Département Informatique

LaMI

Laboratoire de Méthodes Informatiques

C.N.R.S. U.M.R. 8042

Une théorie du raffinement orientée propriétés pour les automates communicants

Delphine Longuet

Mémoire de stage de D.E.A.

Encadrants : Marc Aiguier et Pascale Le Gall (LaMI)
Christophe Gaston (CEA)

e-mail : aiguier, legall, dlonguet@lami.univ-evry.fr, gaston@cea.fr

Juin 2004

LaMI - Université d'Évry Val d'Essonne
Tour Évry 2 / 4$^{\text{ème}}$ étage
523 place des Terrasses
91000 Évry - France

Table des matières

Introduction

L'objectif de ce travail est, d'une part, de définir une théorie du raffinement dans le cadre du formalisme des automates communicants étendus aux données et d'autre part, de définir un formalisme axiomatique dédié à la spécification des systèmes réactifs, et dont la sémantique est fondée sur ces automates communicants.

Les formalismes utilisés pour la spécification de systèmes réactifs sont le plus souvent à base d'automates communicants, c'est-à-dire des automates dont les transitions sont étiquetées par des entrées-sorties. Il existe plusieurs formalismes de ce type, on peut par exemple citer les *Input Output Automata* de N. A. Lynch [Lyn88], les *Input Output State Machines* de M. Phalippou [Pha94], ou les *Input Output Transition Systems* de J. Tretmans [Tre95]. Dans ce cadre de formalisation, l'implantation du système ainsi que sa spécification sont représentées par des automates de même nature. Ces automates abstraient les comportements internes du système pour ne conserver que ses communications sous forme d'entrées-sorties. Le système est donc représenté à deux niveaux d'abstraction, le plus abstrait étant l'automate de la spécification, le plus concret celui représentant l'implantation. On compare ces deux automates à l'aide d'une relation de conformité basée sur l'équivalence comportementale [Jér01]. La comparaison s'effectue entre les traces, c'est-à-dire les suites d'entrées-sorties, obtenues lors de l'exécution de l'implantation, et les traces attendues par la spécification. L'implantation est conforme à la spécification si les traces de l'implantation sont incluses dans l'ensemble des traces possibles de la spécification.

Deux limitations associées à ce type de formalisation apparaissent. La première est directement attachée aux formalismes eux-mêmes. En effet, l'implantation peut être considérée conforme à la spécification pour une suite d'entrées-sorties, alors que ces entrées-sorties ne conduisent pas au même état interne du système. La raison est que la spécification d'un système réactif par des automates communicants réduit le comportement d'un système à ses entrées-sorties et abstrait complètement son comportement interne. La seconde est plus méthodologique. En effet, avec ce type de formalisation, on se contente de deux étapes de spécification, la spécification du système et son implantation. Cependant, on constate en pratique que plusieurs niveux de spécification peuvent être utiles avant d'aboutir à l'implantation. Nous proposons alors, dans le cadre de ce stage, de répondre à ces deux limitations.

Tout d'abord, pour prendre en compte le comportement interne des systèmes, et ainsi

répondre à la première limitation, nous choisissons de nous intéresser à une catégorie d'automates communicants plus riches que ceux évoqués plus haut, les systèmes de transitions étiquetées par des entrées-sorties et étendus aux données (EIOLTS, pour Extended Input Output Labelled Transition System) introduits dans [RGLG03]. Ces automates permettent de représenter, en plus des communications du système, l'évolution de l'état de ses variables au cours des exécutions. Nous proposons alors de définir une théorie du raffinement danx le cadre des EIOLTS. Le raffinement consiste à expliciter une spécification abstraite de haut niveau décrivant les besoins utilisateur en une spécification plus concrète détaillant les choix d'implantation. Ceci permet de construire de façon incrémentale, à partir de l'EIOLTS de la spécification, l'EIOLTS représentant l'implantation réelle du système. Le système sera ainsi représenté à différents niveaux d'abstraction, l'EIOLTS obtenu à chaque étape de raffinement étant de plus en plus concret au fur et à mesure où l'on se rapproche de l'implantation. Cela nous permettra de définir une notion de conformité entre un EIOLTS et son raffinement qui soit conservée à chaque étape, de manière à ce que l'EIOLTS le plus concret soit conforme à la spécification initiale. Cette notion de conformité sera une extension, aux données manipulées par l'EIOLTS, de celle évoquée plus haut. En effet, on ne comparera plus deux automates seulement à travers leurs suites d'entrées-sorties, mais également à travers l'évolution de leur état interne au cours des exécutions.

La spécification d'un système réactif à base d'automates communicants définit un modèle mathématique du système. Dans ce sens, ces modèles définissent une dénotation sémantique des systèmes réactifs. Dans un souci d'abstraire ces comportements, l'idée est de définir un formalisme logique (dit aussi « orienté propriétés ») dont la sémantique sera fondée sur ces automates. C'est ce que nous proposons de faire dans la seconde partie de ce manuscrit. Nous allons définir un formalisme axiomatique dont la sémantique sera fondée sur les EIOLTS et permettre ainsi de spécifier des systèmes réactifs de façon plus abstraite, en s'intéressant aussi au « *quoi* » (c'est-à-dire *ce que le système est supposé faire*), et pas seulement au « *comment* » (c'est-à-dire *comment il est supposé le faire*). À une spécification dans ce formalisme sera donc associé, non pas un EIOLTS, mais un ensemble d'EIOLTS, chacun d'eux devant valider l'ensemble des propriétés exprimées dans la spécification. Les EIOLTS ne seront alors plus considérés comme des spécifications mais comme des modèles de notre formalisme.

Nous définissons alors un formalisme axiomatique dédié à la spécification de systèmes réactifs. Celui-ci devra permettre d'exprimer des propriétés ayant un sens pour les EIOLTS. Il faudra donc qu'il prenne en compte à la fois la dynamique du système et les données. Dans les EIOLTS, la dynamique du système s'exprime par la notion d'exécution (enchaînement d'actions) et par les communications (émissions et réceptions de messages). Pour prendre en compte ces différents aspects, nous nous inspireront des logiques temporelles propositionnelles (LTL, CTL, [AGM92]) pour exprimer des propriétés sur les chemins, et de la logique pour spécifications mixtes introduite par M. Aiguier, F. Barbier et P. Poizat dans [ABP02] pour prendre en compte les communications par l'ajout de modalités. Ces logiques nous permettront d'exprimer des propriétés sur les comportements

du système, tout en prenant en compte les communications. De plus, nous choisirons de placer ce formalisme dans le premier ordre, afin de prendre en compte les données manipulées par les EIOLTS.

Dans le cadre de ce formalisme, la notion de conformité s'exprime alors en termes de satisfaction de propriétés. En effet, l'implantation du système sera conforme à la spécification si elle vérifie les propriétés énoncées dans la spécification. Or l'implantation du système peut être représentée à plusieurs niveaux d'abstraction par des EIOLTS. En effet, à l'aide de la relation de raffinement introduite précédemment, elle peut être représentée aussi bien par l'EIOLTS le plus abstrait que par chacun de ses raffinements. Or toutes ces représentations du système doivent être conformes à la spécification du système écrite dans notre formalisme. On veut donc que la classe des modèles associés à une spécification soit l'ensemble des EIOLTS qui représentent un système aux différentes étapes de raffinement. Pour cela, il faut que les propriétés du système exprimées dans la spécification soient vérifiées par tous les EIOLTS représentant le système. On doit donc s'assurer que les propriétés exprimées dans notre formalisme sont conservées au travers du raffinement.

Le plan du rapport s'organise de la façon suivante :

- le premier chapitre rappelle la logique des prédicats du premier ordre, qui va nous permettre de décrire les données manipulées par les EIOLTS, à travers sa syntaxe et sa sémantique ;
- le deuxième chapitre définit tout d'abord les EIOLTS à travers leur syntaxe et leur sémantique, puis introduit la théorie du raffinement que l'on a définie pour les EIOLTS, et enfin, établit le résultat de transitivité du raffinement, tout en assurant la conservation de la correction et de la complétude du raffinement par transitivité ;
- le troisième chapitre introduit le formalisme axiomatique dédié aux EIOLTS que l'on a défini à travers sa syntaxe et sa sémantique et établit le résultat de conservation des propriétés exprimées dans ce formalisme au travers du raffinement.

Chapitre 1

Formalisme pour les données

Les données manipulées par les systèmes réactifs seront décrites à l'aide d'une logique du premier ordre multi-sortes caractérisée, comme toute logique, par une syntaxe et une sémantique. La syntaxe donne les règles de construction des termes et des formules du langage à partir d'un alphabet appelé signature. La sémantique donne un sens mathématique aux différents éléments de la signature dans la théorie des ensembles.

1.1 Syntaxe

Signatures. La signature introduit les éléments de base des constructions syntaxiques du formalisme que sont les termes et les formules.

Définition 1.1.1 (Signature) *Une* signature Σ *est un triplet* (S, F, R) *où :*

- S *est un ensemble dont les éléments sont appelés types ou sortes ;*
- F *est un ensemble dont les éléments sont des noms d'opérations où chaque nom* f *est muni d'une arité dans* $S^* \times S$ *;*
- R *est un ensemble dont les éléments sont des noms de prédicats où chaque nom* r *est muni d'une arité dans* S^+*.*

On note $f : s_1 \times \ldots \times s_n \to s$ *pour une opération* f *d'arité* $(s_1 \ldots s_n, s)$ *et* $r : s_1 \times \ldots \times s_n$ *pour un prédicat* r *d'arité* $s_1 \ldots s_n$*.*

Exemple 1.1.1 Pour illustrer cette définition, on donne la signature des listes d'entiers naturels, que l'on note $\Sigma_{liste\_nat} = (S, F, R)$ où :

$S = \{nat, liste\}$

$F = \{0 : \to nat,$
$\quad succ : nat \to nat,$
$\quad vide : \to liste,$
$\quad cons : nat \times liste \to liste,$
$\quad queue : liste \to liste,$
$\quad @ : liste \times liste \to liste,$
$\quad long : liste \to nat\}$

$$R = \{= : nat \times nat,$$
$$\equiv : liste \times liste,$$
$$\in : nat \times liste,$$
$$estvide : liste\}$$

Termes. À partir de la signature, on va maintenant construire les termes et les formules. Les termes sont construits inductivement à partir des noms d'opérations et d'un ensemble de variables sur la signature, lui-même défini à partir de la notion de S-ensemble :

Définition 1.1.2 (S-ensemble) *Soit un ensemble S. Un S-*ensemble *A est un ensemble muni d'une partition indexée par S : $A = \coprod_{s \in S} A_s$.*

Définition 1.1.3 (Ensemble de variables) *Soit $\Sigma = (S, F, R)$ une signature. On appelle* ensemble de variables *sur Σ un S-ensemble V tel que pour tout $s \in S$, $V_s \cap (S \cup F \cup R) = \emptyset$.*

Exemple 1.1.2 Un ensemble de variables sur Σ_{liste_nat} est, par exemple :

$$V = V_{nat} \amalg V_{liste} = \{a, b, c\} \cup \{l, l'\}$$

On peut maintenant construire l'ensemble des termes avec variables sur une signature donnée.

