Raffinement B de systèmes de transitions étiquetés

Inès Mouakher, Francis Alexandre

Raffinement B de systèmes de transitions étiquetés

Inès Mouakher, Francis Alexandre
LORIA – Nancy Université
Campus scientifique
F-54506 Vandœuvre-lès-Nancy cedex
{*Ines.Mouakher, Francis.Alexandre*}*@loria.fr*

24 novembre 2009

Résumé

Le raffinement est une notion clé dans la méthode B. Nous étudions le rapport entre cette notion et certaines relations entre systèmes de transitions étiquetés telles que la simulation, la bisimulation... Le point de départ de l'étude est la définition d'une traduction particulière des STEs en spécification B. Nous présentons, ensuite, des schémas composés de spécifications B basées sur les clauses de modularité **INCLUDES** et **REFINES**. Pour chacun de ces schémas nous détaillons une proposition en terme de relation entre STEs. Les schémas mis en évidence pourront ensuite être utilisés dans le cadre de l'assemblage des composants.

Mots clés : méthode B, raffinement, système de transitions étiquetés, simulation

Abstract

Refinement is an important technique of B method. We study the relationship between B refinement and some relations between labeled transition systems (LTS) such that simulation or bisimulation ... The starting point of the study is the definition of a translation of LTSs into B specifications. Then we consider some diagrams composed of B specifications based on the clauses **INCLUDES** and **REFINES**. For every diagram we give a proposition in terms of relation between LTSs. These diagrams are then used to verify the correctness of the component assembly at the protocol level.

Keywords : B method, refinement, labeled transition systems (LTS), simulation

1 Préliminaires

Nous introduisons dans ce paragraphe des définitions préliminaires relatives aux systèmes de transitions étiquetés et à certaines relations entre les systèmes de transitions (simulation, bisimulation, ready-simulation). Nous présentons, également, une nouvelle notion appelée composition d'étiquettes qui permet de mettre des STEs en relation et de modéliser le raffinement B.

1.1 Système de transitions étiqueté (STE)

Definition 1.1 (Système de transitions étiqueté (STE))**.**
Un STE est un quintuplet $T = (A,\ S,\ s_0,\ F,\ \rightarrow)$ où :

- A est un ensemble d'étiquettes
- S est un ensemble d'états
- $s_0 \in S$ désigne l'état initial de T
- $F \subseteq Q$ est le sous-ensemble des états finals
- $\rightarrow \subseteq S \times A \times S$ est la relation de transition du système T

On dénote $s \xrightarrow{e} s'$ la transition $(s,\ e,\ s')$ appartenant à $\rightarrow$.
Dans la suite lorsque nous considérons un STE T, nous supposons que $T = (A,\ S,\ s_0,\ F,\ \rightarrow)$.

Definition 1.2 (Étiquette et STE)**.**
Soient T un STE et e une étiquette.

- $pre(e,\ T) = \{s\ ,\ s \in S\ \wedge\ \exists s' \in S\ .\ (s,\ e,\ s') \in \rightarrow\}$ désigne l'ensemble des états à partir desquels l'étiquette e peut être déclenchée, on le note $pre(e)$ s'il n'y a pas d'ambiguïté sur le STE considéré.
- $post(e,\ \rightarrow) = \{s'\ ,\ s' \in S\ \wedge\ \exists s \in S\ .\ (s,\ e,\ s') \in \rightarrow\}$ désigne l'ensemble des états obtenus après déclenchement de l'étiquette e, on le note $post(e)$ s'il n'y a pas d'ambiguïté sur le STE considéré.
- $s \xrightarrow{e}$ signifie qu'il existe un élément s' de S tel que $s \xrightarrow{e} s'$, c'est-à-dire que $s \in pre(e)$.
- $\xrightarrow{e} = \{(s, s')\ ,\ s \in S \wedge s' \in S \wedge (s,\ e,\ s') \in \rightarrow\}$

Definition 1.3 (Chemins et traces)**.**

- Un chemin c dans T est une suite finie de transitions de la forme $q_i \xrightarrow{e_i} q_{i+1}$ où $0 \leq i \leq n$ telle que $q_0 = s_0$ et $q_{n+1} \in F$. $q_0 = origine(c)$ est l'origine du chemin c. $q_{n+1} = extremite(c)$ est l'extrémité du chemin c. L'ensemble des chemins dans T est noté $Chemin(T)$.
- La suite d'étiquettes $(e_i)_{0 \leq i \leq n}$ associée au chemin précédent est appelée la trace du chemin, l'ensemble des traces finies associées à T est noté $Trace(T)$.

- Nous étendons la notion de chemin à des origines et des extrémités quelconques. Une suite $q_i \xrightarrow{e_i} q_{i+1}$ où $1 \leq i \leq n$ est un chemin initialisé dans l'état q_0 et aboutissant à l'état q_{n+1}. On note $Chemin(T,\ q_0,\ q_{n+1})$ l'ensemble de ces chemins.
  Soient E et E' deux sous-ensembles de S, $Chemin(T,\ E,\ E') = \bigcup_{s \in E, s' \in E'} Chemin(T,\ s,\ s')$.
  Si $\mathcal{C}$ est un ensemble de chemins, on note $origin(\mathcal{C}) = \{origine(c),\ c \in \mathcal{C}\}$, l'ensemble constitué des origines des chemins de $\mathcal{C}$.

Nous définissons, maintenant, le produit libre de plusieurs STEs.

Definition 1.4 (Produit libre [Arn92])**.** Soient $T_i = (A_i, S_i, s_0^i, F_i, \rightarrow_i)$, $1 \leq i \leq n$ des STEs. Le STE associé à ce système est $T = (A,\ S,\ s_0,\ F,\ \rightarrow)$ avec :

1. $A = \bigcup_{i=1}^{n} A_i$
2. $S = S_1 \times \ldots \times S_n$
3. $s_0 = (s_0^1, \ldots, s_0^n)$
4. $F = F_1 \times \ldots \times F_n$
5. $\rightarrow$ désigne l'ensemble des transitions de la forme $((s_1, \ldots,\ s_n), e, (s'_1, \ldots,\ s'_n))$ et telles que $\exists\ k \in [1\ ..\ n]$ vérifiant les conditions suivantes :
   (a) $(s_k, e,\ s'_k) \in \rightarrow_k$
   (b) $(\forall i \in [1\ ..\ n] \setminus \{k\})\ s_i = s'_i$

1.2 Système de transitions étiqueté gardé

Nous étendons la définition des STEs en tenant compte des gardes. Ainsi, on les appelle systèmes de transitions étiquetés gardés.

Definition 1.5 (Système de transitions étiqueté gardé)**.**
Un STE gardé est un sextuplet $T = (A,\ G(x),\ S,\ s_0,\ F,\ \rightarrow)$ où :

- A, S, s_0 et F sont définis de la même manière que dans un STE.
- G est un ensemble de prédicats. Ces prédicats sont définis en termes des variables x (dépendant du contexte) ou par vrai.
- $\rightarrow \subseteq S \times G(x) \times A \times S$ est la relation de transition du système T.

Definition 1.6 (Étiquette et STE)**.**
Soient T un STE gardé, e une étiquette, q un état de S et Q un ensemble d'états de S :

- $q \overset{[?]e}{\rightarrow}$ est un prédicat qui est défini par : $(\exists q' \in S)\ (\exists g\ \in G)\ [q \overset{[g]e}{\rightarrow} q']$
- $q \overset{[?]e}{\rightarrow} q'$ est un prédicat qui est défini par : $(\exists g\ \in G)[q \overset{[g]e}{\rightarrow} q']$

- $gr(e,q,T)$ est défini par : $\bigvee_{g\in G}(g)$ tel que $(\forall g \in G)\ (\exists q' \in S)[q \overset{[g]e}{\rightarrow} q' \in \rightarrow]$
- $gr(e,q,Q,T)\ \bigvee_{g\in G}(g)$ tel que $(\forall g \in G)\ (\exists q' \in S)[q \overset{[g]e}{\rightarrow} q' \in \rightarrow \wedge q' \in Q]$

Definition 1.7 (Transition valide)**.**
La transition $(s,\ [g]\ e,\ s')$ peut être activée à partir de l'état s dans un contexte où sa garde g est vraie. Le déclenchement de l'événement e génère l'état s'. Dans ce cas, la transition est valide. Elle est notée $s \xrightarrow{[g]\ e} s'$.

1.3 Propriétés des systèmes de transitions étiquetés

1.3.1 Déterminisme

Definition 1.8 (STE déterministe)**.** T est déterministe s'il vérifie la propriété suivante :

$$(\forall e \in A)\ (\forall q \in S)\ (\forall q' \in S)\ (\forall q'' \in S)\ [q \xrightarrow{e} q' \wedge q \xrightarrow{e} q'' \Rightarrow q' = q'']$$

1.3.2 Relations entre systèmes de transitions

Definition 1.9 (Relation de simulation entre deux STEs)**.** Soient $T_1 = (A,\ S_1,\ s_0^1,\ \rightarrow_1)$ et $T_2 = (A,\ S_2,\ s_0^2,\ \rightarrow_2)$ deux STEs et R une relation de S_1 vers S_2. T_2 simule T_1 par rapport à R (noté $T_2 \preccurlyeq_R T_1$) ssi :

- $(s_0^1, s_0^2) \in R$.
- $(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ (\forall p' \in S_1)$
  $[(p,q) \in R \wedge p \xrightarrow{e}_1 p' \Rightarrow\ (\exists q' \in S_2)\ [(p',q') \in R \wedge\ q \xrightarrow{e}_2 q']]$

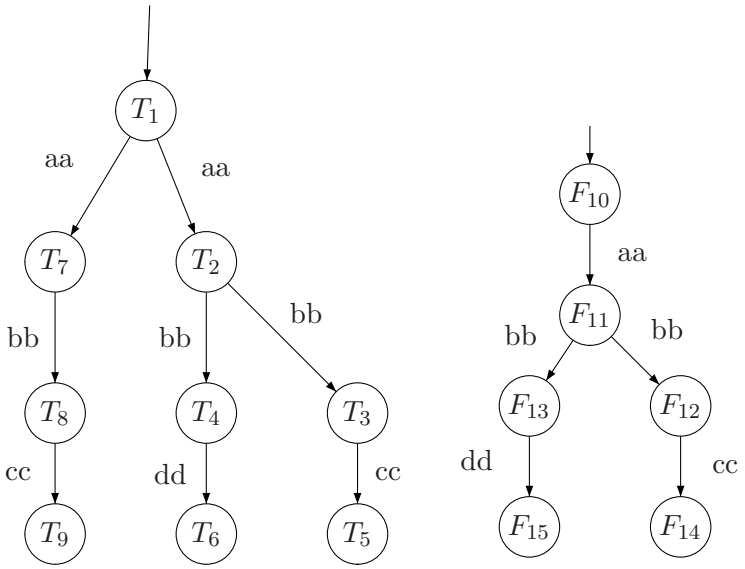
Exemple 1.1. Soient STE_1 et STE_2 les deux STEs de la figure 1 et
$R = \{(F_{10},\ T_1), (F_{11},\ T_2), (F_{11},\ T_7),$
$(F_{12},\ T_3), (F_{12},\ T_8), (F_{13},\ T_4), (F_{14},\ T_5), (F_{14},\ T_9), (F_{15},\ T_6)\}$, on a $STE_1 \preccurlyeq STE_2$. La relation R peut être calculée en partant de couple $(F_{10},\ T_1)$ formant les états initiaux et en générant la suite des couples :
$\Rightarrow (F_{11},\ T_2) \in R \wedge (F_{11},\ T_7) \in R$
$\Rightarrow (F_{12},\ T_3) \in R \wedge (F_{12},\ T_8) \in R \wedge (F_{13},\ T_4) \in R$
$\Rightarrow (F_{14},\ T_5) \in R \wedge (F_{14},\ T_9) \in R \wedge (F_{15},\ T_6) \in R$

Definition 1.10 (Relation de bisimulation entre deux STEs)**.** Soient $T_1 = (A,\ S_1,\ s_0^1,\ \rightarrow_1)$ et $T_2 = (A,\ S_2,\ s_0^2,\ \rightarrow_2)$ deux STEs et R une relation de S_1 vers S_2. T_2 et T_1 sont bisimilaires par rapport à R (noté $T_2 \approx_R T_1$) ssi :

- $T_2 \preccurlyeq_R T_1$
- $T_1 \preccurlyeq_{R^{-1}} T_2$

La bisimilarité implique l'équivalence de traces. Si les STEs T_1 et T_2 sont déterministes, l'équivalence de trace implique la bisimilarité.

La relation de ready-simulation est introduite par Bloom et al. [BIM95].



(a) STE_1 (b) STE_2

Figure 1: STE

Definition 1.11 (Relation de ready-simulation [BIM95])**.** Soient $T_1 = (A,\ S_1,\ s_0^1,\ \rightarrow_1)$ et $T_2 = (A,\ S_2,\ s_0^2,\ \rightarrow_2)$ deux STEs et R une relation de S_1 vers S_2. T_2 et T_1 sont ready-similaires par rapport à R (noté $T_2 \precsim_R T_1$) ssi :

- $T_2 \preccurlyeq_R T_1$
- $(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ \ [(p,q) \in R \wedge (\exists q' \in S_2)\ [q \xrightarrow{e}_2 q'] \Rightarrow (\exists p' \in S_1)\ [p \xrightarrow{e}_1 p']]$

1.4 Composition d'étiquettes

Nous introduisons une nouvelle notion appelée composition d'étiquette. À chaque composition, nous associons un ensemble de chemins, cela nous permet aussi d'établir des relations entre des STEs définis des alphabets différents. La traduction systématique des compositions d'étiquettes dans le langage B nous permet de donner une caractérisation du raffinement pour des STEs définis sur des alphabets différents.

