion of counter-example needs to be made precise. In an LTS, a counter-example can be given by an ambiguous cycle of the system. As with probabilistic systems, this cycle may have a zero probability, the counter-example must thus be able to describe a set of runs with positive probability. For example, by giving a finite faulty run such that any extension of it is ambiguous. However, this kind of counter-example does not necessarily exist, in pushdown systems for example. The generation of counter-examples in stochastic systems has been studied, see [ÁBD+14], but for different objectives than diagnosability.

- The main formalism we chose, probabilistic labelled transition systems, has its limitations. One could be interested in studying higher-level models such as stochastic Petri nets, stochastic process algebra (with PEPA for example [Hil96, Chapter 3]), etc. can be more appropriate to represent some real life systems. Indeed, high-level formalisms usually are (often exponentially) more concise than low level formalisms, hence a greater comfort for designers. Moreover, high-level formalisms usually have a structure, which allows to conceive more efficient algorithms as the generated systems (pLTS, MDP, etc.) benefit from additional properties. For example, when a system is a synchronised product of several components, the transition matrix or the infinitesimal generator can be computed using tensor products of the matrix representing the different components (see *e.g.* [HM95, HM96]).

- During this thesis, we designed many algorithms. In parallel to the previous item, it would be useful to implement these algorithms. This tool could then be used to solve the diagnosability problems for pLTS or for some higher-level models whose semantics is an appropriate pLTS. This implementation should be integrated within an existing tool to benefit from the possibilities offered while enriching it. A good candidate would be COSMOS [BBD+15], a statistical model checker for the hybrid automata stochastic logic.

We now end with the long-term perspectives, more remote to our current work.

- We established many results within this thesis. Some of which used methods that are quite usual, some required to use new ideas such as the pathL logic. These new ideas may be useful in order to tackle other issues related to partial observation such as identification problems, stochastic games... It could therefore be interesting to see which kind of problems, in these other frameworks, would benefit from our approach.

- Another direction comes from another interpretation of faults. Let us proceed through an example. When someone is sick, multiple symptoms appear or do not appear in the body. These symptoms each correspond to a failure of the human body, so each are a fault that must be detected. However, in order to

cure the patient one needs to link the patterns to a common cause: an illness. In other words, one wants to deduce a meta-information from the behaviour of the system. Finding the origin of the fault has been studied under the term causality, but, as explained in [GSS17], this approach focused on static systems. In dynamic systems, this meta-information may be seen as a pattern that must be detected [JMPC06]. This objective can be seen as an extension of diagnosis, but it goes into the long-term category as it corresponds to what seems to be a far more general question. It can also be linked to questions of identification of complex behaviour, possibly similar to what was studied in [Pie14].

- Diagnosis is a research domain with clear applications. As a consequence, it would gain a lot from being studied in cooperation with the industry. This would allow researchers to better understand the industry's need. For example, there could exists definitions of diagnosability that we did not focus on, yet have relevance from an industrial point of view. Moreover, while our methods are efficient in theory, they may raise practical issues that we did not consider. There are already some cooperations such as [HF12] where the authors investigate the problem of building a model appropriate for diagnosis out of a real system, in their case, a network.

- The last perspective discussed here is of a different nature: it is not a research direction. However, it could still have a great impact on the domain of research. There exist many contributions which either establish a known result or have an erroneous proof for a theorem that was already proven false (trying to give a PTIME algorithm for a PSPACE-hard problem for example). This clearly shows the difficulty for researchers to know the current state of the research, even for a specific domain such as diagnosis. To tackle this issue, one possibility may be to try to build a cooperative website that would gather all the results on diagnosis, something in the spirit of the POMDP webpage [POM]. This way, one could efficiently find the current state of the art on the domain and this would save a lot of useful time to many researchers. This obviously has some issues. To be useful, the existence of such a website has to be spread and people must keep it up to date. It would also raise many questions of organisation due to the width of the domain, even when restricted to diagnosis (the many existing frameworks, diagnosability notions, methods of approach,...). The various surveys on diagnosis issues [ZL13, Bas14] would help dealing with this point. But most of all, building such a website is extremely time-consuming.

