La suppression des anomalies dans les protocoles : vers une méthode basée sur l'ajout incrémental de transitions

César Viho

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

La suppression des anomalies dans les protocoles : vers une méthode basée sur l'ajout incrémental de transitions

Gagnon Viho

N° 2526

Mars 1995

PROGRAMME 1

ISSN 0249-6399

La suppression des anomalies dans les protocoles : vers une méthode basée sur l'ajout incrémental de transitions

Gagnon Viho*

Programme 1 — Architectures parallèles, bases de données, réseaux et systèmes distribués
Projet PAMPA

Rapport de recherche n° 2526 — Mars 1995 — 20 pages

Résumé : Ce rapport traite du problème de la mise en œuvre de méthodes destinées à supprimer les anomalies dans les protocoles. A travers l'étude de quelques exemples de protocoles, nous montrons comment on peut utiliser le service et des abstractions du protocole pour apporter une solution à ce problème. Contrairement à nos travaux précédents orientés vers la suppression de transitions, les solutions préconisées ici sont basées sur l'ajout incrémental de transitions dans les entités communicantes.

Mots-clé : protocole, anomalie, réception non spécifiée, service, projection, abstraction, correction

(Abstract: pto)

*. viho@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

Suppressing Errors in Protocols : Towards a Method Based on Incremental Addition of Transitions

Abstract: This paper discusses the problem of the development of methods aimed at suppressing errors in protocols. Through the study of some protocols, we show how the service and abstractions of the protocol can be used to bring a solution to this problem. In opposition to our previous works based on the suppression of transition, solutions proposed here consist in adding incrementally transitions in the communicating entities.

Key-words: protocol, error, unspecified reception, service, projection, abstraction, correctness

Table des matières

1 Introduction

A l'heure actuelle, divers outils permettent de spécifier formellement et de détecter les anomalies dans les protocoles. Ceci constitue une aide importante aux concepteurs de protocoles. Cependant il serait souhaitable d'aller plus loin en développant des méthodes et outils qui proposent automatiquement les modifications à apporter au protocole erronné pour supprimer les anomalies, assurant ainsi la *correction* du protocole [15, 16].

Pour ce faire, il nous faut tenir compte du fait important qu'un protocole doit rendre le *service* pour lequel il a été conçu. Il serait donc judicieux de s'aider du service pour déterminer les corrections à effectuer sur le protocole tout en assurant sa conformité au service : *le service ne doit pas être affecté par les modifications effectuées lors de la correction.* C'est l'objectif que nous nous fixons dans ce travail. A travers l'étude de quelques exemples significatifs de protocoles, nous montrons que l'on peut utiliser la spécification du service du protocole pour supprimer les anomalies détectées.

Nous utilisons un des modèles les plus répandus que sont les automates d'états finis pour modéliser aussi bien les protocoles que les services correspondants. La détection des anomalies s'effectue sur le *graphe d'accessibilité* construit à l'aide de la méthode classique dite de perturbation. Dans le paragraphe 4, nous décrivons les abstractions du graphe d'accessibilité que nous utilisons pour apporter une solution au problème de la correction des protocoles. Nous étudions (aux paragraphes 6, 7, 8.1 et 8.2) quelques exemples significatifs montrant comment ces abstractions permettent la supprssion des anomalies dans les protocoles

2 Un modèle pour la spécification des protocoles et du service

Différents formalismes permettent la modélisation des protocoles : les automates d'états finis [6], les réseaux de Petri [3], la logique temporelle [4], les types abstraits de données [11] etc... Nous utiliserons un des modèles le plus répandu et le plus utilisé que sont les automates d'états finis.

2.1 Entité communicante et protocole

Dans ce modèle, chacune des entités communicantes est décrite par un automate d'états finis (AEF).

Un AEF A_i qui représente une entité (communicante) P_i, est composé d'un *ensemble d'états* E_i, d'un *alphabet de messages* M_i, d'un *ensemble de transitions* T_i et d'*état initial* e_i^0: $A_i = (E_i, M_i, T_i, e_i^0)$.

Soit une transition $t \in T_i$ alors t est de la forme (e,m,e') où e $\in E_i$ est l'état origine de la transition (noté **ori**(t)), e'$\in E_i$ l'état d'arrivée (noté **but**(t)), et m$\in M_i$ le message étiquetant la transition t (noté **éti**(t)).

Schématiquement, chaque AEF est décrite par un graphe orienté. Les nœuds représentent les états des entités. Les arcs représentent des transitions entre deux états : les transitions

étiquetées $+m$ sont des émissions de message et celles étiquetées $-m$ sont des réceptions. Les états 0 sont les états initiaux des automates. A titre d'exemple, la figure 1 décrit un protocole composé de deux entités.

Nous supposons que la communication entre les AEF s'effectue à travers des canaux FIFO bornés et fiables. Nous supposons qu'il existe un canal de communication d'une entité donnée vers n'importe quelle autre entité. Nous considérons alors qu'un protocole est totalement défini par la donnée des AEF décrivant les entités communicantes et la taille des canaux de communication. Dans ce rapport, nous étudions principalement des exemples de protocoles composés de deux entités. Aussi toutes les notions définies par la suite concernent des protocoles composés de deux entités.