Definition 1.12 (Composition d'étiquettes)**.** Soient $T = (A,\ S,\ s_0,\ F,\ \rightarrow)$ un STE, P l'ensemble de prédicats unaires de profil $S \rightarrow Bool$ et $\{seq, alt, opt, loop\}$ un ensemble de constructeurs. L'ensemble SUB (noté aussi $\mathcal{E}(A)$) des compositions d'étiquettes associées à T est défini inductivement de la façon suivante :

- $\mathcal{B} = \{\epsilon\} \cup A$ est l'ensemble des compositions d'étiquettes de base
- Les règles de formation des compositions d'étiquettes sont :

– $(\forall sub_1 \in\ SUB)\ (\forall sub_2 \in\ SUB)\ sub_1\ seq\ sub_2 \in\ SUB$
– $(\forall sub_1 \in\ SUB)\ (\forall sub_2 \in\ SUB)\ sub_1\ alt\ sub_2 \in\ SUB$
– $(\forall sub \in\ SUB)\ (\forall p \in P)\ opt\ p\ sub \in\ SUB$
– $(\forall sub \in\ SUB)\ (\forall p \in P)\ (\forall n \in \mathbb{N})\ loop\ p\ n\ sub \in\ SUB$

Proposition 1.1 (Forme canonique)**.** *Toute composition d'étiquettes peut se mettre sous la forme : sub_1 alt...alt sub_m où chaque sub_i est de la forme : $opt(g_1)e_1$ seq... seq $opt(g_n)e_n$.*
Cette forme de composition d'étiquette est appelée forme canonique.

Proof. La preuve de cette formule découle directement de la traduction des compositions d'étiquettes en substitutions généralisées (Tab. 1) et de la proposition 2.1 et de la proposition qui exprime comment on transforme les substitutions B. □

Definition 1.13 (Précondition d'une composition d'étiquettes)**.** On considère des compositions d'étiquettes associées à un STE T, la précondition d'une composition d'étiquette sub est notée par $pre(sub, T)$ ou $pre(sub)$, c'est un ensemble d'états de T défini comme suit.

- si $sub = opt(g_1)skip$, $pre(sub) = \{q_1; q_1 \in S\}$
- si $sub = opt(g_1)e_1\ seq...\ seq\ opt(g_n)e_n$,

$$\begin{aligned} pre(sub) = \quad & \{q_1; q_1 \in S \bigwedge_{j=1}^{n} [(\forall q_2 \in S), ..., (\forall q_n \in S) \\ & q_1 \xrightarrow{e_1} q_2 \xrightarrow{e_2} ... \xrightarrow{e_{j-2}} q_{j-1} \xrightarrow{e_{j-1}} q_j \wedge g_1(q_1) \wedge ... \wedge g_j(q_j) \Rightarrow q_j \xrightarrow{e_j}]\} \end{aligned}$$

- Pour toute composition d'étiquette sub de forme canonique $sub_1\ alt...alt\ sub_m$, $pre(sub) = \bigcap_{i=1}^{m} pre(sub_i)$.

La notion de chemin associé à une composition d'étiquettes se définit à partir de la notion de chemin valide.

Definition 1.14 (Chemins associés à une composition d'étiquettes)**.**
Soient $T = (A,\ S,\ s_0,\ F,\ \rightarrow)$ un STE.

1. Chemins valides.
   - Un chemin dans T de la forme $s_i \xrightarrow{e_i} s_{i+1}$ où $1 \leq i \leq n$ est valide par rapport à une composition d'étiquettes sub de la forme $(opt\ g_1\ e_1)\ seq...seq\ (opt\ g_n\ e_n)$ si et seulement si $\forall i \in [1, n]\ g_i(s_i)$. L'ensemble des chemins dans T valides par rapport à sub est noté $CheminVal(sub, T)$ ou $CheminVal(sub)$ lorqu'il n'y a pas d'ambiguïté sur T.
     Un chemin vide dans T d'origine et extrémité égale à s_1 est valide par rapport à une composition d'étiquettes sub de la forme $(opt\ g_1\ \epsilon)$ si et seulement si $g_1(s_1)$.
   - L'ensemble des chemins valides dans T par rapport à une composition d'étiquettes sub sous forme canonique (c-à-d de forme $sub_1\ alt...alt\ sub_m$) est défini par : $CheminVal(sub, T) = \bigcup_{i=1}^{m}\ CheminVal(sub_i, T)$.

2. Chemins associés à une composition d'étiquettes.
   L'ensemble des chemins associées à une composition d'étiquettes *sub* est défini par l'ensemble des chemins valides par rapport à *sub* et dont l'origine appartient à la précondition de *sub* (c-à-d. les chemins déclenchables), on le note $Chemin(sub, T)$.
   $Chemin(sub, T) = \{c; c \in CheminVal(sub, T) \wedge origine(c) \in pre(sub)\}$.

La notion de composition d'étiquettes a été définie relativement à un STE, pour prendre en compte plusieurs STE indépendants, nous introduisons la définition suivante.

Definition 1.15. Soient T_1,..., T_n, n STEs, dont les alphabets sont deux à deux disjoints, et T leur produit libre. Une composition d'étiquettes sur T_1,...,T_n est définie par une composition d'étiquettes sur T.

2 Traduction en B

2.1 Système de transitions étiqueté

Dans ce paragraphe nous définissons la traduction des systèmes de transitions étiquetées en des machines B. Soit $T = (A,\ S,\ s_0,\ F,\ \rightarrow)$ un système de transition étiqueté, tel que $S = \{s_0, \ldots, s_n\}$, nous lui associons le modèle B suivant que nous appelons aussi T et qui est défini par les éléments suivants :

- Chaque état de T est modélisé par une variable $stateT$, appelée variable de contrôle dont les valeurs possibles appartiennent à S.
- Pour chaque étiquette $e \in A$, nous considérons la suite des transitions de T de la forme (s_i, e, s'_i) et introduisons une opération B de la forme $e = P|B$ où P est une précondition et B correspondant au corps de l'opération. Plus précisément :

$$P = \bigvee_{i=1}^{m} (stateT = s_i)$$

$$\begin{aligned} B = {} & stateT = s_1 \Rightarrow\ stateT := s'_1 [] \\ & ... [] \\ & stateT = s_m \Rightarrow\ stateT := s'_m [] \end{aligned}$$

P peut aussi s'écrire : $stateT \in pre(e, \rightarrow)$. L'équivalent verbeux de la substitution B est :

$$\begin{aligned} & SELECT stateT = s_1\ THEN\ stateT := s'_1 \\ & \quad WHEN stateT = s_2\ THEN\ stateT := s'_2 \\ & \quad \ldots \\ & \quad WHEN stateT = s_m\ THEN\ stateT := s'_m \\ & END \end{aligned}$$

Dans la traduction présentée le lien entre un STE et sa spécification B est évident puisque les opérations modélisent directement les transitions du STE. Si l'on se place au point de vue de la sémantique opérationnelle en B, le modèle B obtenu par traduction a comme sémantique le STE dont il est lui même la traduction, cette dernière remarque peut être corroborée par proB [LB03, LB08] lorqu'on lui soumet des traductions de STEs.

Exemple 2.1. La figure 2 présente les modèles B associés aux deux STEs de la figure 1. Ces deux STEs ont les mêmes étiquettes.

```
MODEL M1
SETS
  T_States={T1,T2,T3,T4,T5,T6,T7,T8,T9}
VARIABLES stateT
INVARIANT stateT ∈T_States
INITIALISATION stateT:=T1
OPERATIONS
 aa =
  PRE stateT=T1 THEN
   IF stateT = T1 THEN
    stateT:∈{T2,T7}
   END
  END;
 bb=
  PRE stateT∈{T2,T7} THEN
   SELECT stateT = T2 THEN stateT:∈{T3,T4}
    WHEN stateT = T7 THEN stateT:=T8
   END
  END;
 cc=
  PRE stateT∈{T3,T8} THEN
   SELECT stateT = T3 THEN stateT:=T5
    WHEN stateT = T8 THEN stateT:=T9
   END
  END;
 dd=
  PRE stateT=T4 THEN
   SELECT stateT = T4 THEN stateT:=T6
   END
  END
END
```

(a) STE_1

```
MODEL M2
SETS
 F_States = {F10,F11,F12,F13,F14,F15}
VARIABLES stateF
INVARIANT stateF ∈F_States
INITIALISATION stateF:=F10
OPERATIONS
 aa =
  PRE stateF =F10 THEN
   IF stateF = F10 THEN
    stateF:=F11
   END
  END;
 bb=
  PRE stateF =F11 THEN
   IF stateF = F11 THEN
    stateF:∈{F12,F13}
   END
  END;
 cc=
  PRE stateF =F12 THEN
   IF stateF = F12 THEN
    stateF:=F14
   END
  END;
 dd=
  PRE stateF =F13 THEN
   IF stateF = F13 THEN
    stateF:=F15
   END
  END
END
```

(b) STE_2

Figure 2: Modèles B

2.2 Relation entre les états de deux STEs

Nous somme amenés à définir des relations entre les états de deux STEs. En B, il est simple de définir une relation binaire par un ensemble de couples, un couple étant de la forme $x \mapsto y$. Il faut toute fois s'assurer de l'accès en lecture des données permettant de définir la relation. Nous déclarons les relations dans la clause **ABSTRACT_CONSTANTS** et les définissons dans la clause **PROPERTIES**. L'ajout de la formule (stateF $\mapsto$ stateT) $\in$ RR dans la clause **INVARIANT** permet de mettre en relation par RR les états des deux STEs F et T.

Exemple 2.2. La figure **??** présente une relation entre les deux STEs

```
ABSTRACT_CONSTANTS
    RR
PROPERTIES
    RR ∈ F_States ↔ T_States ∧
    RR = {(F10 ↦T1),
          (F11 ↦ T2),(F11 ↦T7),
          (F12 ↦T3),(F12 ↦T8),
          (F13 ↦T4),
          (F14 ↦T5),(F14 ↦T9),
          (F15 ↦T6)}
```

Figure 3: Relation entre les variables de M1 et M2

2.3 Composition d'étiquettes

2.3.1 Traduction en substitutions généralisées

Nous définissons une correspondance entre l'ensemble des compositions d'étiquettes et l'ensemble des substitutions généralisées sous forme d'une application Φ. L'ensemble, SUB, des compositions d'étiquettes ayant été défini par induction (Déf. 1.12, p. 5) l'application Φ est naturellement récursive. La traduction définie par Φ est immédiate. La traduction de la composition d'étiquettes vide ϵ est la substitution généralisée, **skip**. La traduction d'une étiquette e est la substitution généralisée associée à e, que nous notons également e. Lorsque e doit être explicitée, nous lui faisons correspondre la substitution généralisée obtenue par la traduction du STE considéré. Aux constructeurs *seq*, *alt*, *opt* et *loop* nous associons respectivement les constructeurs ; (séquence) $[]$ (substitution de choix borné), $\Rightarrow$ (substitution gardée) et $\mathcal{W}$ (boucle), nous nous restreignons aux boucles ¡¡pour¿¿. Le tableau 1 montre la définition de Φ.

Composition d'étiquettes (sub)	Substitutions généralisées ($\Phi(sub)$)
ϵ	$skip$
e	e_1
$sub_1\ seq\ sub_2$	$\Phi(sub_1); \Phi(sub_2)$
$sub_1\ alt\ \ sub_2$	$\Phi(sub_1)\ [] \ \Phi(sub_2)$
$opt\ p\ sub_1$	$p \Rightarrow \Phi(sub_1)$
$loop\ p\ n\ sub_1$	$\mathcal{W}(n > 1, \Phi(sub_1), p, n)$

Table 1: Correspondance entre composition d'étiquettes et substitutions généralisées

La définition de Φ a une double conséquence :

- comme SUB est défini inductivement $\Phi(SUB)$ peut aussi être défini inductivement avec *skip* et e comme substitutions de base et des règles identiques à celles de la construction inductive de SUB en remplaçant les constructeurs *seq*, *alt*, *opt* et *loop* respectivement par ;, $[]$, $\Rightarrow$ et $\mathcal{W}$, les propriétés de $\Phi(SUB)$ pourront donc se démontrer par induction.
- les propriétés prouvées pour les éléments de $\Phi(SUB)$ peuvent être déduites pour les éléments de SUB.

Les propositions suivantes expriment des propriétés sur les substitutions généralisées image par Φ de compositions d'étiquettes

2.3.2 Propriétés

Proposition 2.1. *Toute substitution généralisée image par Φ d'une composition d'étiquettes peut s'écrire sous la forme $sub_1 \ [] \ ... \ [] \ sub_m$ où chaque sub_i est de la forme $g_1 \Rightarrow e_1; g_2 \Rightarrow e_2; ...; g_n \Rightarrow e_n$ ou $g \Rightarrow skip$. Cette forme est appelée la forme canonique de la substitution généralisée.*

Proof. La preuve se fait par induction structurelle sur la substitution généralisée.