Bibliography

[ABCP13] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of CCS'13*, pages 901–914. ACM, 2013.

[ÁBD+14] E. Ábrahám, B. Becker, C. Dehnert, N. Jansen, J.-P. Katoen, and R. Wimmer. Counterexample generation for discrete-time Markov models: an introductory survey. In *Proceedings of SFM'14*, volume 8483 of *LNCS*, pages 65–121. Springer, 2014.

[ADD99] R. B. Ash and C. A. Doleans-Dade. *Probability and measure theory*. Academic Press, 1999.

[Agu10] M. K. Aguilera. *Stumbling over consensus research: misunderstandings and issues, in Replication: theory and practice*, volume 5959 of *LNCS*. Springer, 2010.

[AM04] R. Alur and P. Madhusudan. Visibly pushdown languages. In *Proceedings of STOC'04*, pages 202–211. ACM, 2004.

[Bar14] B. Barbot. *Acceleration for statistical model checking. (Accélérations pour le model checking statistique)*. PhD thesis, École normale supérieure de Cachan, France, 2014.

[Bas14] F. Basile. Overview of fault diagnosis methods based on Petri net models. In *Proceedings of ECC'14*, pages 2636–2642. IEEE, 2014.

[BBB+14] N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Größer, and M. Jurdzinski. Stochastic timed automata. *Logical Methods in Computer Science*, 10(4), 2014.

[BBD+15] P. Ballarini, B. Barbot, M. Duflot, S. Haddad, and N. Pekergin. HASL: A new approach for performance evaluation and model checking from concepts to experimentation. *Performance Evaluation*, 90:53–77, 2015.

[BBS06] C. Baier, N. Bertrand, and P. Schnoebelen. A note on the attractor-property of infinite-state markov chains. *Information Processing Letters*, 97(2):58 – 63, 2006.

[BCS15] B. Bérard, K. Chatterjee, and N. Sznajder. Probabilistic opacity for Markov decision processes. *Information Processing Letters*, 115(1):52–59, 2015.

[BD08] D. Berwanger and L. Doyen. On the power of imperfect information. In *Proceedings of FSTTCS'08*, volume 2 of *LIPIcs*, pages 73–82. Leibniz-Zentrum für Informatik, 2008.

[BEKK13] T. Brázdil, J. Esparza, S. Kiefer, and A. Kučera. Analyzing probabilistic pushdown automata. *Formal Methods in System Design*, 43(2):124–163, 2013.

[BFG17] H. Bazille, E. Fabre, and B. Genest. Diagnosability degree of stochastic discrete event systems. In *Proceedings of CDC'17*, pages 5726–5731. IEEE, 2017.

[BFH+14] N. Bertrand, E. Fabre, S. Haar, S. Haddad, and L. Hélouët. Active diagnosis for probabilistic systems. In *Proceedings of FoSSaCS'14*, volume 8412 of *LNCS*, pages 29–42. Springer, 2014.

[BGG09] N. Bertrand, B. Genest, and H. Gimbert. Qualitative determinacy and decidability of stochastic games with signals. In *Proceedings of LICS'09*, pages 319–328. IEEE, 2009.

[BGI+01] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *Proceedings of CRYPTO'01*, pages 1–18. ACM, 2001.

[BHL14] N. Bertrand, S. Haddad, and E. Lefaucheux. Foundation of diagnosis and predictability in probabilistic systems. In *Proceedings of FSTTCS'14*, volume 29 of *LIPIcs*, pages 417–429. Leibniz-Zentrum für Informatik, 2014.

[BHL16a] N. Bertrand, S. Haddad, and E. Lefaucheux. Accurate approximate diagnosability of stochastic systems. In *Proceedings of LATA'16*, volume 9618 of *LNCS*, pages 549–561. Springer, 2016.