Définition 1.1.4 (Termes) *Soit $\Sigma = (S, F, R)$ une signature. Soit V un ensemble de variables sur Σ. Le S-ensemble des* termes avec variables*, noté $T_\Sigma(V)$, est le plus petit ensemble tel que :*

- *pour tout $s \in S$, si $x \in V_s$, alors $x \in T_\Sigma(V)_s$;*
- *si $f : s_1 \times \ldots \times s_n \to s \in F$ et $(t_1, \ldots, t_n) \in T_\Sigma(V)_{s_1} \times \ldots \times T_\Sigma(V)_{s_n}$, alors $f(t_1, \ldots, t_n) \in T_\Sigma(V)_s$.*

Formules. On dispose maintenant de tous les éléments syntaxiques pour définir les formules de notre formalisme. On définit l'ensemble des formules inductivement à partir des termes définis précédemment, des noms de prédicats, des connecteurs et quantificateurs logiques usuels et des deux symboles $\top$ et $\bot$ qui dénotent respectivement la formule toujours vraie et celle toujours fausse.

Définition 1.1.5 (Formules) *Soit $\Sigma = (S, F, R)$ une signature. Soit V un ensemble de variables sur Σ. L'ensemble des* formules *sur Σ, noté $Sen(\Sigma)$, est le plus petit ensemble tel que :*

- *$\top, \bot \in Sen(\Sigma)$;*
- *si $r : s_1 \times \ldots \times s_n \in R$ et $(t_1, \ldots, t_n) \in T_\Sigma(V)_{s_1} \times \ldots \times T_\Sigma(V)_{s_n}$, alors $r(t_1, \ldots, t_n) \in Sen(\Sigma)$;*

- *si* $\varphi \in Sen(\Sigma)$, *alors* $\neg\varphi \in Sen(\Sigma)$;
- *si* $x \in V$ *et* $\varphi \in Sen(\Sigma)$, *alors* $\forall x\varphi, \exists x\varphi \in Sen(\Sigma)$;
- *si* $\varphi, \psi \in Sen(\Sigma)$, *alors* $\varphi \wedge \psi, \varphi \vee \psi, \varphi \Rightarrow \psi \in Sen(\Sigma)$.

Exemple 1.1.3 On donne quelques exemples de formules bien formées sur Σ_{liste_nat}. On note $\in$, @ et les relations binaires de façon infixe pour plus de lisibilité.

$queue(cons(b, l)) \equiv l$
$long(vide) = 0$
$long(cons(c, l)) = succ(long(l))$
$estvide(l) \Rightarrow (\forall a\ \neg(a \in l))$
$\forall l'(a \in l \Rightarrow a \in l@l')$
$\neg(a \in cons(b, l)) \Rightarrow \neg(a = b)$

1.2 Sémantique

On donne maintenant un sens mathématique à chacune des constructions symboliques que l'on vient de définir, c'est-à-dire les termes et les formules. On commence par donner un sens aux éléments de base à partir desquels ces deux notions sont construites, c'est-à-dire les éléments de la signature.

Modèles. Un modèle associé à une signature associe à chaque sorte du langage une structure algébrique, c'est-à-dire un ensemble muni de lois internes et externes et d'éléments distingués. Les lois internes et externes vont donner un sens mathématique aux noms de fonctions, tandis que les éléments distingués interprèteront les constantes. De plus, on munit ces structures algébriques de relations n-aires pour interpréter les prédicats. On parle alors de structure du premier ordre.

Définition 1.2.1 (Σ-modèle) *Soit* $\Sigma = (S, F, R)$ *une signature. Un* Σ-modèle *associé à* Σ *est un* S*-ensemble* $\mathcal{M}$ *muni pour chaque nom d'opération* $f : s_1 \times \ldots \times s_n \to s \in F$ *d'une application* $f_{\mathcal{M}} : \mathcal{M}_{s_1} \times \ldots \times \mathcal{M}_{s_n} \to \mathcal{M}_s$, *et pour chaque nom de prédicat* $r : s_1 \times \ldots \times s_n \in R$ *d'une relation* n*-aire* $r_{\mathcal{M}} \subseteq \mathcal{M}_{s_1} \times \ldots \times \mathcal{M}_{s_n}$.

Exemple 1.2.1 Un modèle associé à Σ_{liste_nat} est, par exemple, l'ensemble

$$\mathcal{M} = \mathcal{M}_{nat} \amalg \mathcal{M}_{liste} = \mathbb{N} \cup \mathbb{N}^*$$

muni des applications :

$$\begin{array}{rcl} 0_{\mathcal{M}} : & \to & \mathbb{N} \\ & \mapsto & 0_{\mathbb{N}} \end{array}$$

$$\begin{array}{rcl} succ_{\mathcal{M}} : \mathbb{N} & \to & \mathbb{N} \\ n & \mapsto & n +_{\mathbb{N}} 1_{\mathbb{N}} \end{array}$$

$$\begin{array}{rcl} vide_{\mathcal{M}} : & \to & \mathbb{N}^* \\ & \mapsto & \varepsilon \end{array}$$

$$
\begin{array}{rcl}
cons_{\mathcal{M}} : \mathbb{N} \times \mathbb{N}^* & \rightarrow & \mathbb{N}^* \\
(n, \alpha) & \mapsto & n.\alpha \\
queue_{\mathcal{M}} : \mathbb{N}^* & \rightarrow & \mathbb{N}^* \\
\varepsilon & \mapsto & \varepsilon \\
n.\alpha & \mapsto & \alpha \\
@_{\mathcal{M}} : \mathbb{N}^* \times \mathbb{N}^* & \rightarrow & \mathbb{N}^* \\
(\alpha, \alpha') & \mapsto & \alpha.\alpha' \\
long_{\mathcal{M}} : \mathbb{N}^* & \rightarrow & \mathbb{N} \\
\alpha & \mapsto & |\alpha|
\end{array}
$$

(où « . » et $|\_|$ désignent respectivement le produit de concaténation et la longueur d'un mot du monoïde libre $(\mathbb{N}^*, ., \varepsilon)$ des mots sur $\mathbb{N}$.)

et des relations :

$$
\begin{array}{l}
=_{\mathcal{M}} = =_{\mathbb{N}} \\
\equiv_{\mathcal{M}} = \{(n.\alpha, m.\alpha') \in \mathbb{N}^* \times \mathbb{N}^* \mid n =_{\mathcal{M}} m \wedge \alpha' \equiv_{\mathcal{M}} \alpha'\} \cup \{(\varepsilon, \varepsilon)\} \\
\in_{\mathcal{M}} = \{(n, \alpha) \in \mathbb{N} \times \mathbb{N}^* \mid \exists \alpha', \alpha'' \in \mathbb{N}^*, \alpha \equiv_{\mathcal{M}} \alpha'.n.\alpha''\} \\
estvide_{\mathcal{M}} = \{\varepsilon\}
\end{array}
$$

Interprétation des termes. À partir d'un modèle associé à une signature, on peut maintenant donner un sens dans ce modèle aux termes construits sur cette signature. Comme les termes sont construits inductivement sur les variables et les noms de fonctions, il faut tout d'abord donner un sens aux variables, ce qui se fait au travers d'une application appelée interprétation des variables. On étend ensuite canoniquement cette application aux termes.

Définition 1.2.2 (Interprétation des termes) *Soit $\Sigma = (S, F, R)$ une signature. Soit V un ensemble de variables sur Σ. Soit $\mathcal{M}$ un Σ-modèle. Une* interprétation des variables *est une application $\nu : V \rightarrow \mathcal{M}$, telle que pour tout $s \in S$, pour tout $x \in V_s$, $\nu(x) \in \mathcal{M}_s$. On prolonge toute interprétation des variables ν en une* interprétation des termes *$\nu^\natural : T_\Sigma(V) \rightarrow \mathcal{M}$ de la manière inductive suivante :*

- *si $x \in V$, alors $\nu^\natural(x) = \nu(x)$;*
- *si $f : s_1 \times \ldots \times s_n \rightarrow s \in F$ et $(t_1, \ldots, t_n) \in T_\Sigma(V)_{s_1} \times \ldots \times T_\Sigma(V)_{s_n}$, alors $\nu^\natural(f(t_1, \ldots, t_n)) = f_{\mathcal{M}}(\nu^\natural(t_1), \ldots, \nu^\natural(t_n))$.*

Par abus de notation, on notera également ν le prolongement de l'interprétation des variables aux termes.

Satisfaction des formules. On veut maintenant donner un sens aux formules de notre formalisme, c'est-à-dire être capable de dire si une formule est vraie ou non dans un modèle donné. Pour cela, on étend l'interprétation des termes définie précédemment en un prédicat unaire sur les formules, noté $\mathcal{M} \vDash_\nu$, qui exprime la satisfaction d'une formule par le modèle $\mathcal{M}$ pour une interprétation des variables ν donnée.

Définition 1.2.3 (Satisfaction des formules) *Soit $\Sigma = (S, F, R)$ une signature. Soit V un ensemble de variables sur Σ. Soit $\mathcal{M}$ un Σ-modèle. Soit $\nu : V \to \mathcal{M}$ une interprétation des variables. Soit $\varphi \in Sen(\Sigma)$. La* satisfaction de φ par le modèle $\mathcal{M}$ pour l'interprétation ν, *notée $\mathcal{M} \models_\nu \varphi$, est définie sur la structure de φ de la manière suivante :*

- *si $\varphi = \top$, alors $\mathcal{M} \models_\nu \varphi$;*
- *si $\varphi = \bot$, alors*[1] *$\mathcal{M} \not\models_\nu \varphi$;*
- *si φ est de la forme $r(t_1, \ldots, t_n)$, alors $\mathcal{M} \models_\nu \varphi$ ssi $(\nu(t_1), \ldots, \nu(t_n)) \in r_\mathcal{M}$;*
- *si φ est de la forme $\neg\psi$, alors $\mathcal{M} \models_\nu \varphi$ ssi $\mathcal{M} \not\models_\nu \psi$;*
- *si φ est de la forme $\forall x\psi$, alors $\mathcal{M} \models_\nu \varphi$ ssi pour toute interprétation $\nu' : V \to \mathcal{M}$ qui vérifie $\nu'(y) = \nu(y)$ pour tout $y \in V \setminus \{x\}$, $\mathcal{M} \models_{\nu'} \psi$;*
- *si φ est de la forme $\exists x\psi$, alors $\mathcal{M} \models_\nu \varphi$ ssi il existe une interprétation $\nu' : V \to \mathcal{M}$ qui vérifie $\nu'(y) = \nu(y)$ pour tout $y \in V \setminus \{x\}$, telle que $\mathcal{M} \models_{\nu'} \psi$;*
- *si φ est de la forme $\psi \wedge \chi$, alors $\mathcal{M} \models_\nu \varphi$ ssi $\mathcal{M} \models_\nu \psi$ et $\mathcal{M} \models_\nu \chi$;*
- *si φ est de la forme $\psi \vee \chi$, alors $\mathcal{M} \models_\nu \varphi$ ssi $\mathcal{M} \models_\nu \psi$ ou $\mathcal{M} \models_\nu \chi$;*
- *si φ est de la forme $\psi \Rightarrow \chi$, alors $\mathcal{M} \models_\nu \varphi$ ssi, si $\mathcal{M} \models_\nu \psi$, alors $\mathcal{M} \models_\nu \chi$.*

On dit que $\mathcal{M}$ satisfait φ, noté $\mathcal{M} \models \varphi$, si et seulement si $\mathcal{M} \models_\nu \varphi$ pour tout $\nu : V \to \mathcal{M}$.