- Pour les cas de base :
  - si la substitution est *skip*, on a le résultat suivant
    $skip \equiv true \Rightarrow skip$
  - si la substitution est e, on a le résultat suivant
    $e \equiv true \Rightarrow e$
- Pour les cas d'induction, on suppose que sub et sub' vérifient l'hypothèse d'induction et l'on distingue les cas suivants :
  - Séquencement de compositions d'étiquettes.

$$\begin{array}{rll} sub; sub' \equiv & [sub_1 \ [] \ ... \ [] \ sub_n]; [sub'_1 \ [] \ ... \ [] \ sub'_m] & \text{(par hypothèse d'induction)} \\ \equiv & sub_1; sub'_1 \ [] \ ... \ [] \ sub_1; sub'_m [] & \text{(application de } R8 \text{ et } R9 \text{ Tab. 3)} \\ & ...[] & \\ & sub_n; sub'_1 \ [] \ ... \ [] \ sub_n; sub'_m & \end{array}$$

    La propriété est vérifiée à cause de la forme des sub_i et sub'_i
  - Choix entre des compositions d'étiquettes.

$$\begin{array}{rll} sub \ [] \ sub' \equiv & [sub_1 \ [] \ ... \ [] \ sub_n] \ [] \ [sub'_1 \ [] \ ... \ [] \ sub'_m] & \text{(par hypothèse d'induction)} \\ \equiv & sub_1 \ [] \ ... \ [] \ sub_n \ [] \ sub'_1 \ [] \ ... \ [] \ sub'_m & \text{(par associativité de } []) \end{array}$$

  - Composition d'étiquettes gardée.

$$\begin{array}{rll} p \Rightarrow sub \equiv & p \Rightarrow [sub_1 \ [] \ ... \ [] \ sub_n] & \text{(par hypothèse d'induction)} \\ \equiv & p \Rightarrow sub_1 \ [] \ ... \ [] \ p \Rightarrow sub_n & \text{(application de } R10 \text{ Tab. 3)} \end{array}$$

    Pour chaque sub_i on a :

$$\begin{array}{rll} p \Rightarrow sub_i \equiv & p \Rightarrow (g_1 \Rightarrow e_1; g_2 \Rightarrow e_2; ...; g_n \Rightarrow e_n) & \\ \equiv & (p \wedge g_1) \Rightarrow e_1; g_2 \Rightarrow e_2; ...; g_n \Rightarrow e_n & \text{(application de } R11 \text{ et } R12 \text{ Tab. 3)} \end{array}$$

  - Itération d'une composition d'étiquettes. Toute composition d'étiquettes $\mathcal{W}(n > 1, sub_1, p, n)$ peut s'écrire sous la forme $\underbrace{sub_1; ...; sub_1}_{n \ fois}$.

    Nous prouvons les résultats attendus par induction, puisque cette substitution est composée de substitutions utilisant la séquence (;), $\Rightarrow$ et *skip*.

□

Le lemme suivant est utilisé pour démontrer la proposition 2.2.

Lemme 2.1. *Toute substitution généralisée sous la forme* $g_1 \Rightarrow e_1; g_2 \Rightarrow e_2; ...; g_n \Rightarrow e_n$ *où* $e_i \widehat{=} P_i|B_i$ *peut s'écrire de la forme* $P|B$ *où :*

- $P \equiv (g_1 \Rightarrow P_1) \wedge [g_1 \Rightarrow B_1](g_2 \Rightarrow P_2) \wedge ... \wedge [g_1 \Rightarrow B_1; ...; g_{n-1} \Rightarrow B_{n-1}](g_n \Rightarrow P_n)$
- $B \equiv g_1 \Rightarrow B_1; ...; g_n \Rightarrow B_n$

Proof. On effectue la démonstration par récurrence sur n

1. Cas de base. Si $n = 1$, on a
   $g_1 \Rightarrow P_1|B_1 \equiv g_1 \Rightarrow P_1|g_1 \Rightarrow B_1$ (application de $R1$ Tab. 3)
   le résultat est donc vrai pour 1.

2. Cas général. Hypothèse de récurrence : on suppose la forme vrai pour $n - 1$

   $$\begin{array}{ll} & g_1 \Rightarrow e_1; g_2 \Rightarrow e_2; ...; g_{n-1} \Rightarrow e_{n-1} \\ \equiv & ((g_1 \Rightarrow P_1) \wedge [g_1 \Rightarrow B_1](g_2 \Rightarrow P_2) \wedge ... \wedge [g_1 \Rightarrow B_1; ...; g_{n-2} \Rightarrow B_{n-2}](g_{n-1} \Rightarrow P_{n-1})) \mid \\ & (g_1 \Rightarrow B_1; ...; g_{n-1} \Rightarrow B_{n-1}) \\ \equiv & P'|B' \end{array}$$

   On démontre pour n :

   $$\begin{array}{lll} & g_1 \Rightarrow e_1; g_2 \Rightarrow e_2; ...; g_{n-1} \Rightarrow e_{n-1}; g_n \Rightarrow e_n & \\ \equiv & (P'|B'); (g_n \Rightarrow P_n|B_n) & \text{(application de l'hypothèse de récurrence)} \\ \equiv & (P'|B'); (g_n \Rightarrow P_n|g_n \Rightarrow B_n) & \text{(application } R1 \text{ Tab. 3)} \\ \equiv & P'|(B'; (g_n \Rightarrow P_n|g_n \Rightarrow B_n)) & \text{(application } R2 \text{ Tab. 3)} \\ \equiv & P'|([B'](g_n \Rightarrow P_n)|(B'; g_n \Rightarrow B_n)) & \text{(application } R3 \text{ Tab. 3)} \\ \equiv & (P' \wedge [B'](g_n \Rightarrow P_n))|(B'; g_n \Rightarrow B_n) & \text{(application } R4 \text{ Tab. 3)} \\ \equiv & P|B \text{ avec } P \text{ et } B \text{ sous la forme attendue} & \end{array}$$

□

Proposition 2.2. *Toute substitution généralisée sous forme canonique peut s'écrire sous la forme* $P|B$ *où* $P \equiv P_1 \wedge ... \wedge P_n$ *et* $B \equiv B_1 \,[]\, ... \,[]\, B_n$.

Proof. Soit $sub_1 \,[]\, sub_2 \,[]\, ... \,[]\, sub_n$ la forme canonique de la substitution. Nous effectuons une preuve par récurrence sur n.

1. Cas de base. Pour $n = 1$,

   - Si $sub_1 = g \Rightarrow skip$, alors $sub_1 = True|g \Rightarrow skip$.
   - Si $sub_1 = g_1 \Rightarrow e_1; g_2 \Rightarrow e_2; ...; g_n \Rightarrow e_n$, alors $sub_1 = P_1|B_1$ (d'après le lemme 2.1).

   $sub_1 = P_1|B_1$ c'est démontré

2. Cas général. Hypothèse de récurrence : on suppose la propriété vraie pour $n - 1$.

□

$$
\begin{array}{lll}
 & sub_1 \,[]\, sub_2 \,[]\, ... \,[]\, sub_n & \\
\equiv & P_1|B_1 \,[]\, ... \,[]\, P_{n-1}|B_{n-1} \,[]\, P_n|B_n & \text{(Lem. 2.1)} \\
\equiv & (P'|B') \,[]\, (P_n|B_n) & \text{(Hypothèse de récurrence)} \\
\equiv & P'|(B' \,[]\, (P_n|B_n)) & (R5 \text{ Tab. 3}) \\
\equiv & P'|(P_n|(B' \,[]\, B_n)) & (R6 \text{ Tab. 3}) \\
\equiv & P' \wedge P_n|(B' \,[]\, B_n) & (R4 \text{ Tab. 3}) \\
 & \text{est bien de la forme attendue} &
\end{array}
$$

3 Relations entre STEs et raffinement B

Dans ce paragraphe nous mettons en évidence des relations entre STEs. Pour établir ces relations, nous utilisons des constructions B qui utilisent la traduction des STEs en B et les clauses **REFINES** et **INCLUDES** de B. La correspondance entre STEs se fait par une mise en relation des états et des étiquettes. Nous considérons successivement les cas suivants :

- raffinement B faisant intervenir des STEs sur un même alphabet et une relation entre les états.
- raffinement B faisant intervenir deux STEs sur des alphabets différents, une relation entre les états et une relation entre les étiquettes.
- raffinement B faisant intervenir un STE et plusieurs STEs, une relation entre les états et une relation entre les étiquettes.
- raffinement B entre deux STEs gardés.

Pour chacun de ces cas, nous établissons une proposition.

3.1 Relation entre deux STEs sur un même alphabet

Nous commençons par présenter la construction B utilisée. Nous caractérisons, ensuite, le raffinement B entre deux STEs par la proposition 3.1 et nous comparons cette caractérisation aux relations de simulation introduites par Milner [Mil89], van Glabbeek et Bloom [BIM95].

3.1.1 Construction B

Nous considérons deux STEs $T_1 = (A,\ S_1,\ s_0^1,\ F1,\ \rightarrow_1)$ et $T_2 = (A,\ S_2,\ s_0^2,,\ F2,\ \rightarrow_2)$ dont les transitions sont étiquetées par les éléments d'un même alphabet A et une relation R de S_1 vers S_2. Nous construisons les spécifications B, M_a et M_c associées respectivement à T_1 et T_2. M_a est une machine abstraite B et M_c est un raffinement (au sens de B) de M_a (Fig. 4). M_a et M_c sont obtenues par traductions respectives de T_1 et T_2, en appliquant les règles de traductions définies dans 2.1. Les variables $stateT_1$ et $stateT_2$ modélisant les états des STEs apparaissant respectivement dans T_1 et T_2, la relation R entre les états des deux STEs est définie dans M_c, de même que l'assertion $(stateT_1 \mapsto stateT_2) \in R$ qui est l'invariant de collage.

Dans la suite nous dirons que T_1 est le STE abstrait (représenté par la machine abstraite B, M_a), que T_2 est le STE concret (représenté par le raffinement B, M_c) et que T_2 raffine T_1 si et seulement si M_c est un raffinement de M_a.

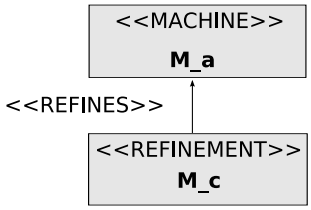


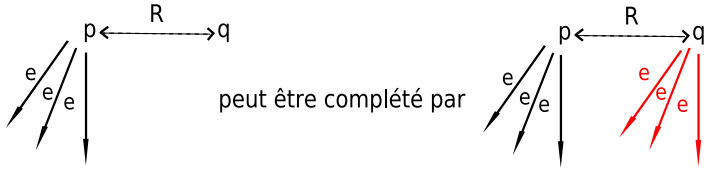
Figure 4: Construction B

3.1.2 Raffinement B

La proposition suivante met en évidence des propriétés nécessaires au raffinement d'un STE par un autre STE.

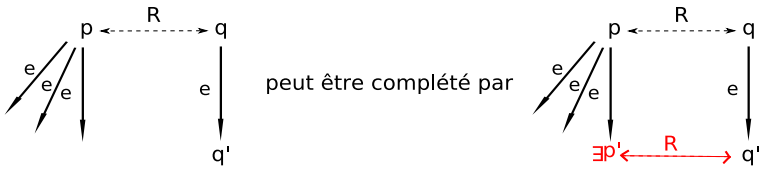
Proposition 3.1. *Si T_2 raffine T_1 par rapport à R, alors on a les trois propriétés suivantes :*

1. $(s_0^1, s_0^2) \in R$
2. $(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ [(p,q) \in R\ \wedge\ p \xrightarrow{e}_1 \Rightarrow q \xrightarrow{e}_2]$



3. $(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ (\forall q' \in S_2)$

$$[(p,q) \in R \wedge q \xrightarrow{e}_2 q' \wedge p \xrightarrow{e}_1 \Rightarrow (\exists p' \in S_1)\ [p \xrightarrow{e}_1 p' \wedge (p',q') \in R]]$$



Proof. La preuve d'un raffinement en B consiste principalement à démontrer des obligations de preuve pour l'initialisation et pour chaque opération apparaissant dans M_a et M_c. Soient e une opération et $P_a|B_a$, $P_c|B_c$ les définitions respectives de e dans M_a et M_c. $I_a \equiv stateT_1 \in S_1$, $I_c \equiv stateT_2 \in S_2 \wedge (stateT_1 \mapsto stateT_2) \in R$ sont les invariants respectifs de M_a et M_c.
L'obligation de preuve relative à l'initialisation est la formule $[Init_c]\neg[Init_a]\neg I_c$. L'obligation de preuve relative à l'opération e est la formule $I_a \wedge I_c \wedge P_a \Rightarrow P_c \wedge [B_c]\neg[B_a]\neg I_c$ qui peut être décomposée en les deux formules $I_a \wedge I_c \wedge P_a \Rightarrow P_c$ et $I_a \wedge I_c \wedge P_a \Rightarrow [B_c]\neg[B_a]\neg I_c$. Remarquons que l'invariant I_a joue simplement un rôle de typage pour la variable $stateT_1$, il en est de même de la partie $stateT_2 \in S_2$ de l'invariant I_c, ils peuvent donc être ignorés dans les obligations de preuves, nous allons donc considérer les trois obligations de preuve suivantes :

1. $\phi_1 \equiv [Init_c]\neg[Init_a]\neg(stateT_1 \mapsto stateT_2) \in R$
2. $\phi_2 \equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow P_c$
3. $\phi_3 \equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]\neg[B_a]\neg(stateT_1 \mapsto stateT_2) \in R.$

Nous allons montrer qu'à chaque formule ϕ_1, ϕ_2, ϕ_3 correspondent les conditions associées aux trois items (1), (2), (3) de la proposition.