[BHL16b] N. Bertrand, S. Haddad, and E. Lefaucheux. Diagnosis in infinite-state probabilistic systems. In *Proceedings of CONCUR'16*, volume 59 of *LIPIcs*, pages 37:1–37:14. Leibniz-Zentrum für Informatik, 2016.

[BHL17a] B. Bérard, S. Haddad, and E. Lefaucheux. Probabilistic disclosure: Maximisation vs. minimisation. In *Proceedings of FSTTCS'17*, volume 93 of *LIPIcs*, pages 13:1–13:14. Leibniz-Zentrum für Informatik, 2017.

[BHL17b] N. Bertrand, S. Haddad, and E. Lefaucheux. Diagnostic et contrôle de la dégradation des systèmes probabilistes. In *Proceedings of MSR'17*. HAL, 2017.

[BHSS18] B. Bérard, S. Haar, S. Schmitz, and S. Schwoon. The complexity of diagnosability and opacity verification for Petri nets. *Fundamenta Informatica*, 2018. To appear.

[BK08] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT Press, 2008.

[BKM12] J. W. Bryans, M. Koutny, and C. Mu. Towards quantitative analysis of opacity. In *Proceedings of TGC'12*, volume 8191 of *LNCS*, pages 145–163. Springer, 2012.

[BKMS16] B. Bérard, O. Kouchnarenko, J. Mullins, and M. Sassolas. Preserving opacity on interval Markov chains under simulation. In *Proceedings of WODES'16*, pages 319–324. IEEE, 2016.

[BKMS18] B. Bérard, O. Kouchnarenko, J. Mullins, and M. Sassolas. Opacity for linear constraint Markov chains. *Discrete Evente Dynamic Systems*, 28(1):83–108, 2018.

[BMS15] B. Bérard, J. Mullins, and M. Sassolas. Quantifying opacity. *Mathematical Structures in Computer Science*, 25(2):361–403, 2015.

[BS84] B. G. Buchanan and E. H. Shortliffe. *Rule based expert systems: the MYCIN experiments of the stanford heuristic programming project*. Addison-Wesley, 1984.

[CDGH10] K. Chatterjee, L. Doyen, H. Gimbert, and T. A. Henzinger. Randomness for free. In *Proceedings of MFCS'10*, volume 6281 of *LNCS*, pages 246–257. Springer, 2010.

[CDH10] K. Chatterjee, L. Doyen, and T. A. Henzinger. Qualitative analysis of partially-observable Markov decision processes. In *Proceedings of MFCS'10*, volume 6281 of *LNCS*, pages 258–269. Springer, 2010.

[CGLS12] M. P. Cabasino, A. Giua, S. Lafortune, and C. Seatzu. A new approach for diagnosability analysis of petri nets using verifier nets. *Transactions on Automatic Control*, 57(12):3104–3117, 2012.

[CGS09] M. P. Cabasino, A. Giua, and C. Seatzu. Diagnosability of bounded Petri nets. In *Proceedings of CDC'09*, pages 1254–1260. IEEE, 2009.

[CGS14] M. P. Cabasino, A. Giua, and C. Seatzu. Diagnosability of discrete-event systems using labeled Petri nets. *Transactions Automation Science and Engineering*, 11(1):144–153, 2014.

[CK13] J. Chen and R. Kumar. Polynomial test for stochastic diagnosability of discrete-event systems. *Transactions on Automation Science and Engineering*, 10(4):969–979, 2013.

[CK14] T. Chen and S. Kiefer. On the total variation distance of labelled Markov chains. In *Proceedings of CSL-LICS'14*, pages 33:1–33:10. ACM, 2014.

[CK15] J. Chen and R. Kumar. Stochastic failure prognosability of discrete event systems. *Transactions on Automatic Control*, 60(6):1570–1581, 2015.