Exemple 1.2.2 • On va vérifier que le modèle $\mathcal{M}$ défini précédemment satisfait la formule φ suivante :

$$queue(cons(b, l)) \equiv l$$

Soit une interprétation des variables $\nu : V \to \mathcal{M}$ telle que $\nu(b) = n$ et $\nu(l) = \alpha$, où $n \in \mathbb{N}$ et $\alpha \in \mathbb{N}^*$. Montrer que $\mathcal{M} \models_\nu queue(cons(b, l)) \equiv l$ est équivalent à montrer que $(\nu(queue(cons(b, l))), \nu(l)) \in \equiv_\mathcal{M}$. Or, d'une part,

$$\begin{aligned} \nu(queue(cons(b, l))) &= queue_\mathcal{M}(cons_\mathcal{M}(n, \alpha)) \\ &= queue_\mathcal{M}(n.\alpha) \\ &= \alpha \end{aligned}$$

d'autre part,

$$\nu(l) = \alpha$$

et $(\alpha, \alpha) \in \equiv_\mathcal{M}$ donc $\mathcal{M}$ satisfait bien φ pour ν, pour tous $n \in \mathbb{N}$ et $\alpha \in \mathbb{N}^*$. D'où $\mathcal{M}$ satisfait φ.

• On va vérifier que $\mathcal{M}$ satisfait la formule ψ suivante :

$$estvide(l) \Rightarrow (\forall a\ \neg(a \in l))$$

Soit une interprétation des variables $\nu : V \to \mathcal{M}$ telle que $\nu(a) = m$ et $\nu(l) = \alpha$, où $m \in \mathbb{N}$ et $\alpha \in \mathbb{N}^*$. Montrer que

[1] $\mathcal{M} \not\models_\nu \varphi$ est une notation abrégée pour « il n'est pas vrai que $\mathcal{M} \models_\nu \varphi$

$$\mathcal{M} \vDash_\nu estvide(l) \Rightarrow (\forall a\ \neg(a \in l))$$

est équivalent à montrer que

$$\text{si } \mathcal{M} \vDash_\nu estvide(l) \text{ alors } \mathcal{M} \vDash_\nu \forall a\ \neg(a \in l)$$

Supposons que $\mathcal{M} \vDash_\nu estvide(l)$, c'est-à-dire qu'on a $estvide_\mathcal{M}(\alpha)$, i.e. $\alpha = \varepsilon$. Il faut maintenant montrer que

$$\mathcal{M} \vDash_\nu \forall a\ \neg(a \in l)$$

ce qui est équivalent à montrer que

$$\forall \nu' : V \to \mathcal{M}\ tq\ \forall y \in V \setminus \{a\}, \nu'(y) = \nu(y), \mathcal{M} \vDash_{\nu'} \neg(a \in l)$$

c'est-à-dire $\mathcal{M} \nvDash_{\nu'} a \in l$. Soit une telle interprétation ν' telle que $\nu'(a) = m'$, où $m' \in \mathbb{N}$. Montrer que

$$\mathcal{M} \nvDash_{\nu'} a \in l$$

est équivalent à montrer que

$$(\nu(a), \nu(l)) \notin \in_\mathcal{M}$$

c'est-à-dire $(m', \varepsilon) \notin \in_\mathcal{M}$. Or il n'existe pas $\alpha', \alpha'' \in \mathbb{N}^*$ tels que $\varepsilon = \alpha'.m'.\alpha''$, donc $(m', \varepsilon) \notin \in_\mathcal{M}$. Donc $\mathcal{M}$ satisfait bien ψ pour ν, pour tous $m \in \mathbb{N}$ et $\alpha \in \mathbb{N}^*$. D'où $\mathcal{M}$ satisfait ψ.

Chapitre 2

Systèmes de transitions étiquetées étendus (EIOLTS)

2.1 Syntaxe

Un système de transitions étiquetées étendu, qu'on notera par la suite EIOLTS (pour Extended Input Output Labelled Transition System) est un automate qui est utilisé pour la spécification de systèmes réactifs, c'est-à-dire de systèmes qui interagissent avec leur environnement, lui-même représenté par d'autres EIOLTS. Un système réactif est alors spécifié par un ensemble d'EIOLTS communicant entre eux. Ces communications consistent à envoyer ou recevoir des messages via des canaux de communication. L'automate décrit également le comportement interne du module, c'est-à-dire les évolutions possibles de son état au cours du temps. Les états sont ici abstraits par ce qu'on appellera des points de contrôle, et les évolutions élémentaires du système sont abstraites par une relation de transition entre les points de contrôle. Chaque transition entre deux états est étiquetée par les actions élémentaires qui vont faire passer le système d'un état à l'autre. Ces actions élémentaires sont des actions de communication (envoi ou réception de message) ou des actions internes au système. Les messages envoyés ou reçus et les actions internes sont exprimés à l'aide d'un langage du premier ordre. On regroupe les éléments dont on a besoin pour décrire un EIOLTS sous le nom d'EIOLTS-signature, définie comme suit :

Définition 2.1.1 (EIOLTS-signature) *On appelle* EIOLTS-signature *un triplet* $\mathscr{L}_{\mathcal{C}} = (\Sigma, V, \mathcal{C})$ *où :*

- Σ *est une signature du premier ordre* (*définition* 1.1.1) *;*
- V *un ensemble de variables sur* Σ (*définition* 1.1.3) *;*
- $\mathcal{C}$ *est un ensemble dont les éléments sont appelés* noms de canaux.

Le langage du premier ordre $\mathscr{L}_{\mathcal{C}} = (\Sigma, V)$ va nous permettre de décrire les données de l'EIOLTS (messages envoyés ou reçus et actions internes), tandis que les noms de canaux de l'ensemble $\mathcal{C}$ vont nous permettre de décrire les communications.

Exemple 2.1.1 On va prendre comme fil conducteur l'exemple d'un distributeur automatique de billets, donc la spécification informelle est la suivante.

Un utilisateur introduit sa carte bancaire. Le distributeur initialise un compteur, qui va compter le nombre de fois où le code va être saisi. Puis il demande à l'utilisateur de saisir son code, celui-ci a droit à trois essais consécutifs. Le distributeur vérifie la validité du code. Si le code est erroné, le compteur est incrémenté et le distributeur demande à l'utilisateur de saisir son code de nouveau, sauf si le compteur est à 3, auquel cas le distributeur ne rend pas la carte et revient dans son état initial. Lorsque le code saisi est valide, le distributeur demande à l'utilisateur le montant qu'il veut retirer, puis lui donne une autorisation en fonction du montant et du numéro de carte. Si la date de validité de la carte est dépassée, l'utilisateur n'obtient pas l'argent demandé et ne récupère pas sa carte. Si la carte est valable mais que l'utilisateur n'a pas l'argent qu'il demande sur son compte, il récupère sa carte mais pas l'argent demandé. Enfin, dans le cas où la carte est valable et où l'utilisateur possède effectivement le montant demandé sur son compte, l'utilisateur récupère sa carte d'abord et obtient ses billets ensuite. Dans tous les cas, l'application de l'autorisation termine l'opération et remet le distributeur dans son état initial.

On décrit ici chacun des ensembles de l'EIOLTS-signature $\mathscr{L}_{\mathcal{C}_{DAB}} = (\Sigma, V, \mathcal{C})$, où $\Sigma = (S, F, R)$, qui va permettre de décrire un tel système :

$$S = \{int, bool\}$$

$$F = \{+ : int \times int \to int,\\ estvalide : int \times int \to bool,\\ autorisation : int \times int \to int\}$$

$$R = \{=_b : bool \times bool,\\ =_i : int \times int,\\ < : int \times int\}$$

$$V = V_{int} \amalg V_{bool} = \{C, compt, code, M, a\} \cup \{b\}$$

$$\mathcal{C} = \{Carte, Code, Montant, Billets\}$$

La fonction $estvalide$ détermine, en fonction du numéro de carte et du code saisi par l'utilisateur, si le code est valide ou non. La fonction $autorisation$ donne, en fonction du numéro de carte et du montant demandé, une autorisation à l'utilisateur : 0 si la carte n'est pas valable, 1 si la carte est valable mais le montant trop élevé, 2 si la carte est valable et le montant disponible.

On dispose maintenant de tout le vocabulaire nécessaire pour décrire les actions élémentaires qui étiquètent les transitions. Une transition est étiquetée par trois éléments :

1. une *action de communication* (envoi ou réception de message) ;
2. une condition de franchissement de la transition, appelée *garde*, qui n'est vérifiée qu'après l'action de communication ;

3. une action strictement interne, appelée *substitution des variables*, qui modifie l'état des variables du système.

On va d'abord introduire chacun de ces éléments avant de donner la définition d'un EIOLTS.

Action de communication. Une action de communication est un envoi ou une réception de message par un canal de communication $c \in \mathcal{C}$, où le message est un terme t construit sur le langage du premier ordre $\mathscr{L}$, c'est-à-dire $t \in T_\Sigma(V)$ (définition 1.1.4). La définition formelle est la suivante :

Définition 2.1.2 (Actions de communication) *L'ensemble* $Act_{\mathscr{L}_\mathcal{C}}$ *des* actions de communication *sur* $\mathscr{L}_\mathcal{C}$ *est défini par :*

$$Act_{\mathscr{L}_\mathcal{C}} := \tau \mid c?x \mid c!t$$

où $c \in \mathcal{C}$, $x \in V$ *et* $t \in T_\Sigma(V)$.

Intuitivement, le terme $c?x$ dénote la réception par le canal c d'une valeur qui est affectée à la variable x, le terme $c!t$ dénote l'émission du terme t par le canal c, et τ indique que le passage de cette transition n'implique pas de communication avec le reste du système.