2.2 Spécification de service

Le but de notre travail est de supprimer les anomalies d'un protocole en s'aidant du service et en s'assurant que le comportement du protocole reste conforme au service. Il est donc utile de disposer d'un modèle permettant de spécifier le service. Cependant, il est tout autant difficile de spécifier le service complet d'un protocole que le protocole lui-même. Du fait que l'on ne traite que les anomalies de blocage, nous nous contenterons de la spécification d'une abstraction du service vis-à-vis des actions de progression du protocole. Le protocole est alors vu comme une boîte noire et l'on observe que les événements correspondant à la progression du protocole.

Nous utiliserons le même modèle des AEF pour spécifier le service d'un protocole. La figure 8 montre le service associé au protocole 1.

3 Graphe d'accessibilité

La détection des anomalies s'effectue sur le graphe d'accessibilité obtenu par la méthode classique de perturbation [17, 8] qui construit tous les états globaux du protocole. Un état global est un quadruplet de la forme [x,y,α,β] où x et y sont les états courants des AEF des deux entités, et α et β les contenus courants des canaux de communication.

L'état global initial étant fixé, la méthode de perturbation génère les états globaux successeurs résultant du tirage des transitions dans les entités communicantes. Ce procédé est itéré ensuite sur les états globaux générés et permet d'obtenir le *graphe d'accessibilité*.

En vue de faire la distinction entre les différents graphes manipulés, les états globaux seront appelés *g-états* et les transitions du graphe d'accessibilité seront appelées des *g-transitions*.

Soit $\mathcal{P}$ un protocole de communication composé d'un ensemble d'entités communiquant à l'aide de canaux. Soit $\mathcal{G}$ le graphe d'accessibilité représentant les exécutions de $\mathcal{P}$, alors $\mathcal{G}$ est défini par un *ensemble de g-états* Π, un *ensemble de g-transitions* $\mathcal{T}$ et un *g-état initial* s^0.

Soit une g-transition $t \in \mathcal{T}$; t est de la forme (s,m,s') où s $\in \Pi$ est le g-état origine de la g-transition, s' $\in \Pi$ le g-état d'arrivée et m $\in \mathcal{M}$ le message étiquette de la g-transition t. $\mathcal{M}$ représente l'union des alphabets de chacun des entités du protocole $\mathcal{P}$: $\mathcal{M} = \bigcup_{i=1,N} M_i$.

On définit par ailleurs $\mathcal{T}^-$ *et* $\mathcal{T}^+$ les ensembles regroupant respectivement les g-transitions de réception (étiquetées -m) et les g-transitions d'émission (étiquetées +m).

Le problème qui se pose ici est la terminaison de la construction du graphe d'accessibilité. Ce problème bien connu est dû à l'explosion combinatoire du nombre d'états globaux générés. Des techniques ont été développées [17, 14, 2, 7, 9] pour réduire la taille du graphe d'accessibilité. Comme nous n'avons pas encore étudié l'influence des techniques de réduction sur les méthodes que nous décrivons dans ce rapport, nous supposerons par la suite que nous disposons du graphe d'accessibilité dans sa totalité.

3.1 Les anomalies

Ces anomalies de blocage sont détectées dans le graphe d'accessibilité et correspondent à des états-puits. Ainsi, une réception non spécifiée (RNS) est caractérisée par un g-état dans lequel la seule possibilité pour le protocole de progresser est de recevoir un message (en tête de canal) et il n'y a pas de transition de réception de ce message dans les entités. Par exemple, le g-état 10 du graphe d'accessibilité de la figure 3 représente une RNS.

Bien que notre objectif soit de corriger les anomalies dans les protocoles (en nous aidant du service), il serait utopique de traiter en même temps toutes les anomalies. Aussi dans ce rapport, nous nous intéressons principalement à la correction des RNS.

4 Abstractions du graphe d'accessibilité

Le graphe d'accessibilité décrit le comportement global du protocole. Afin de mettre en évidence les comportements du protocole qui nous permettront de le corriger, il nous faut opérer des transformations sur ce graphe. On obtient alors une *abstraction* du graphe d'accessibilité; chacune des abstractions apportant une ou plusieurs informations particulières nécessaires à la suppression de l'anomalie considérée.

Pour obtenir les abstractions nécessaires à la correction, nous avons à notre disposition différentes techniques de projection comme la bisimulation, l'équivalence observationnelle ou les projections sur les transitions de réception ou d'émission. En raison de leur facilité de mise en œuvre, le travail présenté dans ce rapport utilise les deux dernières projections et sont décrites ci-après.

Nous illustrerons l'utilisation de ces deux projections au paragraphe 5. Nous verrons alors leur intérêt dans la détermination des ajouts à effectuer pour assurer la correction du protocole.

4.1 Proj1 : Elimination des transitions de réception

La première projection consiste à remplacer toutes les transitions de réception (donc étiquetées -m) par des transitions vides notées λ : $\forall t \in \mathcal{T}$, si éti(t) $\in \mathcal{T}^-$ alors éti(t) = λ.

4.2 Proj2 : Elimination des transitions d'émission

De même, il est possible de projeter le graphe d'accessibilité de façon à éliminer les transitions d'émission. Ceci s'exprime comme suit : $\forall t \in \mathcal{T}$, si éti(t) $\in \mathcal{T}^+$ alors éti(t) = λ.