1. ϕ_1, on a $Init_c \equiv stateT_2 := s_0^2$ et $Init_a \equiv stateT_1 := s_0^1$ d'où
   $\phi_1 \equiv [stateT_2 := s_0^2]\neg[stateT_1 := s_0^1]\neg(stateT_1 \mapsto stateT_2) \in R$
   $\equiv (s_0^1 \mapsto s_0^2) \in R$
2. ϕ_2, on a $P_a \equiv stateT_1 \in pre(e, T_1)$ et $P_c \equiv stateT_2 \in pre(e, T_2)$. ϕ_2 s'écrit donc

   $$\phi_2 \equiv (stateT_1 \mapsto stateT_2) \in R \wedge stateT_1 \in pre(e, T_1) \Rightarrow stateT_2 \in pre(e, T_2)$$

   Dans ϕ_2 $stateT_1$ et $stateT_2$ sont des variables libres qui peuvent être interprétées comme des variables universellement quantifiées, en les renommant respectivement en p et q et en utilisant le fait que $stateT_1 \in pre(e, T_1) \equiv stateT_1 \xrightarrow{e}_1$ et $stateT_2 \in pre(e, T_2) \equiv stateT_2 \xrightarrow{e}_2$, on obtient l'item (2) de la proposition.

   $$(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ [(p, q) \in R\ \wedge\ p \in pre_{T_1}(e)] \Rightarrow q \in pre_{T_2}(e)$$

3. ϕ_3, les définitions de B_a et B_c sont :
   - $B_a \equiv a_1 \Rightarrow b_1 \,[]\, ... \,[]\, a_n \Rightarrow b_n$
     $\equiv stateT_1 = p_1 \Rightarrow stateT_1 := p'_1 \,[]\, ... \,[]\, stateT_1 = p_n \Rightarrow stateT_1 := p'_n$
     avec $\xrightarrow{e}_1 = \{(p_1, p'_1), ..., (p_n, p'_n)\}$
   - $B_c \equiv c_1 \Rightarrow d_1 \,[]\, ... \,[]\, c_m \Rightarrow d_m$
     $\equiv stateT_2 = q_1 \Rightarrow stateT_2 := q'_1 \,[]\, ... \,[]\, stateT_2 = q_m \Rightarrow stateT_2 := q'_m$
     avec $\xrightarrow{e}_2 = \{(q_1, q'_1), ..., (q_m, q'_m)\}$

   En remplaçant B_a, ϕ_3 s'écrit :

   $\phi_3 \equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]$
   $\neg[a_1 \Rightarrow b_1 \,[]\, ... \,[]\, a_n \Rightarrow b_n]\neg(stateT_1 \mapsto stateT_2) \in R]$
   $\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]$ $\quad (\mathcal{WP}\ 5)$
   $\neg([a_1 \Rightarrow b_1]\neg(stateT_1 \mapsto stateT_2) \in R \wedge$
   $\ldots \wedge$
   $[a_n \Rightarrow b_n]\neg(stateT_1 \mapsto stateT_2) \in R)]$

$\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]$ ($\mathcal{WP}$ 6)
$\neg(a_1 \Rightarrow [b_1]\neg(stateT_1 \mapsto stateT_2) \in R \wedge$
$...\wedge$
$(a_n \Rightarrow [b_n])\neg(stateT_1 \mapsto stateT_2) \in R)]$
$\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]$ (Introduire $\neg$)
$a_1 \wedge \neg([b_1]\neg(stateT_1 \mapsto stateT_2) \in R) \vee$
$...\vee$
$a_n \wedge \neg([b_n]\neg(stateT_1 \mapsto stateT_2) \in R)]$
$\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]$
$\bigvee_{j=1}^{n}(a_j \wedge \neg([b_j]\neg(stateT_1 \mapsto stateT_2) \in R))$
$\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]$
$\bigvee_{j=1}^{n}(stateT_1 = p_j \wedge \neg([stateT_1 := p'_j]\neg(stateT_1 \mapsto stateT_2) \in R))$
$\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]$ ($\mathcal{WP}$ 1)
$\bigvee_{j=1}^{n}(stateT_1 = p_j \wedge (p'_j \mapsto stateT_2) \in R))$
$\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]$
$(\exists (p, p') \in \xrightarrow{e}_1)[stateT_1 = p \wedge (p' \mapsto stateT_2) \in R]]$
$\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]$
$(\exists p \in S_1)\ (\exists\ p' \in S_1)\ [stateT_1 = p \wedge p \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]$
$\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B_c]$ ($\{q \backslash StateT_1\}$)
$(\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]$

En remplaçant B_c par sa définition, on obtient :

$\phi_3 \equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow$
$[c_1 \Rightarrow d_1 \,[]\, ... \,[]\, c_m \Rightarrow d_m]$
$(\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]$
$\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow$ ($\mathcal{WP}$ 5)
$[c_1 \Rightarrow d_1](\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R] \wedge$
$...\wedge$
$[c_m \Rightarrow d_m](\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]$
$\equiv \bigwedge_{i=1}^{m}((stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [c_i \Rightarrow d_i]$
$(\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]$
$\equiv \bigwedge_{i=1}^{m}((stateT_1 \mapsto stateT_2) \in R \wedge P_a \wedge c_i \Rightarrow [d_i]$ ($\mathcal{WP}$ 6)
$(\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]$
$\equiv (\forall i \in [1..m])[(stateT_1 \mapsto stateT_2) \in R \wedge P_a \wedge c_i \Rightarrow [d_i]$
$(\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]$
$\equiv (\forall i \in [1..m])[(stateT_1 \mapsto stateT_2) \in R \wedge P_a \wedge stateT_2 = q_i \Rightarrow$
$[stateT_2 := q'_i]$
$(\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]$

$\equiv(\forall i \in [1..m])[(stateT_1 \mapsto stateT_2) \in R \wedge P_a \wedge stateT_2 = q_i \Rightarrow \quad (\mathcal{WP}\ 1)$
$(\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q'_i) \in R]$
$\equiv(\forall (q, q') \in \xrightarrow{e}_2)\ [(stateT_1 \mapsto stateT_2) \in R \wedge P_a \wedge stateT_2 = q \Rightarrow$
$(\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q') \in R]$
$\equiv(\forall q \in S_2)\ (\forall q' \in S_2)\ [(stateT_1 \mapsto stateT_2) \in R \wedge P_a \wedge$
$stateT_2 = q \wedge q \xrightarrow{e}_2 q' \Rightarrow$
$(\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q') \in R]$
$\equiv(\forall q' \in S_2)\ [(stateT_1 \mapsto stateT_2) \in R \wedge P_a \wedge stateT_2 \xrightarrow{e}_2 q' \Rightarrow \quad (\{q \backslash StateT_2\})$
$(\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q') \in R]$

On démontre, ainsi, le troisième item de la proposition.

□

Exemple 3.1. La figure 5 montre un exemple de raffinement d'une étiquette *ee* d'un STE T_1 par un STE T_2, les états de T_1 sont dénotés p_i alors que ceux de T_2 sont dénotés q_i, la relation R est représentée partiellement par des doubles flèches pointillées reliant les états des deux STEs. Le texte des spécifications B est dans l'annexe B. Pour prouver le raffinement 16 POs ont été générées dont 6 ont été prouvées de manière interactive, une partie des POs sont dans l'annexe B.

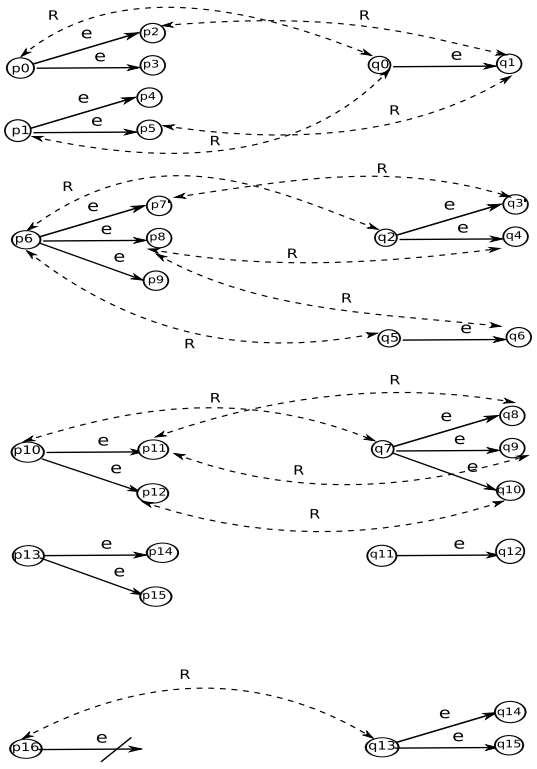
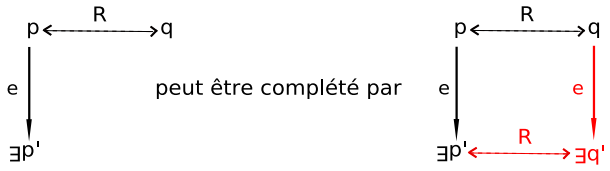


Figure 5: Raffinement d'une étiquette dans deux STEs

Corollaire 3.1. *Si T_2 raffine T_1 par rapport à R, alors on peut déduire la propriété suivante :*
$(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ (\exists p' \in S_1)$

$$[((p,q) \in R\ \wedge\ p \xrightarrow{e}_1 p') \Rightarrow (\exists q' \in S_2)\ [(p',q') \in R\ \wedge\ q \xrightarrow{e}_2 q']]$$



Le corollaire 3.1 est une autre façon d'exprimer la proposition 3.1.

Proof. Distinguons deux cas :

- Si non $p \xrightarrow{e}_1$ la partie gauche de l'implication est fausse, l'implication est donc vraie, la formule est trivialement vraie.
- si $p \xrightarrow{e}_1$, soient $e \in A$, $p \in S_1$ et $q \in S_2$ quelconque tel que $(p,q) \in R$, d'après l'item (2) de la proposition 3.1
  $(p,q) \in R \wedge p \xrightarrow{e}_1 \Rightarrow q \xrightarrow{e}_2$, c'est-à-dire qu'il existe $q' \in S_2$ tel que $q \xrightarrow{e}_2 q'$
  d'après l'item (3) de la proposition 3.1 on a
  $[(p,q) \in R \wedge q \xrightarrow{e}_2 q' \wedge p \xrightarrow{e}_1 \Rightarrow (\exists p' \in S_1)\ [p \xrightarrow{e}_1 p' \wedge (p',q') \in R]]$
  c'est-à-dire qu'il existe $p' \in S_1$ tel que $p \xrightarrow{e}_1 p' \wedge (p',q') \in R$ en définitive, il existe $p' \in S_1$ et $q' \in S_2$ vérifiant $p \xrightarrow{e}_1 p'$, $(p',q') \in R$ et $q \xrightarrow{e}_2 q'$ ce qui démontre le corollaire.

□

3.1.3 Relation entre raffinement B et simulation

Maintenant nous allons présenter certaines cas dans lesquelles le raffinement B entre deux STEs implique une relation de simulation. Nous commençons par introduire ce premier corollaire, dans lequel nous prenons comme hypothèse que le STE abstrait est déterministe.

Corollaire 3.2. *Si T_2 raffine T_1 par rapport à R tel si T_1 est déterministe alors $T_2 \preccurlyeq_R T_1$.*

Proof. La formule du corollaire 3.1 est :
$(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ (\exists p' \in S_1)$

$$[((p,q) \in R\ \wedge\ p \xrightarrow{e}_1 p') \Rightarrow (\exists q' \in S_2)\ [(p',q') \in R\ \wedge\ q \xrightarrow{e}_2 q']]$$

Comme T_1 est déterministe, pour tout état p de S_1 et pour tout e de A il existe au plus une transition de la forme $p \xrightarrow{e}_1 p'$. La quantification existentielle sur la variable p' peut être remplacée par une quantification universelle, on a donc
$(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ (\forall p' \in S_1)$

$$[((p,q) \in R\ \wedge\ p \xrightarrow{e}_1 p') \Rightarrow (\exists q' \in S_2)\ [(p',q') \in R\ \wedge\ q \xrightarrow{e}_2 q']]$$

comme de plus $(s_0^1, s_0^2) \in R$, on a bien $T_2 \preccurlyeq_R T_1$. □

Dans le but de comparer la relation de raffinement entre deux STEs et les relations de ready-simulation et de bisimulation, nous introduisons la propriété suivante J concernant la relation R considérée :

$$(a) \quad J : (\forall e \in A)\ (p \in S_2)\ (q \in S_1)\ [(p,q) \in R \wedge q \xrightarrow{e}_2 \Rightarrow p \xrightarrow{e}_1]$$

D'un point de vue pratique pour tenir compte en B de la propriété (a) il suffit d'ajouter dans la clause **ASSERTIONS** le prédicat défini par $P_c \Rightarrow P_a$ (P_a et P_c étant les préconditions respectives de l'opération e dans les machines B, M_a et M_c), l'ajout de ce prédicat génère l'obligation de preuve suivante

$$I_a \wedge I_c \Rightarrow (P_c \Rightarrow P_a)$$

que l'on peut réduire à la forme (a).

La proposition suivante établit la relation entre raffinement B et simulation.

Proposition 3.2. *Soient T_1 et T_2 deux STEs et R une relation.*

- *Si T_2 raffine T_1 et si R vérifie J alors $T_1 \precsim T_2$.*
- *Si de plus T_1 est déterministe alors $T_1 \approx_R T_2$.*

Proof.