[CP09] E. Chanthery and Y. Pencolé. Monitoring and active diagnosis for discrete-event systems. *IFAC Proceedings Volumes*, 42(8):1545 – 1550, 2009.

[CSH08] K. Chatterjee, K. Sen, and T. A. Henzinger. Model-checking omega-regular properties of interval Markov chains. In *Proceedings of FOSSACS'08*, volume 4962 of *LNCS*, pages 302–317. Springer, 2008.

[CT08] F. Cassez and S. Tripakis. Fault diagnosis with static and dynamic observers. *Fundamenta Informaticae*, 88:497–540, 2008.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[DHR08] L. Doyen, T. A. Henzinger, and J-F. Raskin. Equivalence of labeled markov chains. *International Journal of Foundations of Computer Science*, 19(3):549–563, 2008.

[Dia09] M. Diaz. *Petri nets: fundamental models, verification and applications.* Wiley, 2009.

[DLT00] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems*, 10(1-2):33–86, 2000.

[DMK+99] K. Doi, H. MacMahon, S. Katsuragawa, R. M. Nishikawa, and Y. Jiang. Computer-aided diagnosis in radiology: potential and pitfalls. *European Journal of Radiology*, 31(2):97 – 109, 1999.

[Eme90] E. A. Emerson. Handbook of theoretical computer science (vol. B). In *Temporal and Modal Logic*, pages 995–1072. MIT Press, 1990.

[EMT16] R. Ehlers, S. Moarref, and U. Topcu. Risk-averse control of Markov decision processes with ω-regular objectives. In *Proceedings of CDC'16*, pages 426–433. IEEE, 2016.

[EN94] J. Esparza and M. Nielsen. Decidability issues for Petri nets - a survey. *Elektronische Informationsverarbeitung und Kybernetik*, 30(3):143–160, 1994.

[EY09] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. *Journal of the ACM*, 56(1), 2009.

[EY12] K. Etessami and M. Yannakakis. Model checking of recursive probabilistic systems. *Transactions on Computational Logic*, 13(2):12, 2012.

[FHLM18] E. Fabre, L. Hélouët, E. Lefaucheux, and H. Marchand. Diagnosability of repairable faults. *Discrete Event Dynamic Systems*, 28(2):183–213, 2018.

[FS01] A. Finkel and P. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1):63 – 92, 2001.

[GL09] S. Genc and S. Lafortune. Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, 45(2):301–311, 2009.

[GO10] H. Gimbert and Y. Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *Proceedings of ICALP'10*, volume 6199 of *LNCS*, pages 527–538. Springer, 2010.

[GS14] J. Goubault-Larrecq and R. Segala. Random measurable selections. In *Horizons of the Mind. A Tribute to Prakash Panangaden*, volume 8464 of *LNCS*, pages 343–362. Springer, 2014.

[GSS17] G. Gößler, O. Sokolsky, and J.-B. Stefani. Counterfactual causality from first principles? In *Proceedings of CREST@ETAPS'17*, volume 259 of *EPTCS*, pages 47–53, 2017.

[GTWJ03] J. Z. Gao, J. Tsao, Y. Wu, and T. H.-S. Jacob. *Testing and quality assurance for component-based software*. Artech House, Inc., 2003.

[HC94] L. Holloway and S. Chand. Time templates for discrete event fault monitoring in manufacturing systems. In *Proceedings of ACC'94*, pages 701–706. IEEE, 1994.

[HF12] C. Hounkonnou and E. Fabre. Empowering self-diagnosis with self-modeling. In *Proceedings of CNSM'12*, pages 364–370. IEEE, 2012.

[HHMS13] S. Haar, S. Haddad, T. Melliti, and S. Schwoon. Optimal constructions for active diagnosis. In *Proceedings of FSTTCS'13*, volume 24 of *LIPIcs*, pages 527–539. Leibniz-Zentrum für Informatik, 2013.