4.3 Déterminisation et minimisation des graphes projetés

Dans la démarche visant à corriger le protocole erronné, nous devons ensuite comparer le graphe d'accessibilité projeté et le graphe représentant le service. Ceci ne peut se faire que si ces graphes sont déterministes et minimaux [5]. Aussi, le graphe d'accessibilité obtenu après la projection est-il systématiquement déterminisé puis minimisé à l'aide d'algorithmes classiques [1].

Par la suite, $\mathcal{G}_m^-$ et $\mathcal{G}_m^+$ désigneront les graphes minimaux ainsi obtenus à partir du graphe d'accessibilité abstrait selon respectivement la projection Proj1 et Proj2.

Pour distinguer les états et les transitions d'un graphe minimal de ceux des autres graphes, nous les appellerons par la suite respectivement des *m-états* et des *m-transitions*.

5 Exemple 1 et première tentative de correction

Le protocole étudié ici met en œuvre deux sites ayant en charge le contrôle d'un réseau [12]. Les deux entités disposent chacune des données nécessaires et doivent s'assurer que celles-ci restent cohérentes. Pour cela, le protocole utilise le principe suivant : dès que l'une des entités observe un changement de l'état du réseau, elle met à jour ses informations propres et envoie une requête (*RQST*) à l'autre entité. Cette dernière informe son homologue (par envoi d'un acquittement *DONE*) de la prise en compte de la requête.

5.1 Spécification du protocole

Une spécification du protocole de l'exemple 1 est décrite sur la figure 1.

Les deux entités sont symétriques et peuvent chacune émettre les messages *RQST* et *DONE*. Or, la méthode présentée par la suite nécessite que les alphabets des messages de chacun des entités soient distincts. Nous modifions donc ces alphabets en transformant le message *RQST* en *a* pour l'entité 1 et en *c* pour l'entité 2. De même le message *DONE* devient *b* pour l'entité 1 et *d* pour l'entité 2. Nous verrons en étudiant le service associé à ce protocole que ces transformations ne modifient pas la sémantique du protocole. Les automates obtenus en modifiant ainsi les alphabets sont présentés à la figure 2.

5.2 Graphe d'accessibilité et abstraction

Le graphe d'accessibilité correspondant à ce protocole est présenté à la figure 3.

Nous remarquons dans le graphe d'accessibilité que le g-état 10 est une RNS puisque le canal reliant l'entité 1 à l'entité 2 n'est pas vide : l'entité 2 n'a pas pu consommer le message *a* présent en tête de son canal.

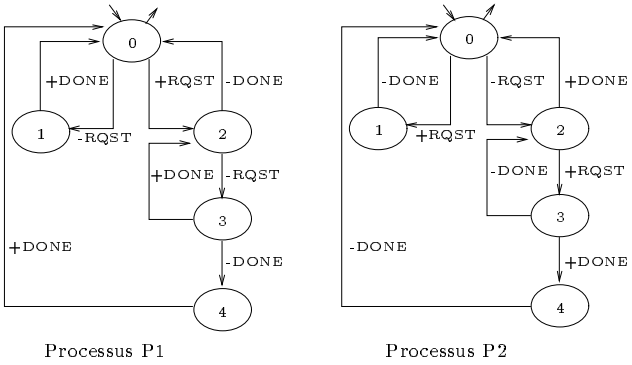

FIG. 1 - *Automates associés au protocole de l'exemple 1.*

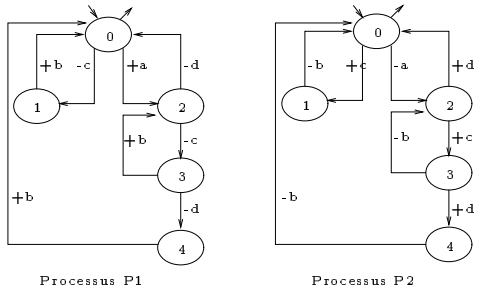

FIG. 2 - *Automates associés au protocole de l'exemple 1 en distingant les alphabets.*

Afin d'extraire du graphe d'accessibilité les informations qui vont nous permettre de corriger cette RNS, le graphe d'accessibilité est successivement projeté, déterminisé puis minimisé. Ces étapes sont détaillées dans [13], mais ici nous ne présenterons que le graphe obtenu après les trois étapes.

Le graphe d'accessibilité est tout d'abord projeté selon la projection Proj1 décrite dans le paragraphe 4.1. Il est ensuite déterminisé et minimisé. La figure 4 montre le graphe final déterministe et minimal.

Du fait du non-déterminisme engendré par les opérations de projection (introduction de λ-transitions), un même g-état peut se retrouver après (déterminisation et minimisation), dans plusieurs m-états différents. Le détail des regroupements est donné à la figure 5) et indique la correspondance entre les g-états du graphe d'accessibilité de la figure 3 et les m-états du graphe $\mathcal{G}_m^-$. Par la suite, $\mathcal{G}_m^-(E)$ désignera le m-état e correspondant à l'ensemble E des g-états regroupés par la minimalisation: $(\mathcal{G}_m^-)^{-1}(e) = E$. On observe par exemple que le g-état 10 correspondant à la RNS se retrouve à la fois dans le m-état 1 et le m-état 6: $10 \in (\mathcal{G}_m^-)^{-1}(1) \bigcap (\mathcal{G}_m^-)^{-1}(6)$.