Garde. Après l'envoi ou la réception d'un message, une condition doit être vérifiée pour que le franchissement de la transition s'exécute. Cette condition, appelée garde, est exprimée à l'aide d'une formule φ construite sur le langage du premier ordre $\mathscr{L}$, c'est-à-dire $\varphi \in Sen(\Sigma)$ (définition 1.1.5).

Substitution des variables. La substitution des variables va faire évoluer le système de façon interne en attribuant de nouvelles valeurs à une partie des variables. C'est ceci qui décrit l'état du système. Plus formellement :

Définition 2.1.3 (Substitution de variables) *Une* substitution des variables *est une application* $\delta : V \to T_\Sigma(V)$ *qui conserve les sortes, c'est-à-dire telle que pour tout* $s \in S$, $\delta(V_s) \subseteq T_\Sigma(V)_s$. *On prolonge toute substitution des variables* δ *en une substitution des termes* $\delta^\natural : T_\Sigma(V) \to T_\Sigma(V)$ *de la manière inductive suivante :*

- *si* $x \in V$, *alors* $\delta^\natural(x) = \delta(x)$;
- *si* $f : s_1 \times \ldots \times s_n \to s \in F$ *et* $(t_1, \ldots, t_n) \in T_\Sigma(V)_{s_1} \times \ldots \times T_\Sigma(V)_{s_n}$, *alors* $\delta^\natural(f(t_1, \ldots, t_n)) = f(\delta^\natural(t_1), \ldots, \delta^\natural(t_n))$.

On note $T_\Sigma(V)^V$ *l'ensemble des applications de* V *dans* $T_\Sigma(V)$.

On peut maintenant définir un EIOLTS sur $\mathscr{L}_\mathcal{C}$, dont l'ensemble des points de contrôle est désigné par $\mathbb{Q}$ et dont les transitions sont étiquetées par les trois éléments définis précédemment : une action de communication *act* de $Act_{\mathscr{L}_\mathcal{C}}$, une garde φ de $Sen(\Sigma)$, et une substitution des variables δ de $T_\Sigma(V)^V$.

Définition 2.1.4 ($\mathscr{L}_\mathcal{C}$-EIOLTS) *Un* $\mathscr{L}_\mathcal{C}$-EIOLTS *est un triplet* $(\mathbb{Q}, q_0, \mathbb{T})$ *où :*

– $\mathbb{Q}$ *est un ensemble dont les éléments sont appelés* points de contrôle *;*
– $q_0 \in \mathbb{Q}$ *est appelé point de contrôle initial ;*
– $\mathbb{T} \subseteq \mathbb{Q} \times Act_{\mathscr{L}_C} \times Sen(\Sigma) \times T_\Sigma(V)^V \times \mathbb{Q}$ *est une relation telle que si l'on note* $\mathbb{T}_\mathbb{Q}$ *la projection de* $\mathbb{T}$ *sur* $\mathbb{Q} \times \mathbb{Q}$ *et* $\mathbb{T}_\mathbb{Q}^+$ *la fermeture transitive de* $\mathbb{T}_\mathbb{Q}$ *alors pour tout* $q \in \mathbb{Q} \setminus \{q_0\}$, $(q_0, q) \in \mathbb{T}_\mathbb{Q}^+$.

La relation de transition $\mathbb{T}$ est donc telle que tout point de contrôle de l'EIOLTS soit atteignable à partir de q_0.

Exemple 2.1.2 La figure 2.1 est un EIOLTS construit sur $\mathscr{L}_{\mathcal{C}_{DAB}}$ qui modélise le distributeur de billets décrit précédemment. On note cet EIOLTS $\mathbb{G}_{DAB}$.

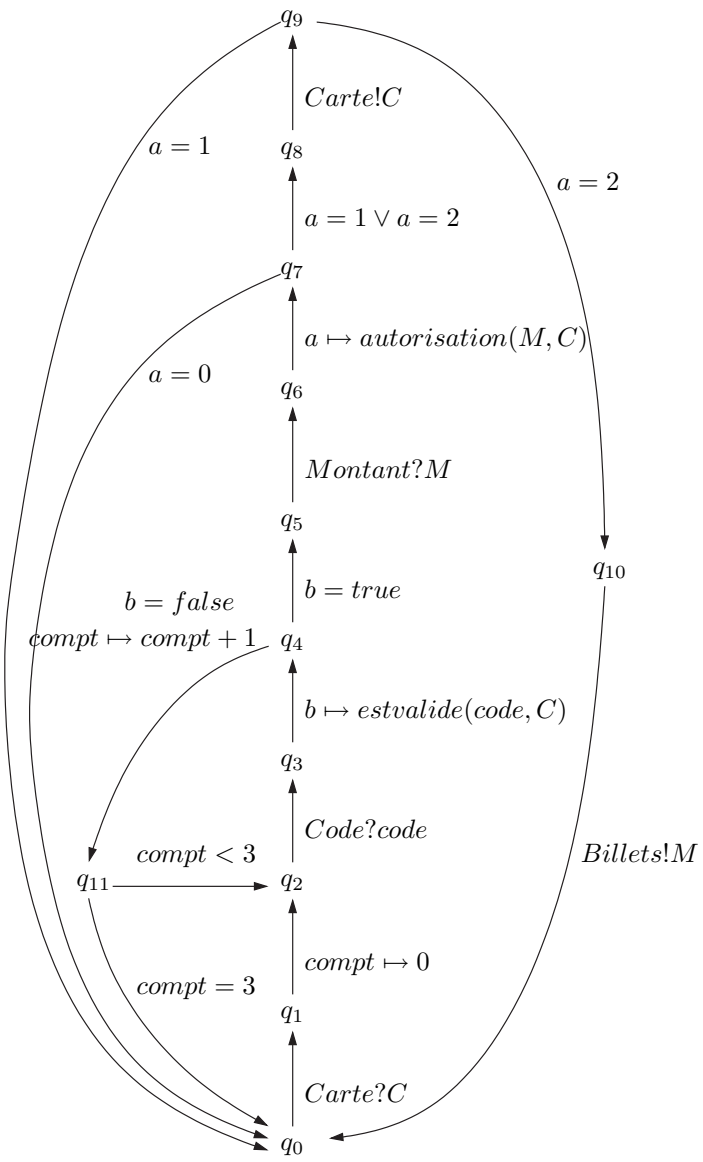


FIG. 2.1 – Un distributeur automatique de billets

Définition 2.1.5 (Source et cible d'une transition) *On définit les applications* $source : \mathbb{T} \to \mathbb{Q}$ *et* $cible : \mathbb{T} \to \mathbb{Q}$ *telles que pour tout* $t = (q, act, \varphi, \delta, q') \in \mathbb{T}$, $source(t) = q$ *et* $cible(t) = q'$.

Un EIOLTS spécifie l'ensemble des comportements d'un système. Ces comportements sont des suites d'actions élémentaires, représentées dans l'EIOLTS par les transitions. Un comportement du système est donc représenté par une suite de transitions : c'est la notion de chemin.

Définition 2.1.6 (Chemin) *Soit* $\mathbb{G} = (\mathbb{Q}, q_0, \mathbb{T})$ *un* $\mathscr{L}_{\mathcal{C}}$*-EIOLTS. Un* chemin *dans* $\mathbb{G}$ *est un mot* $t_1 \dots t_n$ *de* $\mathbb{T}^*$ *tel que pour tout* $1 \leq j < n$, $cible(t_j) = source(t_{j+1})$. *On note* $Path(\mathbb{G})$ *l'ensemble des chemins de* $\mathbb{G}$.

Rappelons que $\mathbb{T}^*$ désigne le monoïde libre de l'ensemble des mots finis sur $\mathbb{T}$ muni du produit de concaténation « $\cdot$ » associatif et possédant comme élément neutre le mot vide ε.

Exemple 2.1.3 Les trois suites de transitions de la figure 2.2 sont des chemins de $\mathbb{G}_{DAB}$.

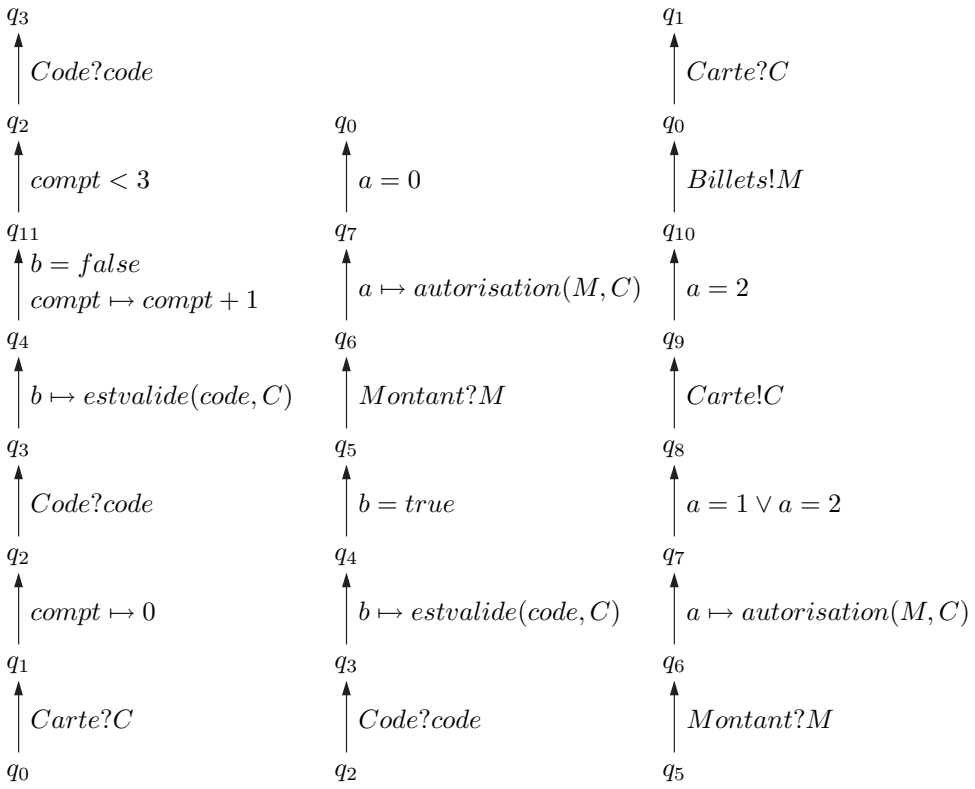


FIG. 2.2 – Chemins de $\mathbb{G}_{DAB}$

Définition 2.1.7 (Source et cible d'un chemin) *On définit* $source^\natural : Path(\mathbb{G}) \setminus \{\varepsilon\} \to \mathbb{Q}$ *et* $cible^\natural : Path(\mathbb{G}) \setminus \{\varepsilon\} \to \mathbb{Q}$ *les extensions canoniques de source et cible de la*

définition 2.1.5 *telles que pour tout* $ch = t_1 \dots t_n \in Path(\mathbb{G})$, $source^\natural(ch) = source(t_1)$ *et* $cible^\natural(ch) = cible(t_n)$. *Par abus de notation, on notera également source et cible ces extensions canoniques.*

2.2 Sémantique

On va tout d'abord donner un sens mathématique aux éléments de $\mathscr{L}_\mathcal{C}$. Un modèle associé à $\mathscr{L}_\mathcal{C}$ va en fait être un modèle associé à la signature du premier ordre Σ de $\mathscr{L}_\mathcal{C}$, c'est-à-dire un Σ-modèle au sens de la définition 1.1.1. C'est dans ce modèle qu'on va interpréter les données de l'EIOLTS. On ne donne pas aux noms de canaux de contrepartie sémantique. En effet, les noms de canaux ne servent qu'à différencier les canaux, ils seront donc représentés au niveau concret par des identificateurs qui leur seront univoquement associés (deux noms de canaux différents désignent deux canaux réels différents) et qui ne donnent pas lieu à une interprétation. On se donne donc ici un $\mathscr{L}_\mathcal{C}$-EIOLTS $\mathbb{G} = (\mathbb{Q}, q_0, \mathbb{T})$ et un Σ-modèle $\mathcal{M}$.