1. Comme T_2 raffine T_1, le trois item de la proposition 3.1 s'écrit :
   $(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ (\forall q' \in S_2)$

   $$[(p,q) \in R \wedge q \xrightarrow{e}_2 q' \wedge\ p \xrightarrow{e}_1 \Rightarrow (\exists p' \in S_1)\ [p \xrightarrow{e}_1 p' \wedge (p',q') \in R]]$$

   J s'écrit :

   $$(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ [(p,q) \in R\ \wedge\ q \xrightarrow{e}_2 \Rightarrow p \xrightarrow{e}_1]$$

   De ces deux formules on déduit
   $(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ (\forall q' \in S_2)$

   $$[(p,q) \in R \wedge q \xrightarrow{e}_2 q' \Rightarrow (\exists p' \in S_1)\ [p \xrightarrow{e}_1 p' \wedge (p',q') \in R]]$$

   qui est équivalent à
   $(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ (\forall q' \in S_2)$

   $$[(q,p) \in R^{-1} \wedge q \xrightarrow{e}_2 q' \Rightarrow (\exists p' \in S_1)\ [p \xrightarrow{e}_1 p' \wedge (p',q') \in R^{-1}]]$$

   de plus d'après le premier item de la proposition 3.1 qui peut s'écrire $(s_0^2, s_0^1) \in R^{-1}$. On a donc bien $T_1 \preccurlyeq_{R^{-1}} T_2$.
   Le second item de la proposition 3.1 s'exprime par

$$(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ [(p,q) \in R\ \wedge\ (\exists p' \in S_1)\ p \xrightarrow{e}_1 p' \Rightarrow\ (\exists q' \in S_2)\ q \xrightarrow{e}_2 q']$$

ou

$$(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ [(q,p) \in R^{-1}\ \wedge\ (\exists p' \in S_1)\ p \xrightarrow{e}_1 p' \Rightarrow\ (\exists q' \in S_2)\ q \xrightarrow{e}_2 q']$$

Cette formule et la condition $T_1 \preccurlyeq_{R^{-1}} T_2$ entraîne que $T_1 \precsim_{R^{-1}} T_2$.

2. Si T_1 est déterministe d'après le corollaire $T_2 \preccurlyeq_R T_1$. Comme de plus $T_1 \precsim_{R^{-1}} T_2$ on a bien $T_2 \approx_R T_1$.

□

Dans la traduction des STEs en B (Sec. 2.1 p. 7), pour chaque étiquette nous avons défini une opération avec des préconditions, une alternative à cette traduction est de ne pas considérer les préconditions pour les opérations (cela revient à avoir des conditions vraies). Sous cette hypothèse nous avons la proposition suivante.

Proposition 3.3. *Si T_2 raffine T_1 par rapport à R et si dans la traduction des STEs les opérations associées aux étiquettes n'ont pas de préconditions alors $T_1 \preccurlyeq_{R^{-1}} T_2$.*

Proof. Considérons les formules ϕ_1, ϕ_2 et ϕ_3 de la preuve de la proposition 3.1. ϕ_1 est inchangée et peut s'exprimer par $(s_0^2, s_0^1) \in R^{-1}$, ϕ_2 est trivialement vraie, ϕ_3 peut se simplifier (en tenant compte que $P_a \equiv vrai$ en
$(\forall e \in A)\ (\forall p \in S_1)\ (\forall q \in S_2)\ (\forall q' \in S_2)$

$$[(p,q) \in R \wedge q \xrightarrow{e}_2 q' \Rightarrow (\exists p' \in S_1)\ [p \xrightarrow{e}_1 p' \wedge (p',q') \in R]]$$

Cette formule avec la condition $(s_0^2, s_0^1) \in R^{-1}$ exprime que $T_1 \preccurlyeq_{R^{-1}} T_2$. □

3.2 Relation entre deux STEs avec des étiquettes différentes

Nous présentons d'abord la construction B utilisée qui fait intervenir deux STEs, ensuite nous montrons les propriétés du raffinement en terme de relation entre les deux STEs.

3.2.1 Construction B

Nous considérons deux STEs $T_1 = (A_1,\ S_1,\ s_0^1,\ F1,\ \rightarrow_1)$ et $T_2 = (A_2,\ S_2,\ s_0^2,,\ F2,\ \rightarrow_2)$ tels que $A_1 \cap A_2 = \emptyset$, une relation R de S_1 vers S_2 et une application χ de A_1 vers $\mathcal{E}(A_2)$. Nous construisons les spécifications B, M_a et M_c traductions respectives de T_1 et T_2. Nous construisons, ensuite, le raffinement M_c' qui raffine M_a et inclut M_c. M_c' est obtenue par traduction de R et de χ. M_c' comporte toutes les opérations de M_a et chaque opération est définie par une composition d'étiquettes grâce à l'application χ. L'invariant de M_c' est un invariant de collage qui lie les variables de M_a et M_c. Il est de la forme :$(stateT_1 \mapsto stateT_2) \in R$. La construction B est donnée par la figure 6. Si le raffinement B, M_c' est démontré correct, on dit que T_2 raffine T_1 par rapport à R et χ.

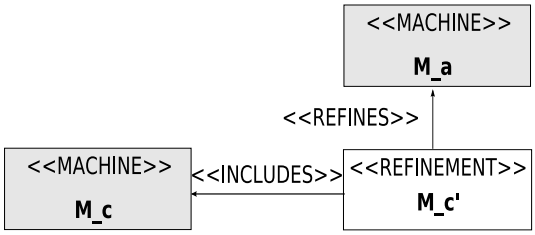


Figure 6: B

3.2.2 Raffinement B

On introduit les deux lemmes suivants qui vont servir à la preuve de la proposition 3.4.

Lemme 3.1. *Soit la substitution* $sub = g_1 \Rightarrow e_1; ...; g_n \Rightarrow e_n$ *où* $e_i \widehat{=} P_i|B_i$ *et* $B_i = stateT = q_{i1} \Rightarrow stateT =: q'_{i1} \,[]\, ... \,[]\, stateT = q_{ik_i} \Rightarrow stateT =: q'_{ik_i}$ *et soit le prédicat* R*, alors* sub *peut s'écrire sous la forme* $P|B$ *et le prédicat* $[B]R$ *est donné par la formule suivante :*

$$\begin{aligned}(\forall q_2 \in S)...(\forall q_{n+1} \in S)\ & [(g_1(stateT) \wedge g_2(q_2) \wedge .... \wedge g_n(q_n)) \wedge \\ & stateT \xrightarrow{e_1} q_2 \wedge ... \wedge q_n \xrightarrow{e_n} q_{n+1} \\ & \Rightarrow [stateT := q_{n+1}]R]\end{aligned}$$

Proof. D'après le lemme 2.1 *sub* peut s'écrire sous la forme $P|B$ avec $B \equiv g_1 \Rightarrow B_1; ...; g_n \Rightarrow B_n$. Démontrons la deuxième partie du lemme par récurrence sur n.

1. Cas de base $n = 1$, on a $[B]R \equiv [g_1 \Rightarrow B_1]R$ et on pose $e_1 = e$.
   on remplace B_1 par $stateT = q_1 \Rightarrow stateT := q'_1 \,[]\, ... \,[]\, stateT = q_k \Rightarrow stateT := q'_k$. Cette

   $$\begin{aligned}
   & [g_1 \Rightarrow (stateT = q_1 \Rightarrow \\
   & stateT =: q'_1 \,[]\, ... \,[]\, stateT = q_k \Rightarrow stateT =: q'_k)]R \\
   \equiv & (g_1(stateT) \wedge stateT = q_1 \Rightarrow [stateT := q'_1]R)[] && (R12 \text{ Tab. 3}) \\
   & ...[] \\
   & (g_1(stateT) \wedge stateT = q_k \Rightarrow [stateT = q'_k]R) \\
   \equiv & (g_1(stateT) \wedge stateT = q_1 \Rightarrow [stateT := q'_1]R) \wedge && (\mathcal{WP}5 \text{ Tab. 2 p. 30}) \\
   & ... \wedge \\
   & (g_1(stateT) \wedge stateT = q_k \Rightarrow [stateT = q'_k]R) \\
   \equiv & (\forall (q, q') \in \xrightarrow{e})[g_1(stateT) \wedge stateT = q \Rightarrow [stateT := q']R] \\
   \equiv & (\forall q \in S)\ (\forall q' \in S)[g_1(stateT) \wedge stateT = q \wedge q \xrightarrow{e} q' \\
   & \Rightarrow [stateT := q']R] \\
   \equiv & (\forall q' \in S)[g_1(stateT) \wedge stateT \xrightarrow{e} q' \Rightarrow [stateT := q']R] && (\{q \backslash StateT\})
   \end{aligned}$$

   formule établit le résultat pour $n = 1$

2. Cas général. Hypothèse de récurrence : on suppose que le résultat est vrai pour tout $k <$

$$
\begin{array}{ll}
 & \equiv(\forall q_2 \in S)...(\forall q_n \in S) \qquad (R13 \text{ et } \mathcal{WP}1\ ) \\
 & \quad (g_1(stateT) \wedge g_2(q_2) \wedge .... \wedge g_{n-1}(q_{n-1})) \wedge \\
 & \quad stateT \stackrel{e_1}{\rightarrow} q_2 \wedge ... \wedge q_{n-1} \stackrel{e_{n-1}}{\rightarrow} q_n \wedge \\
n. & \quad ((\forall q_{n+1} \in S)[g_1(q_n) \wedge q_n \stackrel{e_n}{\rightarrow} q_{n+1} \Rightarrow [stateT := q_{n+1}]R]) \\
 & \equiv(\forall q_2 \in S)...(\forall q_{n+1} \in S)[(g_1(stateT) \wedge g_2(q_2) \wedge .... \wedge g_n(q_n)) \wedge \\
 & \quad stateT \stackrel{e_1}{\rightarrow} q_2 \wedge ... \wedge q_n \stackrel{e_n}{\rightarrow} q_{n+1} \Rightarrow [stateT := q_{n+1}]R]
\end{array}
$$

$$
\begin{array}{ll}
 & [g_1 \Rightarrow B_1; ...; g_{n-1} \Rightarrow B_{n-1}; g_n \Rightarrow B_n]R \\
\equiv & [g_1 \Rightarrow B_1; ...; g_{n-1} \Rightarrow B_{n-1}]([g_n \Rightarrow B_n]R) \qquad (\mathcal{WP}8 \text{ Tab. 2 p. 30}) \\
\equiv & [g_1 \Rightarrow B_1; ...; g_{n-1} \Rightarrow B_{n-1}]\ (\forall q_{n+1} \in S) \qquad (\text{Hyp. R. pour } n = 1) \\
 & [g_1(stateT) \wedge stateT \stackrel{e_n}{\rightarrow} q_{n+1} \Rightarrow [stateT := q_{n+1}]R] \\
\equiv & (\forall q_2 \in S)...(\forall q_n \in S)[g_1(stateT) \wedge g_2(q_2) \wedge .... \wedge g_{n-1}(q_{n-1}) \wedge \qquad (\text{Hyp. R. pour } n-1) \\
 & stateT \stackrel{e_1}{\rightarrow} q_2 \wedge ... \wedge q_{n-1} \stackrel{e_{n-1}}{\rightarrow} q_n \Rightarrow [stateT := q_n]] \\
 & ((\forall q_{n+1} \in S)[g_1(stateT) \wedge stateT \stackrel{e_n}{\rightarrow} q_{n+1} \Rightarrow [stateT := q_{n+1}]R])
\end{array}
$$

Cette formule exprime la propriété au rang n et démontre ainsi le lemme.

□

Lemme 3.2. *Soit sub une composition d'étiquette sur un STE T. La substitution* $\Phi(sub)$ *est composée des appels d'opérations* $e_i \widehat{=} P_i|B_i$ *avec* $B_i = stateT = q_{i1} \Rightarrow stateT =: q'_{i1} [] ... [] stateT = q_{ik_i} \Rightarrow stateT =: q'_{ik_i}$ *et* $P_i = stateT \in pre(e_i, T)$*. La précondition P de* $\Phi(sub)$ *est équivalente à* $P \equiv stateT \in pre(sub, T)$

Proof. On considère trois cas :

1. Si $\Phi(sub) = g \Rightarrow skip$ alors $P = True$ c'est-à-dire on a $stateT \in S$

2. Si $sub = g_1 \Rightarrow e_1; ...; g_n \Rightarrow e_n$, alors on a :

$$
\begin{array}{ll}
P \equiv & \bigwedge_{j=1}^{n}([g_1 \Rightarrow B_1; ...; g_{j-1} \Rightarrow B_{j-1}](g_j \Rightarrow P_j)) \\
\equiv & \bigwedge_{j=1}^{n}((\forall q_1 \in S)...(\forall q_j \in S) \qquad (\text{Lem. 2.1}) \\
 & \qquad [(g_1(stateT) \wedge g_2(q_2) \wedge .... \wedge g_{j-1}(q_{j-1})) \wedge \\
 & \qquad stateT = q_1 \wedge q_1 \stackrel{e_1}{\rightarrow} q_2 \wedge ... \wedge q_{j-1} \stackrel{ej-1}{\rightarrow} q_j \\
 & \qquad \Rightarrow [stateT := q_j](g_j \Rightarrow P_j)]) \\
\equiv & \bigwedge_{j=1}^{n}((\forall q_2 \in S)...(\forall q_j \in S) \qquad (\mathcal{WP}\ 1) \\
 & \qquad [(g_1(stateT) \wedge g_2(q_2) \wedge .... \wedge g_{j-1}(q_{j-1})) \wedge \\
 & \qquad stateT \stackrel{e_1}{\rightarrow} q_2 \wedge ... \wedge q_{j-1} \stackrel{ej-1}{\rightarrow} q_j \\
 & \qquad \Rightarrow (g_j(q_j) \Rightarrow q_j \stackrel{e_j}{\rightarrow})] \\
\equiv & stateT \in pre(sub, T) \qquad (\text{Def. } 1.13)
\end{array}
$$

3. La précondition d'une substitution $\Phi(sub)$, tel que sa forme canonique est $\Phi(sub_1)$ [] ... [] $\Phi(sub_m)$, est définie par :

□

$$\begin{aligned} P = \bigwedge_{i=1}^{m} P_i &\equiv \bigwedge_{i=1}^{m} stateT \in pre(sub_i, T) \\ &\equiv stateT \in \bigcap_{i=1}^{m} (pre(sub_i, T)) \\ &\equiv stateT \in pre(sub, T) \qquad \text{(Def. 1.13)} \end{aligned}$$

La proposition suivante met en évidence des propriétés nécessaires de deux STEs pour que le raffinement B correspondant soit satisfait.