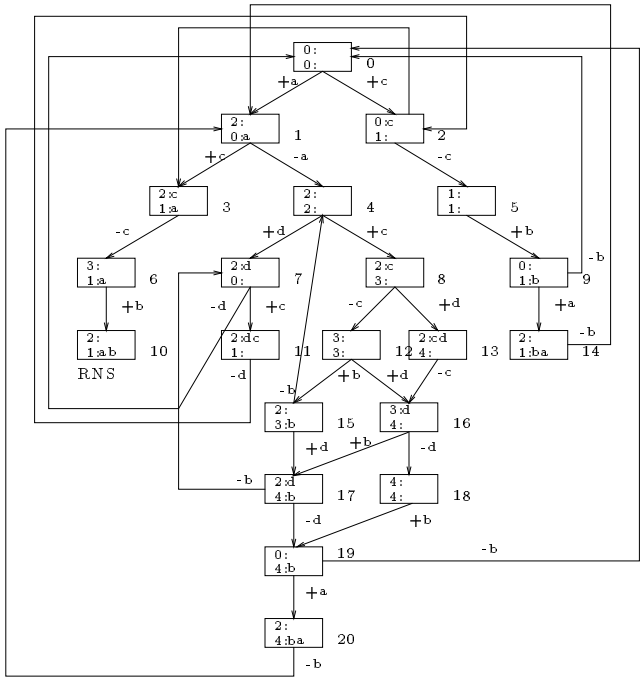

FIG. 3 - *Graphe d'accessibilité du protocole de l'exemple 1.*

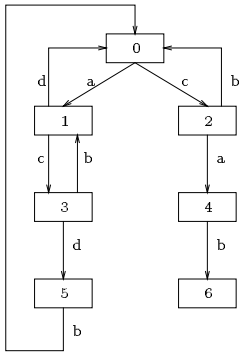

FIG. 4 - *Graphe d'accessibilité projeté, déterminisé puis minimisé ($\mathcal{G}_m^-$).*

5.3 Reconnaissance de l'anomalie et recherche de la transition à ajouter

On peut noter dans le graphe minimal présenté à la figure 4, un m-état-puits 6 et un autre m-état (le 4) à partir duquel il n'est plus possible d'éviter le blocage et on a :

états $\mathcal{G}_m^-$	états globaux
0	0, 7, 9, 17, 19
1	1, 4, 10, 14, 15, 20
2	2, 5, 11
3	3, 6, 8, 12
4	3, 6
5	13, 16, 18
6	10

FIG. 5 - *Correspondance entre les g-états de $\mathcal{G}$ et les m-états de $\mathcal{G}_m^-$.*

$(\mathcal{G}_m^-)^{-1}(4) \bigcup (\mathcal{G}_m^-)^{-1}(6) = \{3, 6, 10\}$ (cf figure 5). Lorsqu'on étudie ces g-états, on remarque qu'ils ont en commun le message *a* en tête de canal. Ce message ne peut être consommé et provoque la RNS. Une première idée est donc d'ajouter une transition de réception dans l'entité 2. Cette transition devra consommer le message *a* ; elle portera donc l'étiquette *-a*.

La figure 6 illustre la recherche de cette transition. L'état origine dans l'entité 2 de cette transition est l'état ou les états dans lesquels se trouvait l'entité 2 dans les g-états correspondant à la RNS. Ainsi d'après les g-états de blocage 3, 6 et 10 présenté à la figure 3, la transition a pour origine l'état 1 de l'entité 2.

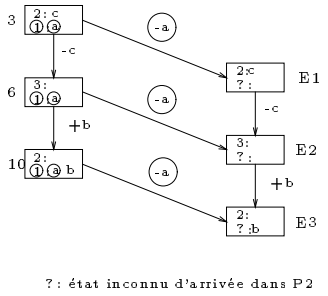

FIG. 6 - *Recherche de la transition à ajouter.*

Il nous faut maintenant rechercher l'état but de la transition. Pour pouvoir ajouter cette transition sans créer un nouvel état dans l'entité 2, il faut que les g-états buts E1, E2 et E3 recherchés sur la figure 6 soient présents dans le graphe d'accessibilité. On remarque par ailleurs, que si la transition à rajouter était une transition d'émission, les g-états E1, E2 et E3 n'existeraient pas dans le graphe d'accessibilité puisque les canaux sont de taille 2.

La recherche des g-états E1, E2 et E3 (par parcours des g-états du graphe d'accessibilité initial 3) indique que ce sont respectivement les g-états globaux 8, 12 et 15 qui y correspondent. En effet, dans chacun de ces g-états, l'entité 2 se trouve dans l'état 3. La transition à

rajouter relie donc les états 1 et 3 de l'entité 2 et est étiquetée *-a*. Le protocole comportant l'ajout de cette transition est présenté à la figure 7 et ne présente aucune anomalie [13].

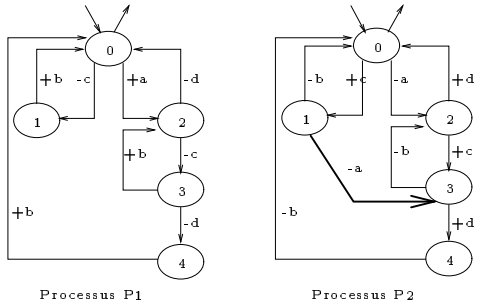

FIG. 7 - *Protocole de l'exemple 1 corrigé*

5.4 Bilan de l'exemple 1

Cet exemple montre que l'on peut supprimer les anomalies dans les protocoles sans tenir compte du service. Cependant rien ne prouve que le protocole obtenu après ces ajouts reste conforme au service initial. Pour être complet, il faut donc entreprendre la tâche fastidieuse supplémentaire qui consiste à vérifier la préservation du service. Afin d'éviter cette vérification après la correction, nous utiliserons le service dans la suite de notre travail pour déterminer les corrections à effectuer sur le protocole mais aussi pour garantir que le protocole obtenu reste conforme au service.