Un EIOLTS spécifie l'ensemble des comportements d'un système depuis son état initial. Ces comportements sont décrits par les chemins d'origine q_0, la sémantique d'un EIOLTS est donc l'interprétation de tous les chemins d'origine q_0 dans $\mathcal{M}$. Or pour donner un sens à un chemin, il faut tout d'abord donner un sens aux éléments qui le compose, c'est-à-dire les transitions de l'EIOLTS. On va donc commencer par donner une interprétation aux transitions. Une transition est interprétée par une relation entre les états du système, qui sont ici des interprétations des variables, c'est-à-dire des applications de V dans $\mathcal{M}$.

Notation. On note $\mathcal{M}^V$ l'ensemble des applications de V dans $\mathcal{M}$.

Définition 2.2.1 (Sémantique d'une transition) *Soit* $t = (q, act, \varphi, \delta, q') \in \mathbb{T}$ *une transition. La sémantique de* t *est la relation* $Sem_\mathcal{M}(t)$ *sur* $\mathcal{M}^V \times \big((\mathcal{C} \times \{?, !\} \times \mathcal{M}) \cup \{\tau\}\big) \times \mathcal{M}^V$ *telle que* $(\nu^i, e, \nu^f) \in Sem_\mathcal{M}(t)$ *ssi :*

- *si* $act = \tau$, *alors* $\mathcal{M} \vDash_{\nu^i} \varphi$, $e = \tau$ *et* $\nu^f = \nu^{i\natural} \circ \delta$
- *si act est de la forme* $c!t$, *alors* $\mathcal{M} \vDash_{\nu^i} \varphi$, $e = c!\nu^{i\natural}(t)$ *et* $\nu^f = \nu^{i\natural} \circ \delta$
- *si act est de la forme* $c?x$, *alors il existe* $\nu^a \in \mathcal{M}^V$ *définie par* $\nu^a = \nu^i$ *sur* $V \setminus \{x\}$ *et* $c?\nu^a(x) = e$, *telle que* $\mathcal{M} \vDash_{\nu^a} \varphi$ *et* $\nu^f = \nu^{a\natural} \circ \delta$.

On note $Sem_\mathcal{M}(\mathbb{T})$ *l'ensemble* $\bigcup_{t \in \mathbb{T}} Sem_\mathcal{M}(t)$.

L'application ν^i est l'interprétation des variables avant le franchissement de la transition (interprétation *i*nitiale) et ν^f désigne l'interprétation obtenue après le franchissement de la transition (interprétation *f*inale). La valeur e est la valeur du message envoyé ou reçu, munie du nom de canal et du symbole indiquant une émission ou une réception. On donne donc un sens à une transition, d'une part au travers du changement d'état du système qu'elle induit (niveau interne du module), et d'autre part au travers de la valeur de l'entrée ou de la sortie qu'elle produit (niveau global du système). Cette valeur

nous sera utile par la suite, pour établir la correction du raffinement d'une transition (cf sous-section 2.3.2).

Exemple 2.2.1 On va expliciter la sémantique de quelques transitions de $\mathbb{G}_{DAB}$. On donne tout d'abord un Σ-modèle associé à $\mathscr{L}_{\mathcal{C}_{DAB}}$, dans lequel on donne une interprétation des éléments de la signature du premier ordre contenue dans $\mathscr{L}_{\mathcal{C}_{DAB}}$:

$$\mathcal{M} = \mathcal{M}_{bool} \amalg \mathcal{M}_{int} = \{\mathbf{vrai}, \mathbf{faux}\} \cup \mathbb{N}$$

muni des applications :

$$+_{\mathcal{M}} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}, \quad (n, m) \mapsto n +_{\mathbb{N}} m$$

$$estvalide_{\mathcal{M}} : \mathbb{N} \times \mathbb{N} \to \mathbb{B}$$

$$(n, m) \mapsto \begin{cases} \mathbf{vrai} & \text{si } n \text{ est le code de la carte } m \\ \mathbf{faux} & \text{si } n \text{ n'est pas le bon code} \end{cases}$$

$$autorisation_{\mathcal{M}} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

$$(n, m) \mapsto \begin{cases} 0 & \text{si la carte } m \text{ n'est pas valable} \\ 1 & \text{si la carte } m \text{ est valable mais} \\ & \text{la somme } n \text{ non disponible} \\ 2 & \text{si la carte } m \text{ est valable et} \\ & \text{la somme } n \text{ disponible} \end{cases}$$

et des relations :

$$=_{b_{\mathcal{M}}} = \{(\mathbf{vrai}, \mathbf{vrai}), (\mathbf{faux}, \mathbf{faux})\}$$

$$=_{i_{\mathcal{M}}} = =_{\mathbb{N}}$$

$$<_{\mathcal{M}} = <_{\mathbb{N}}$$

• On considère la transition $t_1 = (q_2, Code?code, \top, id, q_3)$, où id désigne l'identité sur les variables, qui représente la réception du code par le distributeur :

$$q_2 \xrightarrow{Code?code} q_3$$

La sémantique de t_1 est l'ensemble des triplets (ν^i, e, ν^f) tels qu'il existe ν^a définie par $\nu^a = \nu^i$ sur $V \setminus \{code\}$ et $Code?\nu^a(code) = e$, telle que $\mathcal{M} \vDash_{\nu^a} \top$, ce qui est toujours vrai, et $\nu^f = \nu^{a\natural} \circ id = \nu^a$.

Par exemple, si l'utilisateur saisit le code 5469, alors tous les triplets $(\nu^i, Code?5469, \nu^f)$ tels que $\nu^i = \nu^f$ sur $V \setminus \{code\}$ et $\nu^f(code) = 5469$ appartiennent à la sémantique de t_1.

• On considère la transition $t_2 = (q_3, \tau, \top, b \mapsto estvalide(code, C), q_4)$ qui représente le test de validité du code saisi par l'utilisateur :

$$q_3 \xrightarrow{b \mapsto estvalide(code, C)} q_4$$

La sémantique de t_2 est l'ensemble des triplets (ν^i, e, ν^f) tels que :

– $\mathcal{M} \vDash_{\nu^i} \top$, ce qui est toujours vrai

- $e = \tau$
- $\nu^f = \nu^{i\sharp} \circ \delta$ où $\delta = id$ sur $V \setminus \{b\}$ et $\delta(b) = estvalide(code, C)$. On a donc $\nu^f = \nu^i$ sur $V \setminus \{b\}$, et $\nu^f(b) = \nu^{i\sharp}(estvalide(code, C)) = estvalide_{\mathcal{M}}(\nu^i(code), \nu^i(C))$.

Par exemple, si le code n'est pas valide, tous les triplets (ν^i, τ, ν^f) tels que $\nu^f = \nu^i$ sur $V \setminus \{b\}$ et $\nu^f(b) = \textbf{faux}$ appartiennent à la sémantique de t_2.

• On considère la transition $t_3 = (q_4, \tau, b = false, compt \mapsto compt + 1, q_{11})$ qui représente l'incrémentation du compteur si le code est erroné :

$$q_4 \xrightarrow[compt \mapsto compt + 1]{b = false} q_{11}$$

La sémantique de t_3 est l'ensemble des triplets (ν^i, e, ν^f) tels que :

- $\mathcal{M} \vDash_{\nu^i} b = false$, c'est-à-dire $\nu^i(b) = \nu^{i\sharp}(false) = \textbf{faux}$
- $e = \tau$
- $\nu^f = \nu^{i\sharp} \circ \delta$ où $\delta = id$ sur $V \setminus \{compt\}$ et $\delta(compt) = compt + 1$. On a donc $\nu^f = \nu^i$ sur $V \setminus \{compt\}$, et $\nu^f(compt) = \nu^{i\sharp}(compt + 1) = \nu^i(compt) +_{\mathbb{N}} 1$.

Par exemple, si le compteur était à 0 et que le code n'est pas valide, c'est-à-dire si ν^i est tel que $\nu^i(compt) = 0$ et $\nu^i(b) = \textbf{faux}$, alors tous les triplets (ν^i, τ, ν^f) tels que $\nu^f = \nu^i$ sur $V \setminus \{compt\}$ et $\nu^f(compt) = 1$ appartiennent à la sémantique de t_3.

On peut maintenant donner une sémantique aux chemins. Comme les chemins sont des mots sur l'ensemble des transitions, la sémantique d'un chemin va être un mot sur l'ensemble des sémantiques des transitions. On munit l'ensemble des mots sur $Sem_{\mathcal{M}}(\mathbb{T})$ d'un produit de concaténation noté « . », on obtient alors le monoïde que l'on note $Sem_{\mathcal{M}}(\mathbb{T})^*$ à partir duquel on va construire la sémantique des chemins.

Définition 2.2.2 (Sémantique d'un chemin) *Soit $ch = t_1 \dots t_n \in Path(\mathbb{G})$ un chemin. La sémantique de ch, notée $Sem_{\mathcal{M}}(ch)$, est l'ensemble des éléments $s = s_1 \dots s_n$ de $Sem_{\mathcal{M}}(\mathbb{T})^*$ tels que pour tout $1 \leq j \leq n$, $s_(\nu^i_j, e_j, \nu^f_j) \in Sem_{\mathcal{M}}(t_i)$ et pour tout $1 \leq j < n$, $\nu^f_j = \nu^i_{j+1}$.*

Une interprétation d'un chemin est donc la concaténation d'une interprétation de chacune de ses transitions, telle que les interprétations des variables en chaque point de contrôle coïncident.