[HHMS17] S. Haar, S. Haddad, T. Melliti, and S. Schwoon. Optimal constructions for active diagnosis. *Journal of Computer and System Sciences*, 83(1):101–120, 2017.

[Hil96] J. Hillston. *A Compositional approach to performance modelling*. Cambridge University Press, 1996.

[HM95] Serge Haddad and Patrice Moreaux. Evaluation of high-level Petri nets by means of aggregation and decomposition. In *Proceedings of PNPM'95*, pages 11–20. IEEE, 1995.

[HM96] Serge Haddad and Patrice Moreaux. Asynchronous composition of high level Petri nets: A quantitative approach. In *Proceedings of APN'96*, volume 1091 of *LNCS*, pages 192–211. Springer, 1996.

[HM09] S. Haddad and P. Moreaux. *Stochastic Petri nets, in Petri nets: fundamental models, verification and applications*. Wiley, 2009.

[HS10] J. Hromkovič and G. Schnitger. On probabilistic pushdown automata. *Information and Computation*, 208(8):982 – 995, 2010.

[JHCK01] S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial algorithm for testing diagnosability of discrete-event systems. *Transactions on Automatic Control*, 46(8):1318–1321, 2001.

[JL91] B. Jonsson and K. G. Larsen. Specification and refinement of probabilistic processes. In *Proceedings of LICS'91*, pages 266–277. IEEE, 1991.

[JMPC06] T. Jeron, H. Marchand, S. Pinchinat, and M. O. Cordier. Supervision patterns in discrete event systems diagnosis. In *Proceedings of WODES'06*, pages 262–268. IEEE, 2006.

[KEM06] A. Kučera, J. Esparza, and R. Mayr. Model checking probabilistic pushdown automata. *Logical Methods in Computer Science*, 2(1):12–21, 2006.

[KLC98] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99 – 134, 1998.

[KS60] John G Kemeny and James Laurie Snell. *Finite Markov chains*, volume 356. van Nostrand Princeton, 1960.

[KS16] S. Kiefer and A. P. Sistla. Distinguishing hidden Markov chains. In *Proceedings of LICS'16*, pages 66–75. ACM, 2016.

[Kur64] S.-Y. Kuroda. Classes of languages and linear-bounded automata. *Information and Control*, 7(2):207 – 223, 1964.

[LGS18] E. Lefaucheux, A. Giua, and C. Seatzu. Basis coverability graph for partially observable Petri nets with application to diagnosability analysis. In *Proceedings of PETRI NETS'18*, volume 10877 of *LNCS*, pages 164–183. Springer, 2018.

[MHC03] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147(1-2):5–34, 2003.

[Mos80] Y. N. Moschovakis. Studies in logic and the foundations of mathematics. In *Descriptive Set Theory*, volume 100, pages 11 – 64. Elsevier, 1980.

[MP09] C. Morvan and S. Pinchinat. Diagnosability of pushdown systems. In *Proceedings of HVC'09*, volume 6405 of *LNCS*, pages 21–33. Springer, 2009.

[MS72] A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *SWAT'72*, pages 125–129. IEEE, 1972.

[Mur89] T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[ND08] F. Nouioua and P. Dague. A probabilistic analysis of diagnosability in discrete event systems. In *Proceedings of ECAI'08*, volume 178 of *FAIA*, pages 224–228. IOS Press, 2008.

[Paz71] A. Paz. *Introduction to probabilistic automata*. Academic Press, 1971.

[Pie14] A. Piel. *Online event flow processing for complex behaviour recognition*. PhD thesis, Paris 13 University, France, 2014.

[Pnu77] A. Pnueli. The temporal logic of programs. In *Proceedings of FOCS'77*, pages 46–57. IEEE, 1977.

[POM] POMDP webpage. `http://www.pomdp.org/`.

[Pos46] E. L. Post. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society*, 52:264–268, 1946.

[PT87] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[Put94] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. Wiley, 1994.

[Å65] K.J. Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174 – 205, 1965.

[Rac78] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223 – 231, 1978.