6 Exemple 2 et première utilisation du service

Nous utilisons à nouveau le protocole de l'exemple 1 décrit à la figure 2 mais cette fois-ci, nous allons tenter de corriger la RNS en nous aidant de la spécification du service.

6.1 Présentation du service associé à l'exemple 2

Une modélisation du service associé au protocole est présentée à la figure 8 et peut être résumée en ces termes : les messages *a* et *c* de demandes de mise à jour (*RQST* dans le protocole original) sont toujours suivis respectivement par les acquittements *d* et *b* (*DONE* dans le protocole original). Lors d'une collision entre les demandes de mise à jour, un traitement spécial permet d'assurer la cohérence des informations du réseau.
Remarque : La distinction des alphabets de messages des entités n'a pas d'incidence sur le service. En effet, le graphe de service obtenu en remplaçant les messages *a* et *c* par le message *RQST* et les messages *d* et *b* par le message *DONE* correspond (après déterminisation et minimisation) au graphe représentant le service du protocole original.

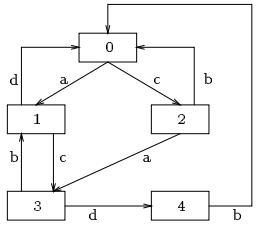

FIG. 8 - *Service associé au protocole de l'exemple 2.*

6.2 Comparaison du service avec le protocole

Comparons maintenant le graphe du service avec le graphe minimal obtenu par abstraction du graphe d'accessibilité décrivant les comportements du protocole. Le but de cette manipulation est de faire apparaître les différences entre ces deux graphes puis de les éliminer en modifiant le comportement du protocole. La comparaison de ces deux graphes est schématisée à la figure 9. Celui-ci est construit en parcourant parallèlement (c'est à dire en utilisant des transitions portant les mêmes étiquettes) chacun des deux graphes. La construction se termine lorsque l'un des graphes a été entièrement parcouru ou bien lorsque les différences entre les deux graphes ne permettent plus de continuer la construction. Ce graphe de comparaison peut donc être considéré comme une vue partielle du graphe minimal $\mathcal{G}_m^-$.

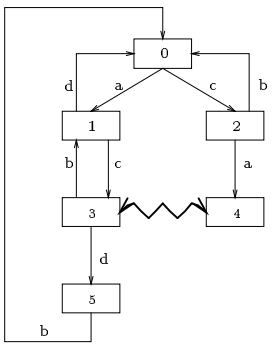

FIG. 9 - *Comparaison du graphe du service avec* $\mathcal{G}_m^-$.

L'étude de ce graphe montre que les m-états 0, 1, 2, 3 et 5 correspondent à un comportement correct du protocole. Le protocole diffère cependant du service lors du passage dans le m-état 4 : selon le service, la m-transition étiquetée *a* devrait amener le protocole non pas dans le m-état 4 mais dans le m-état 3. En d'autres termes, les m-états 3 et 4 de $\mathcal{G}_m^-$ devraient être *regroupés* en un seul m-état.

Compte tenu de la projection utilisée, l'unique façon de les regrouper est de rajouter une transition de réception puisque, lors de la projection Proj1 (cf paragraphe 4), les transitions de réception seront remplacées par des transitions vides. On a : $(\mathcal{G}_m^-)^{-1}(3) = \{3, 6, 8, 12\}$ et $(\mathcal{G}_m^-)^{-1}(4) = \{3, 6\}$ (cf figure 5) et la figure 10 illustre la recherche des transitions de réception (dans les entités) permettant de regrouper ces deux m-états.

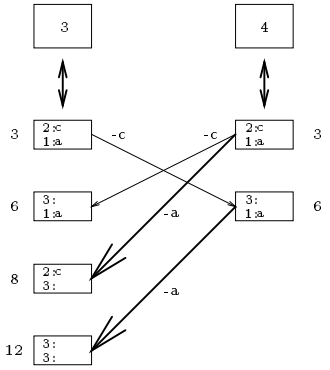

FIG. 10 - *Recherche de transitions de réception entre deux m-états de* $\mathcal{G}_m^-$.

On constate qu'il existe deux transitions de réception possibles : la transition (2,-c,3) de l'entité 1 qui existe déja (elle n'est donc pas à rajouter) et la transition de réception (1,-a,3) de l'entité 2. Dans le paragraphe 5, c'est l'ajout de cette transition qui a permis de corriger le protocole ; c'est rassurant.

On peut cependant remarquer que cet ajout va modifier le comportement de tous les g-états de la forme [?, λ, 1 ,a]. Or, seuls les deux g-états 3 et 6 sont de cette forme. En conséquence, le comportement du protocole n'est modifié que dans ces g-états; et c'est ce qui est recherché.