Exemple 2.2.2 On considère le chemin formé par la concaténation des trois transitions de l'exemple précédent $ch = t_1 t_2 t_3$:

$$q_2 \xrightarrow{Code?code} q_3 \xrightarrow{b \mapsto estvalide(code, C)} q_4 \xrightarrow[compt \mapsto compt + 1]{b = false} q_{11}$$

La sémantique de ch est l'ensemble des mots $s_1 s_2 s_3$ tels que s_1, s_2 et s_3 appartiennent respectivement à la sémantique de t_1,t_2 et t_3, et tels que $\nu^f_1 = \nu^i_2$ et $\nu^f_2 = \nu^i_3$.

Il ne reste plus qu'à prendre l'union des sémantiques de tous les chemins d'origine q_0 pour obtenir la sémantique d'un $\mathscr{L}_{\mathcal{C}}$-EIOLTS. On note $Path_q(\mathbb{G})$ l'ensemble $\{ch \in Path(\mathbb{G}) \mid source(ch) = q\}$.

Définition 2.2.3 (Sémantique d'un $\mathscr{L}_{\mathcal{C}}$-EIOLTS) *La sémantique de $\mathbb{G}$, notée $Sem_{\mathcal{M}}(\mathbb{G})$, est définie par :*

$$Sem_{\mathcal{M}}(\mathbb{G}) = \bigcup_{ch \in Path_{q_0}(\mathbb{G})} Sem_{\mathcal{M}}(ch)$$

2.3 Raffinement

2.3.1 Syntaxe

Un EIOLTS étant une spécification d'un système, on aimerait avoir une notion de raffinement de cette spécification, qui permettrait à chaque étape de raffinement de s'approcher un peu plus de l'implantation réelle du système. On va présenter ici une façon d'effectuer ce raffinement. Celle-ci va consister à ajouter des informations sur le système décrit et sur ses comportements, tout d'abord en élargissant le langage dont on se sert pour les décrire. Si on considère une EIOLTS-signature $\mathscr{L}_{\mathcal{C}_1} = (S_1, F_1, R_1, V_1, \mathcal{C}_1)$ et un $\mathscr{L}_{\mathcal{C}_1}$-EIOLTS noté $\mathbb{G}_1 = (\mathbb{Q}_1, q_{0_1}, \mathbb{T}_1)$, on va parler d'un raffinement de $\mathbb{G}_1$ en considérant une deuxième EIOLTS-signature $\mathscr{L}_{\mathcal{C}_2} = (S_2, F_2, R_2, V_2, \mathcal{C}_2)$ telle que $S_1 \subseteq S_2$, $F_1 \subseteq F_2$, $R_1 \subseteq R_2$, $V_1 \subseteq V_2$, $\mathcal{C}_1 \subseteq \mathcal{C}_2$. On notera $\mathscr{L}_{\mathcal{C}_1} \subseteq \mathscr{L}_{\mathcal{C}_2}$.

Exemple 2.3.1 On prend $\mathscr{L}_{\mathcal{C}_1} = \mathscr{L}_{\mathcal{C}_{DAB}}$ de l'exemple 2.1.1. On ajoute à $\mathscr{L}_{\mathcal{C}_1}$ un nom de prédicat $>$, deux variables T et d et trois noms de canaux $Valide$, $ValiditeDate$ et $TotalCompte$ pour former l'EIOLTS-signature $\mathscr{L}_{\mathcal{C}_2} = (S_2, F_2, R_2, V_2, \mathcal{C}_2)$:

$S_2 = S_1$

$F_2 = F_1$

$R_2 = R_1 \cup \{>: int \times int\}$

$V_2 = V_{2_{int}} \amalg V_{2_{bool}} = (V_{1_{int}} \cup \{T\}) \cup (V_{1_{bool}} \cup \{d\})$

$\mathcal{C}_2 = \mathcal{C}_1 \cup \{Valide, ValiditeDate, TotalCompte\}$

On peut alors expliciter certains comportements du système à l'aide de ce langage plus riche. Le comportement induit par une transition t de $\mathbb{G}_1$ va être détaillé par un EIOLTS $\mathbb{G}_t$ qui prendra la place de la transition dans $\mathbb{G}_1$. Comme on veut que $\mathbb{G}_t$ remplace t, il faut qu'il ait pour point de contrôle initial la source de la transition, et que tous les chemins terminent sur la cible de la transition. De plus, on veut que tous les chemins de $\mathbb{G}_t$ contiennent l'action de communication de t et aucune autre action de communication de la signature "pauvre". On a donc la définition suivante du raffinement d'une transition du point de vue syntaxique :

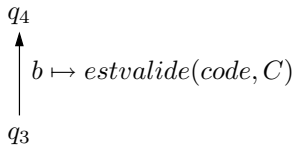
Définition 2.3.1 (Raffinement syntaxique d'une transition) *Soit une transition* $t = (q, act, \varphi, \delta, q') \in \mathbb{T}_1$. *Un* raffinement syntaxique *de* t *est un* $\mathscr{L}_{\mathcal{C}_2}$*-EIOLTS* $\mathbb{G}_t = (\mathbb{Q}_t, q_{0_t}, \mathbb{T}_t)$ *tel que :*

- $\mathbb{Q}_t \cap \mathbb{Q}_1 = \{q, q'\}$
- $q_{0_t} = q$
- *pour tout* $q'' \in \mathbb{Q}_t$*, il existe* $ch \in Path_{q''}(\mathbb{G}_t)$ *tel que* $cible(ch) = q'$
- *pour tout* $ch = t_1 \dots t_n \in Path_q(\mathbb{G}_t)$ *tel que* $cible(ch) = q'$*, il existe un unique* $k \in [\![1, n]\!]$*, tel que l'action de communication de* t_k *est act, et pour tout* $j \in [\![1, n]\!] \setminus \{k\}$*, l'action de communication de* t_j *est* τ *ou utilise un nom de canal de* $\mathcal{C}_2 \setminus \mathcal{C}_1$.

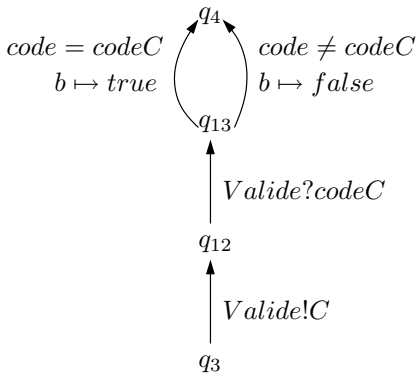
La première condition impose que tous les points de contrôle introduits par le raffinement n'existent pas déjà dans $\mathbb{G}_1$. Les deux suivantes imposent qu'on puisse remplacer t par son raffinement dans $\mathbb{G}_1$, puisque tous les chemins commencent en q et finissent en q'. La dernière condition impose de retrouver l'action de communication de t dans chacun des chemins qui va de q à q', et que toutes les autres actions de communication de ces chemins utilisent des nouveaux noms de canaux.

Exemple 2.3.2 On va raffiner deux transitions de l'EIOLTS $\mathbb{G}_{DAB}$ défini précédemment :

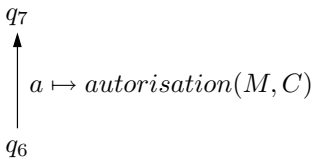
• On considère la transition suivante :



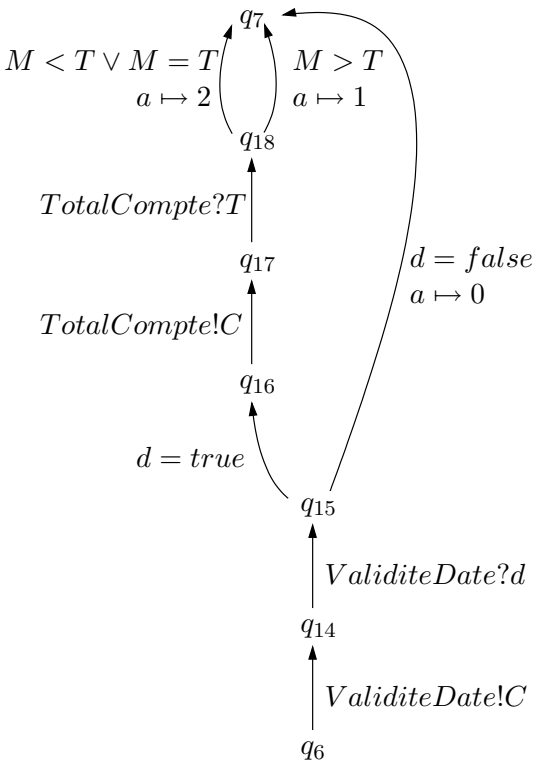
On va raffiner cette transition en explicitant la fonction $estvalide$. On imagine par exemple que pour vérifier la validité du code, le distributeur va rentrer en communication avec la banque dont il dépend. Il va envoyer le numéro de carte de l'utilisateur C à la banque, qui va lui renvoyer le code qui correspond à ce numéro de carte, sur la variable $codeC$. Puis le distributeur va comparer le code obtenu avec le code saisi par le client : s'ils sont égaux, on affecte $true$ à la variable booléenne b, sinon, on lui affecte $false$. L'EIOLTS qui raffine cette transition représente donc ces deux comportements de la manière suivante :



• On raffine également la transition suivante :



De la même manière que pour la transition précédente, on va raffiner en explicitant la fonction $autorisation$. Le distributeur va de nouveau faire appel à la banque pour choisir l'autorisation à donner à l'utilisateur. Tout d'abord, le distributeur va demander à la banque de vérifier la date de validité de la carte. Si la date est dépassée, le distributeur donne l'autorisation 0. Si la date est correcte, le distributeur va demander à la banque de lui donner le montant total qui se trouve sur le compte de l'utilisateur. Il va ensuite le comparer au montant demandé, et donner l'autorisation 1 si l'utilisateur n'a pas l'argent qu'il demande sur son compte, et l'autorisation 2 dans le cas contraire. La transition est donc raffinée par l'EIOLTS suivant :



Remarque. Une transition $t = q, act, \varphi, \delta, q')$ peut être considérée comme un EIOLTS $\mathbb{G}_t = (\mathbb{Q}_t, q_{0_t}, \mathbb{T}_t)$ où $\mathbb{Q}_t = \{q, q'\}$, $q_{0_t} = q$ et $\mathbb{T}_t = \{t\}$. D'après la définition 2.3.1, une transition peut donc se raffiner en elle-même.

On peut maintenant donner la définition du raffinement d'un EIOLTS par un autre. Elle va consister à étendre la définition du raffinement d'une transition à toutes les transitions de l'EIOLTS que l'on veut raffiner.