[RSX12] N. Rampersad, J. Shallit, and Z. Xu. The computational complexity of universality problems for prefixes, suffixes, factors, and subwords of regular languages. *fundamenta informaticae*, 116(1-4):223–236, 2012.

[Sav70] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177 – 192, 1970.

[SH14] A. Saboori and Ch. N. Hadjicostis. Current-state opacity formulations in probabilistic finite automata. *Transactions on Automatic Control*, 59(1):120–133, 2014.

[Sip06] M. Sipser. *Introduction to the theory of computation*. Thomson Course Technology, 2006.

[SLT98] M. Sampath, S. Lafortune, and D. Teneketzis. Active diagnosis of discrete-event systems. *Transactions on Automatic Control*, 43(7):908–929, 1998.

[SSL+95] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *Transactions on Automatic Control*, 40(9):1555–1575, 1995.

[SVA06] K. Sen, M. Viswanathan, and G. Agha. Model-checking Markov chains in the presence of uncertainties. In *Proceedings ofTACAS'06*, volume 3920 of *LNCS*, pages 394–410. Springer, 2006.

[SZF11] A. P. Sistla, M. Zefran, and Y. Feng. Monitorability of stochastic dynamical systems. In *Proceedings of CAV'11*, volume 6806 of *LNCS*, pages 720–736. Springer, 2011.

[TT05] D. Thorsley and D. Teneketzis. Diagnosability of stochastic discrete-event systems. *Transactions on Automatic Control*, 50(4):476–492, 2005.

[TT07] D. Thorsley and D. Teneketzis. Active acquisition of information for diagnosis and supervisory control of discrete-event systems. *Discrete Event Dynamic Systems*, 17:531–583, 2007.

[Var96] M. Y. Vardi. *An automata-theoretic approach to linear temporal logic, in Logics for concurrency: structure versus automata*, volume 1083 of *LNCS*. Springer, 1996.

[Var99] M. Y. Vardi. Probabilistic linear-time model checking: An overview of the automata-theoretic approach. In *Proceedings of ARTS'99*, volume 1601 of *LNCS*, pages 265–276. Springer, 1999.

[YL02] T-S. Yoo and S. Lafortune. Polynomial-time verification of diagnosability of partially observed discrete-event systems. *Transactions on Automatic Control*, 47(9):1491–1495, 2002.

[ZL13] J. Zaytoon and S. Lafortune. Overview of fault diagnosis methods for discrete event systems. *Annual Reviews in Control*, 37(2):308 – 320, 2013.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Le contrôle de l'information dans les systèmes probabilistes

Mot clés : Vérification de modèles, systèmes probabilistes, diagnostic, chaînes de Markov, observation partielle

Resumé : Le contrôle de l'information émise par un système a vu son utilité grandir avec la multiplication des systèmes communicants. Ce contrôle peut être réalisé par exemple pour révéler une information du système, ou au contraire pour en dissimuler une. Le diagnostic notamment cherche à déterminer, grâce à l'observation du système, si une faute a eu lieu au sein de celui-ci. Dans cet document, nous établissons des bases formelles à l'analyse des problèmes du diagnostic pour des modèles stochastiques. Nous étudions ensuite ces problèmes dans plusieurs cadres (fini/infini, passif/actif).

Title : Controlling Information in Probabilistic Systems

Keywords : Model checking, probabilistic systems, diagnosis, Markov chains, partial observation

Abstract : The control of the information given by a system has recently seen increasing importance due to the omnipresence of communicating systems, the need for privacy, etc. This control can be used in order to disclose an information of the system, or, oppositely, to hide one. Diagnosis for instance tries to determine from the observation produced by the system whether a fault occurred within it or not. In this PhD, we study the diagnosis of stochastic systems through a model-based approach. The goal is to establish the decidability and optimal complexity of the decision problems and to build the adequate diagnosers. We consider these problems in multiple frameworks (finite/infinite, passive/active).