6.3 Bilan de l'exemple 2

Cet exemple montre qu'il est donc possible de corriger des protocoles auxquels il manque une transition de réception (c'est souvent le cas lors de réceptions non spécifiées). Pour cela, la comparaison entre le graphe $\mathcal{G}_m^-$ et le service indique quels m-états de $\mathcal{G}_m^-$ devraient être regroupés en un seul et même m-état. La connaissance de la projection utilisée permet ensuite de les regrouper. Si aucun autre état global de $\mathcal{G}$ n'est affecté par la correction (il peut en effet exister d'autres erreurs puisque la comparaison entre le protocole et son service est arrêtée dès que les graphes diffèrent), une première anomalie a été corrigée et le nouveau protocole ainsi obtenu préserve mieux (que dans l'exemple 1) la conformité au service.

Cependant, il est intéressant de souligner l'importance du type de projection effectué. En effet, on peut se demander si cette unique projection suffit lorsque la correction de la RNS nécessite l'ajout d'une transition d'émission plutôt qu'une transition de réception.

7 Exemple 3

Dans ce paragraphe, nous reprenons le protocole corrigé de la figure 7 mais nous y supprimons la transition d'émission $(2,+c,3)$ de l'entité 2. Le but de cette opération est de recommencer ce qui a été fait au paragraphe 6 mais en utilisant chacune des abstractions définies au paragraphe 4 pour voir dans quelle mesure, elles apportent une solution. Ceci permettra également d'étudier les différences induites par le changement de projection dans la correction du protocole.

Le graphe d'accessibilité du protocole est représenté à la figure 11 et ne possède aucune situation de blocage.

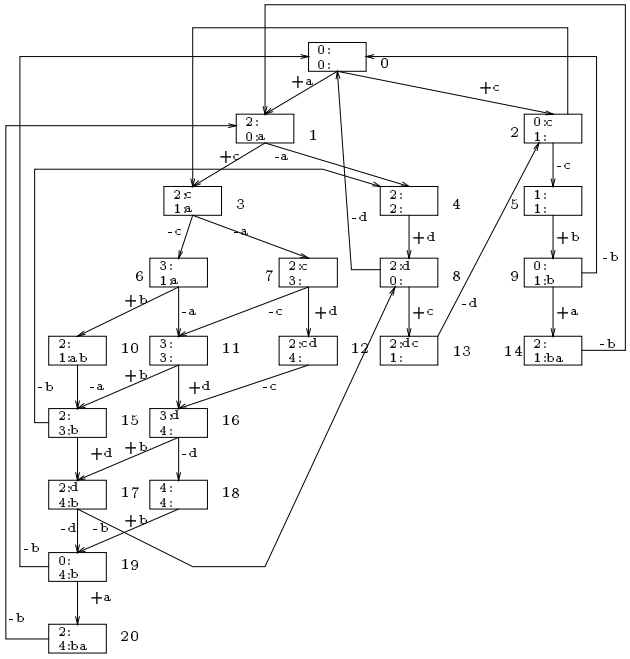

FIG. 11 - *Graphe d'accessibilité du protocole de l'exemple 3.*

7.1 Utilisation de la projection Proj2 : graphe minimal $\mathcal{G}_m^-$

Le graphe minimal $\mathcal{G}_m^-$ (obtenu en utilisant la projection Proj1) est représenté sur la figure 12. Il ne contient pas de situations de blocage mais ne correspond pas au service.

Comme précédemment, nous allons dans un premier temps comparer le protocole avec le service. La comparaison du protocole avec le service (cf figure 13) indique que la m-transition étiquetée b issue du m-état 3 ne devrait pas avoir comme m-état but le m-état 4 mais le

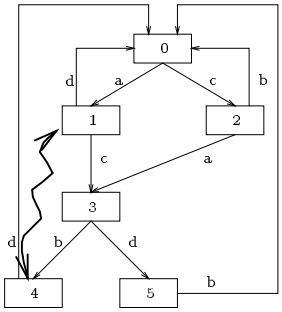

FIG. 12 - *Graphe minimal* $\mathcal{G}_m^-$.

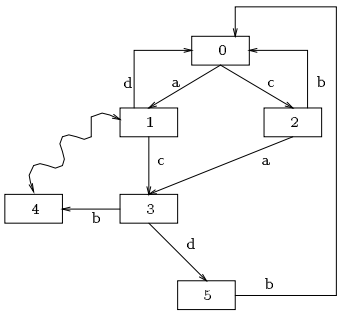

FIG. 13 - *Comparaison service-protocole.*

m-état 1. Les m-états 1 et 4 devraient donc être regroupés et on a : $(\mathcal{G}_m^-)^{-1}(1) = \{1,4,14,20\}$ et $(\mathcal{G}_m^-)^{-1}(4) = \{4,10,15\}$.

Il nous faut donc rechercher une transition de réception (à rajouter dans une des entités) pour *regrouper* ces deux m-états puisque la projection Proj1 transforme les transitions de réception en transitions vides. Nous recherchons donc une transition de réception qui pourrait relier les états globaux constituant les états 1 et 4 de $\mathcal{G}_m^-$. Cette recherche est représentée sur la figure 14. Les deux transitions détectées existent déjà dans le protocole : ce sont les transitions (0,-*a*,2) et (3,-*b*,2) de l'entité 2. Il n'apparaît donc pas de transition à rajouter (ce qui est rassurant puisqu'aucune transition de réception ne devait être rajoutée).