Définition 2.3.2 (Raffinement syntaxique d'un EIOLTS) *Un* raffinement syntaxique *de* $\mathbb{G}_1$ *est un* $\mathscr{L}_{\mathcal{C}_2}$*-EIOLTS* $\mathbb{G}_2 = (\mathbb{Q}_2, q_{0_2}, \mathbb{T}_2)$ *tel que, si pour tout* $t \in \mathbb{T}_1$*, on note* $\mathbb{G}_t = (\mathbb{Q}_t, q_{0_t}, \mathbb{T}_t)$ *un* $\mathscr{L}_{\mathcal{C}_2}$*-EIOLTS qui raffine syntaxiquement* t*, on a :*

- $\mathbb{Q}_2 = \bigcup_{t \in \mathbb{T}_1} \mathbb{Q}_t$
- $q_{0_2} = q_{0_1}$
- $\mathbb{T}_2 = \bigcup_{t \in \mathbb{T}_1} \mathbb{T}_t$

Un raffinement de $\mathbb{G}_1$ est donc un EIOLTS composé des raffinements de chacune des transitions de $\mathbb{G}_1$.

Remarque. On déduit de la définition 2.3.1 du raffinement d'une transition et de la définition précédente que $\mathbb{Q}_1 \subseteq \mathbb{Q}_2$ et $\mathbb{T}_1 \subseteq \mathbb{T}_2$.

Exemple 2.3.3 On considère l'EIOLTS $\mathbb{G}_{DAB}$ dans lequel on a remplacé les deux transitions raffinées précédemment par leurs raffinements respectifs. On obtient l'EIOLTS de la figure 2.3, qui est donc un raffinement syntaxique de $\mathbb{G}_{DAB}$.

2.3.2 Correction et complétude

Le raffinement d'un EIOLTS doit décrire les mêmes comportements que cet EIOLTS sur le langage partagé. On va donc vouloir, d'une part, que le raffinement ne décrive pas de comportements que ne décrirait pas l'EIOLTS initial : c'est ce qu'on appellera la correction du raffinement. D'autre part, on va vouloir être capable de retrouver tous les comportements décrits par l'EIOLTS initial dans le raffinement : c'est ce qu'on appellera la complétude du raffinement.

On va donc comparer un EIOLTS à l'EIOLTS qu'il raffine, en comparant les comportements que chacun d'eux décrit, modulo le langage de l'EIOLTS initial. Comme un raffinement d'un EIOLTS est un EIOLTS formé d'un raffinement de chacune de ses transitions, on va commencer par comparer chaque transition à l'EIOLTS qui la raffine. Une transition représente un changement d'état du système par une action élémentaire. Lorsqu'on raffine cette transition, on veut que le raffinement fasse évoluer le système de la même façon, c'est-à-dire qu'il induise le même changement d'état. On veut donc être capable de comparer la sémantique de l'EIOLTS qui raffine une transition t à la sémantique de cette transition. On se donne un Σ_2-modèle $\mathcal{M}_2$. On définit alors l'application $\rho_t^{2,1}$ qui va nous permettre de ramener la sémantique de chacun des chemins de l'EIOLTS à un triplet (ν^i, e, ν^f), de manière à pouvoir la comparer à la sémantique de t. Dans ce triplet, ν^i sera l'interprétation à la source du chemin restreinte aux variables communes au chemin et à la transition, c'est-à-dire V_1, ν^f l'interprétation à la cible restreinte à V_1, et e la valeur du message qui correspond à l'action de communication commune au chemin et à la transition. De cette manière, on pourra comparer les changements d'états et la communication induits par une transition et son raffinement. L'indice t désigne la transition par rapport à laquelle on veut restreindre la sémantique

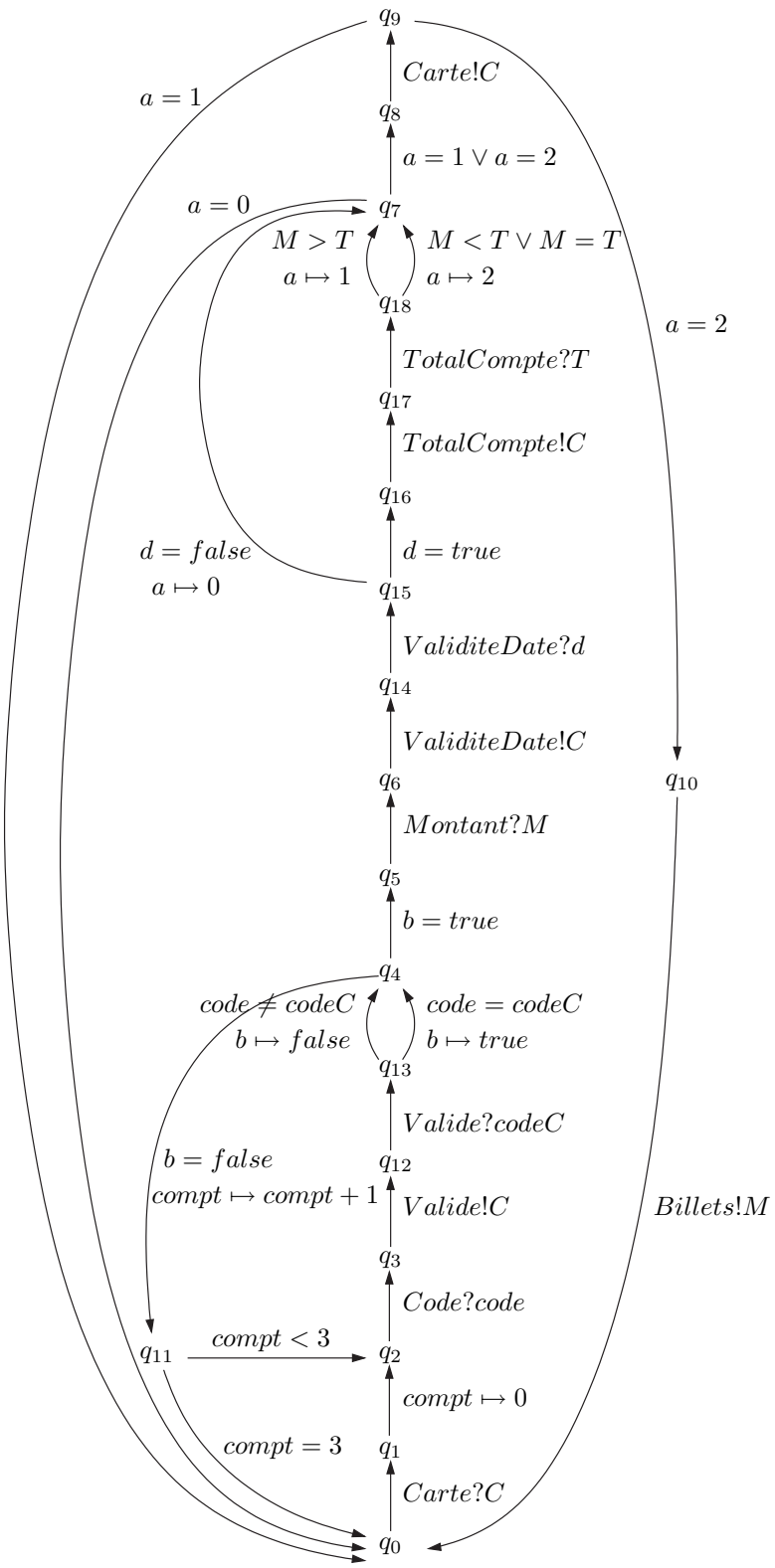


FIG. 2.3 – Un raffinement syntaxique de $\mathbb{G}_{DAB}$

du chemin. L'exposant $2,1$ exprime la restriction de l'interprétation du chemin dans le modèle $\mathcal{M}_2$ à l'oubli de $\mathcal{M}_2$ sur Σ_1, noté $\mathcal{M}_1$ et défini comme suit :

Définition 2.3.3 (Oubli) *L'*oubli *de $\mathcal{M}_2$ sur Σ_1, noté $\mathcal{M}_1$, est l'ensemble $\mathcal{M}_2$ muni, pour tout $f \in F_1$, de l'application $f_{\mathcal{M}_2}$, et pour tout prédicat $r \in R_1$, de la relation $r_{\mathcal{M}_2}$.*

Définition 2.3.4 (Sémantique restreinte) *Soit $t = (q, act, \varphi, \delta, q') \in \mathbb{T}_1$. Soit $\mathbb{G}_t$ un $\mathscr{L}_{\mathcal{C}_2}$-EIOLTS qui raffine t. Soit $ch = t_1 \dots t_n \in Path_q(\mathbb{G}_t)$ tel que $cible(ch) = q'$. Soit $k \in [\![1, n]\!]$ l'indice tel que t_k soit étiquetée par act. On note :*

$$\rho_t^{2,1} : \left(\mathcal{M}_2^{V_2} \times \left((\mathcal{C} \times \{?,!\} \times \mathcal{M}_2) \cup \{\tau\}\right) \times \mathcal{M}_2^{V_2}\right)^* \to \mathcal{M}_1^{V_1} \times \left((\mathcal{C} \times \{?,!\} \times \mathcal{M}_1) \cup \{\tau\}\right) \times \mathcal{M}_1^{V_1}$$

l'application définie pour tout $s = s_1 \dots s_n \in Sem_{\mathcal{M}_2}(ch)$ par :

$$\rho_t^{2,1}(s) = \left(\nu_1^i{}_{|_{V_1}}, e_k, \nu_n^f{}_{|_{V_1}}\right)$$

On note alors :

$$\rho_t^{2,1}(Sem_{\mathcal{M}_2}(\mathbb{G}_t)) = \bigcup_{\substack{ch \in Path_q(\mathbb{G}_t) \\ cible(ch) = q'}} \{\rho_t^{2,1}(s) \mid s \in Sem_{\mathcal{M}_2}(ch)\}$$

Cette restriction nous permet désormais de comparer la sémantique d'une transition à la sémantique de l'EIOLTS qui la raffine.

Pour que le raffinement d'une transition soit correct, on a dit plus haut qu'il fallait qu'il ne décrive pas plus de comportements que la transition elle-même. En d'autres termes, on veut que tout comportement décrit par le raffinement soit un comportement décrit par la transition.

Définition 2.3.5 (Correction du raffinement d'une transition) *Soit $t \in \mathbb{T}_1$ et $\mathbb{G}_t$ un $\mathscr{L}_{\mathcal{C}_2}$-EIOLTS qui raffine t. On dit que $\mathbb{G}_t$ est un* raffinement correct *de t ssi :*

$$\rho_t^{2,1}(Sem_{\mathcal{M}_2}(\mathbb{G}_t)) \subseteq Sem_{\mathcal{M}_1}(t)$$

Exemple 2.3.4 Par exemple, dans la figure 2.4, l'EIOLTS n'est pas un raffinement correct de la transition. En effet, l'application *autorisation* ne peut pas rendre la valeur 3, donc le comportement décrit par ce chemin n'existe pas dans la transition.