Proposition 3.4. *Soient T_1 et T_2 deux STEs, R une relation S_1 vers S_2 et χ une application de A_1 vers $\mathcal{E}(A_2)$, si T_2 raffine T_1 par rapport à R et χ, on a les propriétés suivantes :*

1. $(s_0^1, s_0^2) \in R$
2. $(\forall p \in S_1)\ (\forall q \in S_2)\ (\forall e \in A_1)\ (\forall sub \in \mathcal{E}(A_2))$

$$[sub = \chi(e) \wedge (p, q) \in R\ \wedge p \xrightarrow{e}_1 \Rightarrow q \in pre(sub, T_2)]$$

3. $(\forall p \in S_1)\ (\forall q \in S_2)\ (\forall q' \in S_2)\ \ (\forall e \in A_1)\ (\forall sub \in \mathcal{E}(A_2))\ (\forall c \in Chemin(sub, T_2))$
   $[sub = \chi(e) \wedge (p, q) \in R\ \wedge origin(c) = q \wedge target(c) = q' \wedge p \xrightarrow{e}_1 \Rightarrow$

$$(\exists p' \in S_1)\ [p \xrightarrow{e}_1 p' \wedge (p', q') \in R]]$$

Proof. La preuve consiste à démontrer les obligations de preuve pour l'initialisation et pour chaque opération e apparaissant dans M_a et M'_c.
Soient $I_a \equiv stateT_1 \in S_1$, $I_c \equiv stateT_2 \in S_2$ et $I'_c \equiv (stateT_1 \mapsto stateT_2) \in R$ les invariants respectifs de M_a, M_c et M'_c. Soient $Init_a$, $Init_c$ et $Init'_c$ respectivement les initialisations de M_a, M_c et M'_c. Soient $P_a|B_a$ et $P'_c|B'_c$ les définitions respectives de l'opération e dans M_a et M'_c.

Les obligations de preuve relative au raffinement de l'initialisation et de l'opération e sont les formules ϕ'_1 et ϕ'_2 suivantes :

- $\phi'_1 \equiv [Init_c; Init'_c]\neg[Init_a]\neg I'_c$
- $\phi'_2 \equiv I_a \wedge I_c \wedge I'_c \wedge P_a \Rightarrow P'_c \wedge [B'_c]\neg[B_a]\neg I'_c$

Dans ϕ'_1 la substitution $Init'_c$ est définie par *skip*, alors on obtient une formule équivalente à la formule ϕ_1 de la démonstration de la proposition 3.1. Ce qui démontre le premier item de la proposition.

Analysons la deuxième formule ϕ'_2, nous avons $P'_c \equiv True$, I_a et I_c sont des invariants de typage, ϕ'_2 se réduit à
$\phi'_2 \equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B'_c]\neg[B_a]\neg((stateT_1 \mapsto stateT_2) \in R)$
Rappelons, on a :

- $B_a \equiv a_1 \Rightarrow b_1\ [] \ ...\ []\ a_n \Rightarrow b_n$
  $\equiv stateT_1 = p_1 \Rightarrow stateT_1 := p'_1\ [] \ ...\ []\ stateT_1 = p_n \Rightarrow stateT_1 := p'_n$
- $B'_c = \Phi(sub)$ s'écrit sous la forme canonique $\Phi(sub_1)\ [] \ ...\ []\ \Phi(sub_n)$

ϕ'_2 a la même forme que la formule ϕ_3 dans la démonstration de la proposition 3.1, la seule différence est que B_c a été remplacé par B'_c. En remplaçant B_a par sa définition, on a
$\phi'_2 \equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B'_c]\neg[a_1 \Rightarrow b_1 \,[]\, ... \,[]\, a_n \Rightarrow b_n]\neg((stateT_1 \mapsto stateT_2) \in R)$
et en appliquant les mêmes transformations qu'à la formule ϕ_3 de la démonstration de la proposition 3.1 nous obtenons.
$\phi'_2 \equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [B'_c]((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R])$
Nous allons montrer les items (2) et (3) de la proposition en faisant une démonstration par induction structurelle sur la substitution B'_c.

1. cas de base : on a deux cas $B'_c = g \Rightarrow skip$ et $B'_c = g_1 \Rightarrow e_1; g_2 \Rightarrow e_2; ...; g_n \Rightarrow e_n$.

   - Si $B'_c = \Phi(sub) = g \Rightarrow skip$, alors

$$\begin{aligned}
\phi'_2 \equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [g \Rightarrow skip] \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]) \\
\equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \wedge g(stateT_2) \Rightarrow \qquad (\mathcal{WP}\text{ 3 Tab. 2}) \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R])
\end{aligned}$$

     En remplaçant $stateT_1$ par p et $stateT_2$ par q et en quantifiant universellement p et q, nous obtenons :
     $\phi'_2 \equiv (\forall p \in S_1)\ (\forall q \in S_2)\ [(p,q) \in R \wedge p \xrightarrow{e}_1 \wedge\, g(q) \Rightarrow (\exists p' \in S_1)\ [p \xrightarrow{e}_1 p' \wedge (p',q) \in R]]$
     Comme $B'_c = g \Rightarrow skip$, $pre(B'_c, T_2) = S_2$ alors le deuxième item de la proposition est trivialement vérifié. De plus, l'ensemble des chemins de $Chemin(sub, T_2)$ sont les chemins d'origine et d'extrémité q tel que $g(q) = True$. Ceci entraîne : $\phi'_2 \equiv (\forall p \in S_1)\ (\forall q \in S_2)\ (\forall c \in Chemin(sub, T_2))$
     $[(p,q) \in R \wedge p \xrightarrow{e}_1 \wedge\, origine(c) = q \wedge extrimite(c) = q \Rightarrow (\exists p' \in S_1)\ [p \xrightarrow{e}_1 p' \wedge (p',q) \in R]]$
     Ce qui démontre le troisième item de la proposition pour $B'_c = g \Rightarrow skip$.

   - Si $B'_c = \Phi(sub) = g_1 \Rightarrow e_1; g_2 \Rightarrow e_2; ...; g_n \Rightarrow e_n$. En remplaçant B'_c par sa définition, on obtient :
     Cette formule peut être décomposée en deux formules. La première formule est

$$(stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow stateT_2 \in pre(sub, T_2)$$

     qui est équivalente au deuxième item de la proposition à un renommage près.
     La deuxième formule est la suivante.

$$\begin{aligned}
\equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow \\
& ((\forall q_2 \in S_2)...(\forall q_{n+1} \in S_2)\ [(g_1(stateT_2) \wedge g_2(q_2) \wedge .... \wedge g_n(q_n)) \wedge \\
& stateT_2 \xrightarrow{e}_1 q_2 \wedge ... \wedge q_n \xrightarrow{e}_1 q_{n+1} \Rightarrow \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q_{n+1}) \in R]) \\
\equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow \qquad (\text{Def. 1.14 p. 6}) \\
& (\forall q_{n+1} \in S_2)\ (\forall c)\ (c = CheminVal(sub, T_2) \wedge \\
& stateT_2 = origine(c) \wedge q_{n+1} = extremite(c)) \Rightarrow \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q_{n+1}) \in R])
\end{aligned}$$

$$
\begin{aligned}
\phi'_2 \equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [g_1 \Rightarrow e_1; g_2 \Rightarrow e_2; ...; g_n \Rightarrow e_n] \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]) \\
\equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow [P | g_1 \Rightarrow B_2; ...; g_2 \Rightarrow B_n] && \text{(Lem. 2.1 p. 11)} \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]) \\
\equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow P\ \wedge\ [g_1 \Rightarrow B_2; ...; g_2 \Rightarrow B_n] && (\mathcal{WP}\text{ 4 Tab. 2}) \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]) \\
\equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow P \wedge && \text{(Lem. 3.1 p. 20)} \\
& ((\forall q_2 \in S_2)...(\forall q_{n+1} \in S_2)\ ((g_1(stateT_2) \wedge g_2(q_2) \wedge .... \wedge g_n(q_n)) \wedge \\
& stateT_2 \xrightarrow{e}_1 q_2 \wedge ... \wedge q_n \xrightarrow{e}_1 q_{n+1} \Rightarrow [stateT_2 := q_{n+1}]) \\
& ((\exists\ p' \in S_1)\ (stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R)) \\
\equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow P \wedge && (\mathcal{WP}\text{ 1 Tab. 2}) \\
& ((\forall q_2 \in S_2)...(\forall q_{n+1} \in S_2)\ [(g_1(stateT_2) \wedge g_2(q_2) \wedge .... \wedge g_n(q_n)) \wedge \\
& stateT_2 \xrightarrow{e}_1 q_2 \wedge ... \wedge q_n \xrightarrow{e}_1 q_{n+1} \Rightarrow \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q_{n+1}) \in R]) \\
\equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow stateT_2 \in pre(sub, T_2) \wedge && \text{(Lem. 3.2 p. 21)} \\
& ((\forall q_2 \in S_2)...(\forall q_{n+1} \in S_2)\ [(g_1(stateT_2) \wedge g_2(q_2) \wedge .... \wedge g_n(q_n)) \wedge \\
& stateT_2 \xrightarrow{e}_1 q_2 \wedge ... \wedge q_n \xrightarrow{e}_1 q_{n+1} \Rightarrow \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q_{n+1}) \in R])
\end{aligned}
$$

$$
\begin{aligned}
\equiv & (\forall c \in Chemin(sub, T_2))\ (\forall q_{n+1} \in S_2)\ ((stateT_1 \mapsto stateT_2) \in R \wedge \\
& P_a \wedge (stateT_2 = origine(c) \wedge q_{n+1} = extremite(c)) \Rightarrow \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q_{n+1}) \in R]))
\end{aligned}
$$

Cette formule est équivalente au troisième item de la proposition à un renommage près.

2. Cas général. $B'_c = \Phi(sub)$ de forme canonique $\Phi(sub_1) \,[\!]\, ... \,[\!]\, \Phi(sub_n)$.

$$
\begin{aligned}
\phi'_2 \equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow \\
& [\Phi(sub_1) \,[\!]\, ... \,[\!]\, \Phi(sub_n)] \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]) \\
\equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow && \text{(Prop. 2.2 p. 11)} \\
& [P_1 \wedge ... \wedge P_n | B_1 \,[\!]\, ... \,[\!]\, B_n] \\
& ((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]) \\
\equiv & (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow && (\mathcal{WP}\ 4, 5\text{ Tab. 2}) \\
& P_1 \wedge ... \wedge P_n \wedge \\
& ([B_1]((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R])) \wedge \\
& ... \wedge \\
& ([B_n]((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]))
\end{aligned}
$$

$\equiv (stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow$ (Lem. 3.2 p. 21)
$stateT_2 \in pre(sub, T_2) \wedge$
$([B_1]((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R])) \wedge$
$... \wedge$
$([B_n]((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]))$

Cette formule se décompose en deux. La première permet de prouver le deuxième item de la proposition.

$$(stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow stateT_2 \in pre(sub, T_2)$$

La deuxième est la suivante.

$((stateT_1 \mapsto stateT_2) \in R \wedge P_a \Rightarrow$
$([B_1]((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R])) \wedge$
$... \wedge$
$([B_n]((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto stateT_2) \in R]))$
$\equiv \bigwedge_{i=1}^{n}((\forall c \in Chemin(sub_i, T_2))\ (\forall q_{n+1} \in S_2)\ ((stateT_1 \mapsto stateT_2) \in R \wedge$ (Hyp. d'induction)
$P_a \wedge (stateT_2 = origine(c) \wedge q_{n+1} = extremite(c)) \Rightarrow$
$((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q_{n+1}) \in R])))$
$\equiv (\forall c \in Chemin(sub, T_2))\ (\forall q_{n+1} \in S_2)\ ((stateT_1 \mapsto stateT_2) \in R \wedge$ (Def. 1.14 p. 6)
$P_a \wedge (stateT_2 = origine(c) \wedge q_{n+1} = extremite(c)) \Rightarrow$
$((\exists\ p' \in S_1)\ [stateT_1 \xrightarrow{e}_1 p' \wedge (p' \mapsto q_{n+1}) \in R]))$

Cette formule est équivalente au troisième item de la proposition.

□

Le corollaire suivant est un cas particulier de la proposition 3.4 lorsque χ est une application de A_1 vers A_2, on parle alors de renommage d'étiquettes.