L'utilisation de la première abstraction ne nous permettant pas de déterminer quelle transition est à ajouter au protocole, il convient d'étudier ce que peut apporter la projection du graphe d'accessibilité en utilisant cette fois la projection Proj2 qui transforme les transitions d'émission en transitions vides.

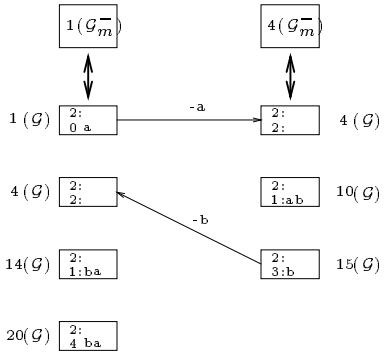

FIG. 14 - *Recherche d'une transition de réception regroupant les m-états 1 et 4 de* $\mathcal{G}_m^-$

7.2 Utilisation de la projection Proj2 : graphe minimal $\mathcal{G}_m^+$

Le graphe $\mathcal{G}_m^+$ obtenu en utilisant Proj2 est identique au graphe $\mathcal{G}_m^-$ présenté à la figure 12; ceci est dû au fait que les deux entités sont symétriques. On a également le même graphe issu de la comparaison entre le protocole et le service (cf figure 13). Pour corriger le protocole, nous constatons (comme précédemment) que les m-états 1 et 4 de $\mathcal{G}_m^+$ devraient être regroupés. Le fait de changer de projection intervient ici. En effet, d'une part les g-états associés en utilisant la projection Proj2, ne sont plus les mêmes qu'avec Proj1; par exemple, on a : $(\mathcal{G}_m^+)^{-1}(1) = \{4,7,8,12,13\}$ et $(\mathcal{G}_m^+)^{-1}(4) = \{4,8,13\}$. D'autre part pour effectuer le regroupement des deux m-états 1 et 4, nous utiliserons des transitions d'émission puisque ce sont celles qui ont été transformées en transitions vides par la projection Proj2 utilisée. La recherche de transitions d'émission pouvant être créées est schématisée à la figure 15. Deux des trois transitions détectées existent déjà. Seule la transition $(2,+c,3)$ de l'entité 2 est nouvelle et c'est bien la transition que nous voulions trouver.

7.3 Bilan de l'exemple 3

Les abstractions ont été choisies au départ pour leur pertinence dans la correction des RNS. Cet exemple montre qu'elles peuvent être utilisées également pour traiter des protocoles qui ne comportent pas de situations de blocage mais qui sont non conformes au service.

Nous avons successivement essayé les deux projections. Ceci nous a permis de déterminer la transition à rajouter, qu'il s'agisse d'une transition d'émission ou d'une transition de réception. Cependant, on peut soupçonner la symétrie des entités d'avoir rendu ceci possible.

Par ailleurs, les transitions ajoutées dans cet exemple ne nécessitent pas la création de nouveaux états dans les processus. Or, il est évident que l'ajout de nouveaux états soient parfois nécessaires pour assurer le bon fonctionnement d'un protocole.

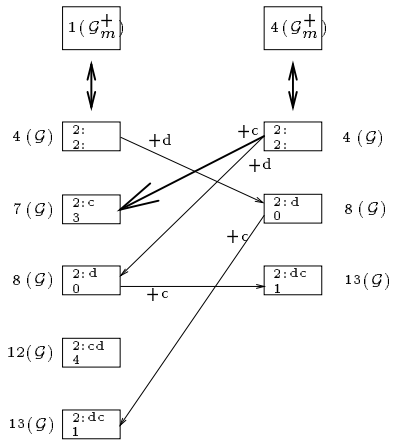

FIG. 15 - *Recherche d'une transition d'émission reliant les états 1 et 4 de* $\mathcal{G}_m^+$

Il nous faut donc étudier d'autres exemples avant de conclure à la possibilité de systématiser l'approche utilisée.

8 Autres exemples

Nous avons étudié d'autres exemples de protocole mettant en évidence des problèmes dont il faudra tenir compte dans une démarche visant à développer une méthodologie globale de correction. Nous présentons ci-après les principaux résultats de l'utilisation des deux mêmes abstractions précédentes. Le détail des étapes se trouve dans [13].

8.1 Exemple 4

Nous avons supprimé dans le protocole de l'exemple 1 corrigé, les deux transitions symétriques (3,+*b*,2) et (3,-*b*,2). Le protocole obtenu ne comporte aucune anomalie de fonctionnement mais n'est pas confrome au service souhaité.

La comparaison des graphes minimaux $\mathcal{G}_m^+$ et $\mathcal{G}_m^-$ avec le graphe du service montre qu'une transition étiquetée *b* doit relier le m-état 3 au m-état 1 des graphes $\mathcal{G}_m^+$ et $\mathcal{G}_m^-$. Pour ce faire, il reste trois problèmes à résoudre :

- déterminer le type de la transition : émission ou réception,
- trouver l'entité dans laquelle cette transition doit être ajoutée,
- et enfin déterminer les états origine et but de la transition.

L'utilisation de la première projection Proj1 nous a permis d'ajouter une transition d'émission dans le protocole alors que la seconde projection ne nous indique aucune modification à effectuer. A l'issue de l'addition de la transition d'émission, une nouvelle itération de la méthode nous a alors permis d'ajouter la transition de réception manquante.

Cet exemple montre que la correction d'un protocole où plusieurs transitions sont absentes, peut donc également être effectuée à l'aide du service même si cette correction oblige à itérer la méthode.

Les paragraphes 6, 7 et 8.1 nous ont permis de montrer qu'il était possible de corriger un protocole de communication en s'aidant du service et en assurant que le protocole était conforme au service. Cependant les corrections étaient relativement simples puisqu'elles consistaient à ajouter une transition à l'un et/ou à l'autre des entités du protocole. Cependant, il existe des protocoles pour lesquels il est nécessaire de créer de nouveaux états dans les entités communicantes.

8.2 Exemple 5

Nous avons considéré un protocole de type connexion-déconnexion [10] présentant une RNS. En tentant de la corriger avec la méthode utilisée dans les paragraphes précédents, on aboutit au fait qu'aucune des deux projections ne nous propose de solution permettant de corriger le protocole tout en assurant le service demandé.

Ceci montre que l'approche de résolution utilisée jusqu'ici n'est pas généralisable. Malgré des ajouts d'informations complémentaires dans le service, la correction du protocole exigeait l'ajout de nouveaux états dans les entités. Nous avons alors défini des règles d'ajout d'états dans les entités et quatre itérations de la méthode ont été nécessaires pour aboutir à un protocole sans RNS et conforme au nouveau service.

9 Conclusion

Dans notre démarche visant à développer une méthodologie globale pour supprimer les anomalies dans les protocoles, nous avons expérimenté une approche sur quelques exemples significatifs de protocoles. Cette approche orientée vers l'ajout incrémental de transitions dans les entités communicantes consiste à effectuer une abstraction du graphe d'accessibilité. Cette opération a pour but de faire ressortir les informations utiles à la correction. Ensuite une comparaison de cette abstraction avec une spécification du service permet par ajout incrémental de transitions dans les entités communicantes de supprimer les anomalies dans un protocole tout en garantissant sa conformité au service initial.

Dans l'optique d'une généralisation de l'approche, nous étudions actuellement d'autres exemples de protocoles. A défaut d'une méthode globale pouvant s'appliquer à n'importe quel protocole, notre objectif est d'obtenir les classes de protocoles pour lesquelles cette approche est applicable.

Par ailleurs, nous avons considéré principalement deux abstractions. L'étude d'autres abstractions telles que les bisimulations apporterait certainement des éléments complémentaires pour la suppression des anomalies dans les protocoles.

Enfin, notre travail utilise le graphe d'accessibilité qui comporte généralement un très grand nombre d'états. Une autre étude intéressante serait de voir dans quelle mesure les techniques de réduction du graphe d'accessibilité conservent les possibilités de correction.

Références

[1] C. Berge. *Théorie des graphes et ses applications.* Gauthiers-Villars, Paris, 1983.

[2] G.V. Bochman. Finite state description of communication protocols. *Computer networks 2*, 361–372, February 1978.

[3] G. W. Brams. *Théorie et pratiques.* Volume 1 et 2, Masson, Paris, 1983.

[4] M. Diaz and G Guidacci Da Silveira. Specification and validation of protocols by temporal logic and nets. *3rd European Workshop on Applications and Theory of Petri Nets*, September 1982. Varenna Italy.

[5] A. Gill. *Introduction To The Theory Of Finite-Sate Machines.* McGRAW-HILL BOOK COMPANY, 1962.

[6] M. G. Gouda and Ji-Yun Han. A finite state model for protocol processes and services. *IEEE Transaction on Communication*, COM-28:624–631, April 1980.

[7] M.G. Gouda. Closed covers : to verify progress for communicating finite state machines. *IEEE transactions on software enginering*, SE-10(6):846–855, November 1984.

[8] G. Holzmann. Design and validation of computer protocol. *Prentice Hall*, 1991.

[9] G. Holzmann. On limits and possibilities of automated protocol analysis. *PSTV VII - Protocol Specification Testing and Verification*, June 1987.

[10] C. Jard and M. Raynal. *De la nécessité de spécifier des propriétés pour la vérification des algorithmes distribués.* Technical Report, INRIA, December 1986.

[11] D. R. Musser, J. V. Guttag, and E. Horowitz. Abstract data types and software validation. *Communication of the Ass. Comput. Mach (ACM)*, 21(12):1048–1064, December 1978.

[12] C.V. Ramamoorthy, S.T. Dong, and Y. Usuda. An implementation of an automated protocol synthesizer (aps) and its application to the x.21 protocol. *IEEE Transactions on Software Enginering*, SE-11(9):886–908, September 1985.

[13] B. Richard. *Utilisation du service pour la correction des anomalies dans les protocoles de communication.* Technical Report, DEA - Université de Rennes I, September 1994.

[14] J. Rubin and C.H. West. An improved protocol validation technique. *Computer Networks*, 65–73, 1982.

[15] G. Viho. Comment utiliser le service pour la correction des anomalies dans les protocoles : étude de quelques exemples. *CFIP'95 - Colloque Francophone sur l'Ingénierie des Protocoles*, à paraître, May 1995.

[16] G. Viho. Errors suppression in protocols and service presevation. *International Telecommunication Symposium ITS'94*, 398–402, August 1994.

[17] P. Zafiropulo, C. West, H. Rudin, D.D. Cowan, and D. Brand. Towards analysing and synthesing protocols. *IEEE Transactions on Communications*, COM-28(4):651–661, April 1980.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399