Corollaire 3.3. *Si T_2 raffine T_1 par rapport à une relation R et un renommage d'étiquettes χ, alors on a les propriétés suivantes :*

1. $(s_0^1, s_0^2) \in R$
2. $(\forall e \in A_1)\ (\forall e' \in A_2)\ (\forall p \in S_1)\ (\forall q \in S_2)\ [(p, q) \in R\ \wedge\ e' = \chi(e)\ \wedge\ p \xrightarrow{e}_1] \Rightarrow q \xrightarrow{e'}_2$
3. $(\forall e \in A_1)\ (\forall e' \in A_2)\ (\forall p \in S_1)\ (\forall q \in S_2)\ (\forall q' \in S_2)$
   $[(p, q) \in R \wedge e' = \chi(e) \wedge q \xrightarrow{e'}_2 q' \wedge p \xrightarrow{e}_1] \Rightarrow (\exists p' \in S_1)\ [p \xrightarrow{e}_1 p' \wedge (p', q') \in R]$

3.3 Relation entre un STE et plusieurs STEs

Nous présentons la construction B utilisée. Ensuite, nous définissons le raffinement B entre un STE et plusieurs STEs.

3.3.1 Construction B

Nous considérons les STEs $T_0 = (A_0,\ S_0,\ s_0^0,\ F_0,\ \rightarrow_0)$ et $T_i = (A_i, S_i, s_0^i, F_i, \rightarrow_i)$, $1 \leq i \leq n$. dont les alphabets A_i sont deux à deux disjoints, une relation $R \subseteq S_0 \times (S_1 \times ... \times S_k)$ et une application χ de A_0 vers $\mathcal{E}(\bigcup_{i=1}^{n} A_i)$. Nous construisons les spécifications B, M_a, M_{c1},... et M_{cn} associées respectivement à T_0, T_1,... et T_n avec $stateT_i$ comme variable. M_c' qui raffine M_a et inclut M_{c1},... et M_{cn} est obtenu par traduction de R et de χ. M_c' comporte toutes les opérations de M_a définies par l'application χ à partir des opérations des spécifications M_{c1},... ,M_{cn}. L'invariant de collage de M_c' est de la forme $(stateT \mapsto (stateT_1 \mapsto ...(stateT_{n-1} \mapsto stateT_n)...) \in R$. Nous dirons que les STEs T_1, ..., T_n raffinent T_0 par rapport à R et χ, si le raffinement M_c' est vérifié en B. Cette construction B est décrite figure 7.

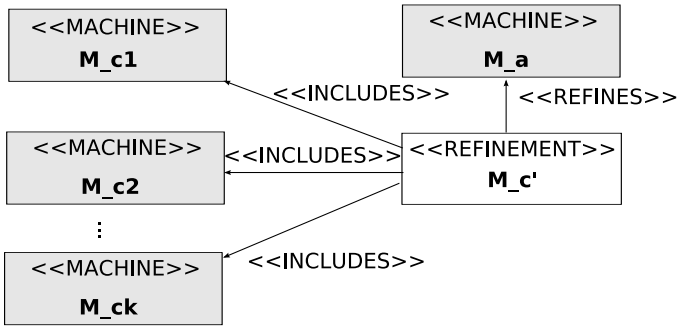


Figure 7: B

3.3.2 Raffinement B

Proposition 3.5. *Soient $T_0, ..., T_n$ $n+1$ STEs et $T = (A,\ S,\ s^0,\ F,\ \rightarrow)$ le produit libre de $T_1, ..., T_n$. Si les STEs $T_1, ..., T_n$ raffinent T_0 par rapport à une relation R et une application χ, alors on a les trois propriétés suivantes, pour toute étiquette e appartenant à A_0 et toute composition d'étiquette $sub = \chi(e)$:*

1. $(s_0^0, (s_0^1, ..., s_0^n)) \in R$

2. $(\forall p \in S_0)\ (\forall (q_1, ..., q_n) \in S)\ [(p, (q_1, ..., q_n)) \in R\ \wedge p \xrightarrow{e}_1 \Rightarrow (q_1, ..., q_n) \in pre(sub, T)]$

3. $(\forall p \in S_0)\ (\forall (q_1, ..., q_n) \in S)\ (\forall (q_1', ..., q_n') \in S)\ (\forall c \in Chemin(sub, T))$
$[(p, (q_1, ..., q_n)) \in R\ \wedge origine(c) = (q_1, ..., q_n) \wedge extremite(c) = (q_1', ..., q_n') \wedge p \xrightarrow{e}_0$
$\Rightarrow (\exists p' \in S_0)\ [p \xrightarrow{e}_0 p' \wedge (p', (q_1', ..., q_n')) \in R]]$

Proof. Pour démonter cette proposition, nous procédons de la même manière que dans la proposition 3.4. □

4 Travaux connexes

Certains travaux se sont intéressés à définir la sémantique du raffinement B en utilisant des relations entre STEs. Citons [BJK00] et [LB03]. Cette manière de décrire le raffinement a deux principaux avantages : (i) la possibilité d'effectuer la vérification de raffinements par d'autres outils que les prouveurs de théorèmes (ii) la possibilité de combiner les techniques des prouveurs de théorèmes avec les techniques de "model-checking" et ainsi de vérifier de nouvelles propriétés qui ne sont pas supportées dans la méthode B. Comme toute approche se basant sur les techniques de "model-checking" ces deux approches sont limitées lorsqu'il a une explosion combinatoire de l'ensemble des états.

D'autres travaux [BPS05] se sont intéressés à la description du comportement des spécifications B en se basant sur leurs spécifications abstraites antérieures. Le but est de donner une vue graphique des spécifications B qui aide à mieux les comprendre et non de vérifier la relation de raffinement.

Les approches citées supportent les spécifications en B classique [LB03], en B événementiel [BJK00, BPS05] ou dans les deux formalismes [BLLS08, Sto07]. Notons que dans ces différentes approches, les liens **INCLUDES** et **SEES** ne sont pas pris en considération. Dans ce qui suit, nous présentons brièvement ces trois travaux.

4.1 Les travaux de [BJK00]

Les auteurs s'intéressent à la relation de raffinement B en rapport avec la sémantique opérationnelle. Ils définissent une sémantique opérationnelle des spécifications B événementiel, en utilisant des systèmes de transitions interprétés. Les systèmes de transitions interprétés sont caractérisés par une fonction d'interprétation associée aux états.

Ils se restreignent à un raffinement particulier appelé raffinement modulaire. Le raffinement modulaire est caractérisé par une relation de collage entre les états de deux systèmes de transitions associés aux spécifications abstraite et concrète, et par un ensemble de relations qui lient ces deux systèmes de transitions. La notion de raffinement modulaire est comparée à la relation de simulation de Milner et à la relation de ready-simulation.

Le but est de combiner les techniques de prouveur de théorème aux techniques de "model-checking" pour effectuer des vérifications. La vérification est ainsi faite en deux étapes : au niveau syntaxique par le prouveur de théorème et au niveau opérationnel pour vérifier des formules LTL par le model-checker.

4.2 L'approche *GeneSyst*

Dans [BC00, PS04, BPS05, Sto07], les auteurs présentent une méthode et un outil, appelé *GeneSyst* pour construire des systèmes de transitions symboliques (SLTS) à partir de spécifications B. Le SLTS construit donne une vue graphique et représente tous les comportements du modèle B. Pour définir les liens entre la spécification B et le STLS correspondant, ils utilisent les traces associées à une spécification B et aux chemins associés au système de transitions symboliques correspondant et ils montrent leur équivalence. Cette approche est applicable à des spécifications

B événementielles et dans [Sto07] une traduction d'événements en opérations est proposée pour pouvoir étendre l'approche au B classique.

L'approche prend en considération la représentation du comportement d'un raffinement en introduisant la notion de hiérarchie dans les systèmes de transitions. Dans un système de transition associé à un raffinement, on trouve la structure générale de la représentation des comportements du système abstrait, ainsi que les noms des états abstraits. En effet, les états du système de transition associés au raffinement sont construits par la projection des états du système de transitions en tenant compte de l'invariant de collage. Ensuite, la relation de transition entre les états est construite en se basant sur le comportement abstrait. Il démontre que les chemins associés à un système de transition obtenu par la projection du système de transition abstrait sont égaux aux traces générées par le raffinement. Le point fort de cette approche et qu'elle est applicable à des systèmes infinis. Ils ne s'intéressent pas à la vérification du raffinement par rapport à la spécification abstraite, mais se basent sur les propriétés du raffinement pour la construction du système de transition.

4.3 *ProB*

ProB [LB03, LB08] est un animateur et un "model-checker" pour la méthode B. Il a été développé pour visualiser le comportement dynamique d'une machine. Ce model-checker peut dans le cas d'ensembles finis de petite taille, explorer de façon exhaustive l'ensemble des états pouvant être atteints par une machine B et ainsi valider une machine. Même lorsque la recherche ne peut se faire de façon exhaustive, il est possible, grâce à un model-checker, de découvrir des contre-exemples de violation d'un invariant. Il peut être vu comme un complément aux démonstrateurs de théorèmes.

ProB a été développé initialement pour le B classique et il a été étendu ensuite et porté sur la plate-forme Rodin pour supporter le B événementiel [BLLS08], *ProB* ne supporte pas l'utilisation des substitutions préconditionnées ni les liens de composition **INCLUDES** et **SEES**.

ProB supporte la vérification automatique du raffinement entre spécifications B [LB05]. Le raffinement considéré par *ProB* est différent du raffinement B, car ProB ne prend pas en compte l'invariant de collage, la vérification concerne seulement le raffinement de traces. Pour cela, une sémantique de traces est associée aux spécifications B. Plus précisément une machine M est un raffinement de traces d'une machine N si n'importe quelle trace de N est une trace de M (si une trace est possible dans le système concret elle est possible dans le système abstrait). Le rôle du *ProB* est d'essayer de trouver un contre-exemple pour le raffinement, c'est-à-dire de trouver une séquence d'appels d'opérations qui sont permis dans un raffinement et qui ne sont pas permis dans la spécification abstraite correspondante. Techniquement parlant, un algorithme est proposé qui parcourt les états des deux systèmes, en construisant une structure de collage R entre eux. Dans le cas de succès de l'algorithme, R lie chaque état initial concret à un ensemble d'états abstraits et la condition de simulation est vérifiée pour chaque couple d'états liés. Ils vérifient ainsi le raffinement de trace entre les deux STEs.

5 Bilan et perspectives

Conclusion

Nous avons étudié le raffinement B en terme de STEs, cette étude est basée sur une traduction de STEs en spécifications B, les éléments des alphabets étant traduits directement par des opérations B. Du point de vue de la sémantique opérationnelle, si l'on considère que la sémantique d'une machine B peut être représentée par un STE, la sémantique d'une machine B traduction d'un STE n'est autre que ce ce STE lui-même. Notre étude a consisté à considérer plusieurs sortes de raffinement.

Le raffinement le plus simple avec la clause **REFINES**, dans ce cas nous avons comparé le raffinement avec plusieurs relations entre STEs (simulation, bisimulation et ready-simulation). Nous avons montré que nous pouvons modéliser en B les différentes relations, ainsi dans le cadre de la vérification de l'assemblage des composants au niveau protocol, nous pouvons choisir la relation à vérifier.

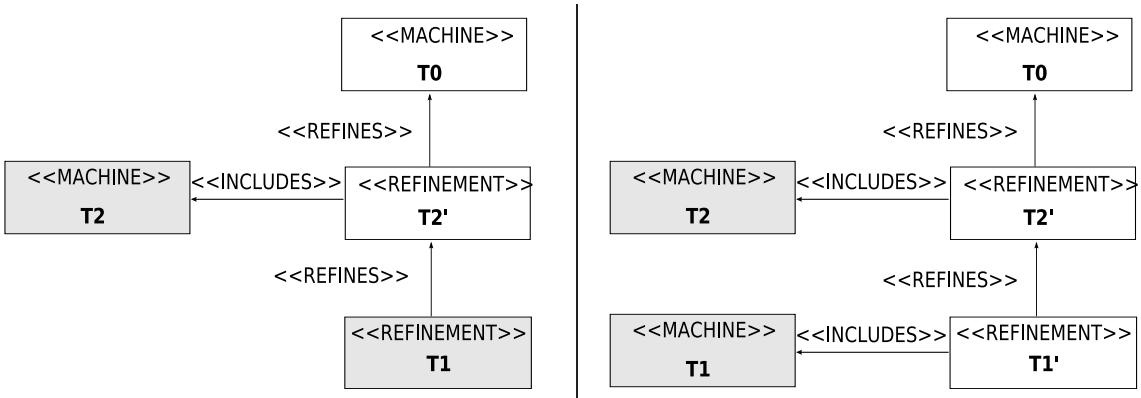
Nous avons introduit une nouvelle notion, celle de composition d'étiquettes, et considéré un schéma B basé non seulement sur la clause **REFINES**, mais aussi sur la clause **INCLUDES** et impliquant deux STEs, dans ce cas nous avons des relations entre les chemins des STEs.

Enfin la dernière étude consiste à généraliser le schéma précédent en considérant plusieurs inclusions de STEs par la clause **INCLUDES**. Le schéma mis en évidence est utilisé lorsque l'on veut vérifier l'assemblage de plusieurs composants. Une limitation à cette approche est que les différents composants doivent être indépendants au niveau protocole, c'est-à-dire que la communication entre composants est modélisée par l'intermédiaire d'un médiateur ou d'un contrôleur.

Perspectives

Nous avons présenté le raffinement par rapport à une relation d'étiquette qui associe à chaque étiquette du STE abstrait, une étiquette ou une composition d'étiquettes du STE concret. Une extension possible serait de considérer des relations plus élaborées qui permettent de lier des compositions d'étiquettes entre elles. Cela correspond au schéma B de la figure 8.

D'autres extensions possibles seraient de considérer d'une part des STEs gardés, mais aussi des données associées aux STEs.



(a) B (b) B

Figure 8: Construction B

A Quelques règles sur les substitutions généralisées

Les axiomes du calcul de $\mathcal{WP}$ sur les substitutions primitives sont présentés dans le tableau 2.

Substitution	$\mathcal{WP}$	Condition
1. $[x := E]R$	remplacement par E des occurrences libres de x dans R	
2. $[x, y := E, F]R$	$[z := F][x := E][y := z]R$	$z\ E, E, R$
3. $[skip]R$	R	
4. $P\|S$	$P \wedge [S]R$	
5. $[P \parallel S]R$	$[P]R \wedge [S]R$	
6. $[P \Rightarrow S]R$	$P \Rightarrow [S]R$	
7. $[@z.S]R$	$\forall z.[S]R$	$z\ R$
8. $[S;T]R$	$[S][T]R$	
9. $\mathcal{W}(P, S, J, V)$	$J \wedge$ $\forall x.((J \wedge P) \Rightarrow [S]J) \wedge$ $\forall x.(J \Rightarrow V \in \mathbb{N}) \wedge$ $\forall x.((J \wedge P) \Rightarrow [n := V][S](V < n) \wedge$ $\forall x.((J \wedge \neg P) \Rightarrow R)$	

Table 2: Les $\mathcal{WP}$ des substitutions primitives

B POs générées pour l'exemple 3.1

Règle	Gauche	Droite
$R1$	$P \Rightarrow (Q\|S)$	$(P \Rightarrow Q)\|(P \Rightarrow S)$
$R2$	$(P\|S);T$	$P\|(S;T)$
$R3$	$S;(P\|T)$	$[S]P\|(S;T)$
$R4$	$P\|(Q\|S)$	$(P \wedge Q)\|S$
$R5$	$(P\|S) \,[\!]\, T$	$P\|(S \,[\!]\, T)$
$R6$	$S \,[\!]\, T$	$T \,[\!]\, S$
$R7$	$T \,[\!]\, (P\|S)$	$P\|(T \,[\!]\, S)$
$R8$	$S;(T \,[\!]\, U)$	$(S;T) \,[\!]\, (S;U)$
$R9$	$(T \,[\!]\, U);S$	$(T;S) \,[\!]\, (U;S)$
$R10$	$P \Rightarrow (T \,[\!]\, S)$	$(P \Rightarrow T) \,[\!]\, (P \Rightarrow S)$
$R11$	$P \Rightarrow (T;S)$	$(P \Rightarrow T);S$
$R12$	$P \Rightarrow (Q \Rightarrow S)$	$(P \wedge Q) \Rightarrow S)$
$R13$	$(x := E);(P \Rightarrow T)$	$[x := E]P \Rightarrow (x := E;T)$

Table 3: Quelques règles d'équivalence de substitutions

PO1 "'Check precondition (stateT2∈ q0,q2,q5,q7,q11,q13) deduction'"
⇒ stateT2$1∈ {q0,q2,q5,q7,q11,q13}

PO2 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧ "'Local hypotheses'" \ stateT2$1 = q0 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(q1∈ P2 ∧ stateT1$0↦q1∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(q1∈ P2 ∧ stateT1$0↦q1∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(q1∈ P2 ∧ stateT1$0↦q1∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(q1∈ P2 ∧ stateT1$0↦q1∈ RR)) ∧
"'Check operation refinement - ref 4.4, 5.5'"
⇒ stateT1 = p0

PO3 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧
"'Local hypotheses'"
stateT2$1 = q0 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(q1∈ P2 ∧ stateT1$0↦q1∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(q1∈ P2 ∧ stateT1$0↦q1∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(q1∈ P2 ∧ stateT1$0↦q1∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(q1∈ P2 ∧ stateT1$0↦q1∈ RR)) ∧
"'Check that the invariant (stateT1↦stateT2∈ RR) is preserved by the operation - ref 4.4, 5.5'"
"'Check operation refinement - ref 4.4, 5.5'"
⇒ ∃(stateT1$0).(stateT1$0∈ p2,p3 ∧ stateT1$0↦q1∈ RR)

PO4 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧

```
aa =
  PRE stateT1 ∈{p0,p1,p6,p10} THEN
      SELECT stateT1 = p0 THEN stateT1 :∈{p2,p3}
      WHEN stateT1 = p1 THEN stateT1 :∈{p4,p5}
      WHEN stateT1 = p6 THEN stateT1 :∈{p7,p8,p9}
      WHEN stateT1 = p10 THEN stateT1 :∈{p11,p12}
      WHEN  stateT1 = p13 THEN stateT1 :∈{p14,p15}
      END
  END
```

(a) Abstrait

```
aa =
      PRE stateT2 ∈{q0,q2,q5,q7,q11,q13} THEN
          SELECT stateT2 = q0 THEN stateT2 :=  q1
          WHEN stateT2 = q2 THEN stateT2 :∈{q3,q4}
          WHEN stateT2 = q5 THEN stateT2 :=q6
          WHEN stateT2 = q7 THEN stateT2 :∈{q8,q9,q10}
          WHEN stateT2 = q11 THEN stateT2 := q12
          WHEN stateT2 = q13 THEN stateT2 :∈{q14,q15}
          END
      END
```

(b) Raffinement

```
PROPERTIES
    RR ∈ P1 ↔ P2 ∧
    RR = {(p0 ↦q0), (p1 ↦ q0), (p2 ↦q1), (p5 ↦ q1),
        (p6 ↦q2), (p6 ↦ q5), (p7 ↦q3), (p8 ↦ q4),
        (p8 ↦ q6), (p10 ↦ q7), (p11 ↦ q8), (p11 ↦ q9),
        (p11 ↦ q10), (p16 ↦ q13),...}
```

(c) Relation entre états

Figure 9: Modèle B

"'Local hypotheses'"
stateT2$0∈ q3,q4 ∧
stateT2$1 = q2 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
"'Check that the invariant (stateT1—-¿stateT2: RR) is preserved by the operation - ref 4.4, 5.5'"
"'Check operation refinement - ref 4.4, 5.5'"
⇒ stateT1 = p0

PO5 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧
"'Local hypotheses'"
stateT2$0∈ q3,q4 ∧
stateT2$1 = q2 ∧

stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
"'Check that the invariant (stateT2∈ P2) is preserved by the operation - ref 4.4, 5.5'"
⇒ stateT2$0∈ P2

PO6 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧
"'Local hypotheses'"
stateT2$0∈ q3,q4 ∧
stateT2$1 = q2 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
"'Check that the invariant (stateT1—-¿stateT2: RR) is preserved by the operation - ref 4.4, 5.5'"
⇒ ∃(stateT1$0).(stateT1$0∈ p2,p3 ∧ stateT1$0↦stateT2$0∈ RR)

PO7 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10
"'Local hypotheses'"
stateT2$1 = q5 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(q6∈ P2 ∧ stateT1$0↦q6∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(q6∈ P2 ∧ stateT1$0↦q6∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(q6∈ P2 ∧ stateT1$0↦q6∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(q6∈ P2 ∧ stateT1$0↦q6∈ RR)) ∧
"'Check that the invariant (stateT1↦stateT2∈ RR) is preserved by the operation - ref 4.4, 5.5'"
⇒ stateT1 = p0

PO8 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧ "'Local hypotheses'"
stateT2$1 = q5 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(q6∈ P2 ∧ stateT1$0↦q6∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(q6∈ P2 ∧ stateT1$0↦q6∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(q6∈ P2 ∧ stateT1$0↦q6∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(q6∈ P2 ∧ stateT1$0↦q6∈ RR)) ∧
"'Check that the invariant (stateT1↦stateT2∈ RR) is preserved by the operation - ref 4.4, 5.5'"
⇒ ∃(stateT1$0).(stateT1$0∈ p2,p3 ∧ stateT1$0↦q6∈ RR)

PO9 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧

"'Local hypotheses'"
stateT2$0∈ q8,q9,q10 ∧
stateT2$1 = q7 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
"'Check that the invariant (stateT1↦stateT2∈ RR) is preserved by the operation - ref 4.4, 5.5'"
⇒ stateT1 = p0

P10 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧ "'Local hypotheses'"
stateT2$0∈ q8,q9,q10 ∧
stateT2$1 = q7 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR))
"'Check that the invariant (stateT2: P2) is preserved by the operation - ref 4.4, 5.5'"
⇒ stateT2$0∈ P2

P11 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧ "'Local hypotheses'"
stateT2$0∈ q8,q9,q10 ∧
stateT2$1 = q7 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
"'Check that the invariant (stateT1↦stateT2∈ RR) is preserved by the operation - ref 4.4, 5.5'"
⇒ ∃(stateT1$0).(stateT1$0∈ p2,p3 ∧ stateT1$0↦stateT2$0∈ RR)

P12 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧
"'Local hypotheses'"
stateT2$1 = q11 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(q12∈ P2 ∧ stateT1$0↦q12∈ RR))∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(q12∈ P2 ∧ stateT1$0↦q12∈ RR))∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(q12∈ P2 ∧ stateT1$0↦q12∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(q12∈ P2 ∧ stateT1$0↦q12∈ RR)) ∧
"'Check that the invariant (stateT1↦stateT2∈ RR) is preserved by the operation - ref 4.4, 5.5'"
⇒ stateT1 = p0

P13 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧
"'Local hypotheses'"
stateT2$1 = q11 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(q12∈ P2 ∧ stateT1$0↦q12∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(q12∈ P2 ∧ stateT1$0↦q12∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(q12∈ P2 ∧ stateT1$0↦q12∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(q12∈ P2 ∧ stateT1$0↦q12∈ RR)) ∧
"'Check that the invariant (stateT1↦stateT2∈ RR) is preserved by the operation - ref 4.4, 5.5'"
⇒ ∃(stateT1$0).(stateT1$0∈ p2,p3 ∧ stateT1$0↦q12∈ RR)

P14 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧
"'Local hypotheses'"
stateT2$0∈ q14,q15 ∧
stateT2$1 = q13 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
"'Check that the invariant (stateT1—-¿stateT2: RR) is preserved by the operation - ref 4.4, 5.5'"
⇒ stateT1 = p0

P15 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧
"'Local hypotheses'"
stateT2$0∈ q14,q15 ∧
stateT2$1 = q13 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
"'Check that the invariant (stateT2∈ P2) is preserved by the operation - ref 4.4, 5.5'"
⇒ stateT2$0∈ P2

P16 "'aa preconditions in this component'"
stateT2$1∈ q0,q2,q5,q7,q11,q13 ∧ stateT1∈ p0,p1,p6,p10 ∧
"'Local hypotheses'"
stateT2$0∈ q14,q15 ∧
stateT2$1 = q13 ∧
stateT1 = p13 ⇒ ∀(stateT1$0).(stateT1$0∈ p14,p15 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR))∧
stateT1 = p10 ⇒ ∀(stateT1$0).(stateT1$0∈ p11,p12 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR))∧
stateT1 = p6 ⇒ ∀(stateT1$0).(stateT1$0∈ p7,p8,p9 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR))∧

stateT1 = p1 ⇒ ∀(stateT1$0).(stateT1$0∈ p4,p5 ⇒ ¬(stateT2$0∈ P2 ∧ stateT1$0↦stateT2$0∈ RR)) ∧
"'Check that the invariant (stateT1↦stateT2∈ RR) is preserved by the operation - ref 4.4, 5.5'"
⇒ ∃(stateT1$0).(stateT1$0∈ p2,p3 ∧ stateT1$0↦stateT2$0∈ RR)

References

[Arn92] A. Arnold. *Systèmes de transitions finis et sémantique des processus communicants.* Masson, Paris, 1992.

[BC00] D. Bert and F. Cave. Construction of finite labelled transistion systems from b abstract systems. In *IFM '00: Proceedings of the Second International Conference on Integrated Formal Methods*, pages 235–254. Springer-Verlag, 2000.

[BIM95] B. Bloom, S. Istrail, and R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, January 1995.

[BJK00] F. Bellegarde, J. Julliand, and O. Kouchnarenko. Ready-simulation is not ready to express a modular refinement relation. In *Fundamental Aspects of Software Engineering (FASE'00)*, volume 1783 of *LNCS*, pages 266–283. Springer Verlag, 2000.

[BLLS08] J. Bendisposto, M. Leuschel, O. Ligot, and M. Samia. La validation de modèles Event-B avec le plug-in ProB pour Rodin. *Technique et Science Informatiques*, 27(8):1065–1084, 2008.

[BPS05] D. Bert, M.-L. Potet, and N. Stouls. Genesyst: a tool to reason about behavioral aspects of b event specifications. application to security properties. In *ZB 2005: Formal Specification and Development in Z and B, 4th International Conference of B and Z Users*, volume 3455 of *LNCS*, pages 299–318. Springer-Verlag, 2005.

[LB03] M. Leuschel and M. Butler. ProB: A model checker for B. In Keijiro Araki, Stefania Gnesi, and Dino Mandrioli, editors, *FME 2003: Formal Methods*, LNCS 2805, pages 855–874. Springer-Verlag, 2003.

[LB05] M. Leuschel and M. Butler. Automatic refinement checking for b. In *ICFEM*, pages 345–359, 2005.

[LB08] M. Leuschel and M. Butler. Prob: an automated analysis toolset for the b method. *STTT*, 10(2):185–203, 2008.

[Mil89] R. Milner. *Communication and concurrency.* Prentice-Hall, Inc., 1989.

[PS04] M.-L. Potet and N. Stouls. Explicitation du contrôle de développement B événementiel. In J. Julliand, editor, *Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL'04)*, pages 13–27, JUN 2004.

[Sto07] N. Stouls. *Systèmes de transitions symboliques et hiérarchiques pour la conception et la validation de modèles B raffinés*. PhD thesis, Institut polytechnique de Grenoble, décembre 2007.