Pour que le raffinement d'une transition soit complet, on veut qu'il décrive au moins tous les comportements décrits par la transition, c'est-à-dire que tout comportement décrit par la transition soit un comportement décrit par le raffinement.

Définition 2.3.6 (Complétude du raffinement d'une transition) *Soit $t \in \mathbb{T}_1$ et $\mathbb{G}_t$ un $\mathscr{L}_{\mathcal{C}_2}$-EIOLTS qui raffine t. On dit que $\mathbb{G}_t$ est un* raffinement complet *de t ssi :*

$$Sem_{\mathcal{M}_1}(t) \subseteq \rho_t^{2,1}(Sem_{\mathcal{M}_2}(\mathbb{G}_t))$$

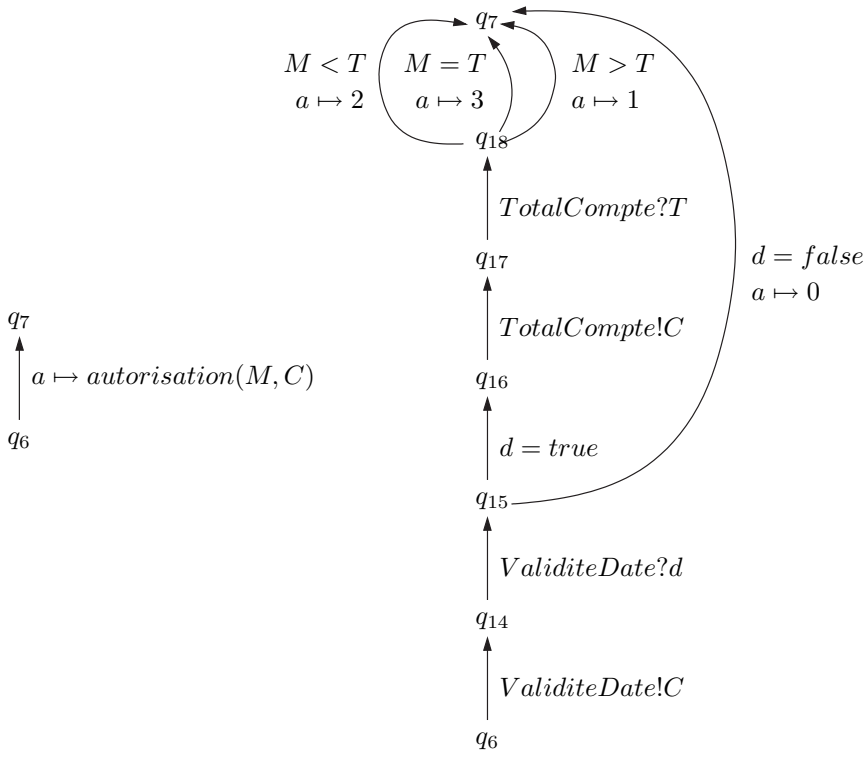


FIG. 2.4 – Un raffinement incorrect

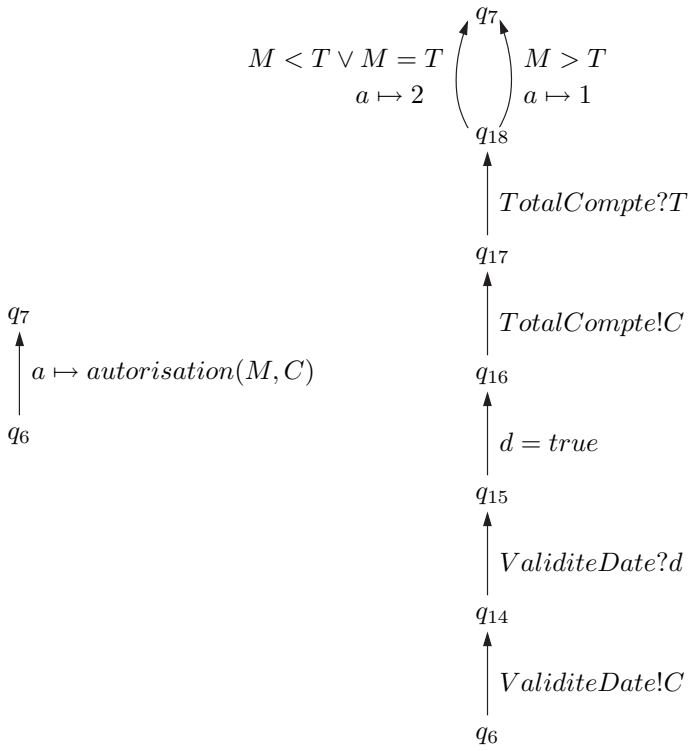


FIG. 2.5 – Un raffinement incomplet

Exemple 2.3.5 Par exemple, dans la figure 2.5, l'EIOLTS est un raffinement correct mais pas complet de la transition. En effet, les comportements que décrit l'EIOLTS existent dans la transition, mais le cas où $autorisation(M, C)$ rend la valeur 0 n'est pas spécifié, donc le raffinement ne décrit pas tous les comportements de la transition.

Remarque : le raffinement de l'exemple précédent était complet.

Maintenant qu'on a les conditions de correction et de complétude du raffinement d'une transition, on peut donner ces conditions pour le raffinement d'un EIOLTS. Le raffinement d'un EIOLTS $\mathbb{G}_1$ est l'EIOLTS formé par le raffinement de chacune des transitions de $\mathbb{G}_1$, donc pour qu'il soit correct, il suffit que chacun de ces raffinements soit correct au sens de la définition 2.3.5.

Définition 2.3.7 (Correction du raffinement d'un EIOLTS) *On dit que $\mathbb{G}_2$ est un* raffinement correct *de $\mathbb{G}_1$ ssi pour tout $t \in \mathbb{T}_1$, l'EIOLTS inclus dans $\mathbb{G}_2$ qui raffine t est un raffinement correct de t.*

La condition de complétude du raffinement d'un EIOLTS se déduit de la même façon de la définition 2.3.6 :

Définition 2.3.8 (Complétude du raffinement d'un EIOLTS) *On dit que $\mathbb{G}_2$ est un* raffinement complet *de $\mathbb{G}_1$ ssi pour tout $t \in \mathbb{T}_1$, l'EIOLTS inclus dans $\mathbb{G}_2$ qui raffine t est un raffinement complet de t.*

2.3.3 Transitivité du raffinement

Une propriété importante que l'on attend d'une relation de raffinement est qu'elle soit transitive. En effet, le but du raffinement est, en partant d'une spécification abstraite, de s'approcher par étapes successives de raffinement de l'implantation réelle du système. On veut alors s'assurer que le résultat de chaque étape de raffinement est toujours un raffinement de la spécification initiale. Autrement dit, on veut qu'en raffinant $\mathbb{G}_1$ par $\mathbb{G}_2$ puis $\mathbb{G}_2$ par $\mathbb{G}_3$, on ait bien raffiné $\mathbb{G}_1$ par $\mathbb{G}_3$.

On se donne donc ici trois EIOLTS-signatures $\mathscr{L}_{\mathcal{C}_1}$, $\mathscr{L}_{\mathcal{C}_2}$ et $\mathscr{L}_{\mathcal{C}_3}$ telles que $\mathscr{L}_{\mathcal{C}_1} \subseteq \mathscr{L}_{\mathcal{C}_2} \subseteq \mathscr{L}_{\mathcal{C}_3}$. On se donne aussi un $\mathscr{L}_{\mathcal{C}_1}$-EIOLTS $\mathbb{G}_1 = (\mathbb{Q}_1, q_{0_1}, \mathbb{T}_1)$.

Transitivité du raffinement syntaxique. On va tout d'abord montrer que le raffinement syntaxique est transitif, en commençant par le raffinement d'une transition. On va avoir besoin pour cela de la notion de chemin raffinant, définie comme suit :

Définition 2.3.9 (Chemin raffinant) *Soit $\mathbb{G}_2$ un $\mathscr{L}_{\mathcal{C}_2}$-EIOLTS raffinant $\mathbb{G}_1$. Soit $ch = t_1 \dots t_n \in Path(\mathbb{G}_1)$. Pour tout $i \in [\![1, n]\!]$, on note $\mathbb{G}_{t_i}$ l'EIOLTS qui raffine t_i dans $\mathbb{G}_2$. On appelle* chemin raffinant *de ch un chemin $ch' = ch_1 \cdot \dots \cdot ch_n \in Path(\mathbb{G}_2)$ tel que pour tout $i \in [\![1, n]\!]$, $ch_i \in Path(\mathbb{G}_{t_i})$, $source(ch_i) = source(t_i)$ et $cible(ch_i) = cible(t_i)$.*

Théorème 2.3.1 (Transitivité du raffinement syntaxique d'une transition) *Soit $t \in \mathbb{T}_1$. Soit $\mathbb{G}_t$ un $\mathscr{L}_{\mathcal{C}_2}$-EIOLTS qui raffine t syntaxiquement. Soit $\mathbb{G}'_t$ un $\mathscr{L}_{\mathcal{C}_3}$-EIOLTS qui raffine $\mathbb{G}_t$ syntaxiquement. Alors $\mathbb{G}'_t$ est un raffinement syntaxique de t.*

Démonstration. Montrons que $\mathbb{G}'_t = (\mathbb{Q}'_t, q'_{0_t}, \mathbb{T}'_t)$ raffine $t = (q, act, \varphi, \delta, q')$.

• Montrons que $\mathbb{Q}'_t \cap \mathbb{Q}_1 = \{q, q'\}$.

$$\begin{aligned}
\mathbb{Q}'_t \cap \mathbb{Q}_1 &= \Big(\bigcup_{t' \in \mathbb{T}_t} \mathbb{Q}_{t'} \Big) \cap \mathbb{Q}_1 \\
&= \bigcup_{t' \in \mathbb{T}_t} (\mathbb{Q}_{t'} \cap \mathbb{Q}_1) \\
&= \bigcup_{t' \in \mathbb{T}_t} (\mathbb{Q}_{t'} \cap \mathbb{Q}_2 \cap \mathbb{Q}_1) \qquad (\text{car } \mathbb{Q}_1 \subseteq \mathbb{Q}_2) \\
&= \bigcup_{t' \in \mathbb{T}_t} \big(\{q_{t'}, q'_{t'}\} \cap \mathbb{Q}_1\big) \qquad (\text{où } q_{t'} \text{ et } q'_{t'} \text{ sont la source et la cible de } t') \\
&= \Big(\bigcup_{t' \in \mathbb{T}_t} \{q_{t'}, q'_{t'}\} \Big) \cap \mathbb{Q}_1 \\
&= \mathbb{Q}_t \cap \mathbb{Q}_1 \\
&= \{q, q'\}
\end{aligned}$$

• On a $q'_{0_t} = q_{0_t} = q$

• Montrons que pour tout $q'' \in \mathbb{Q}'_t$, il existe $ch \in Path_{q''}(\mathbb{G}'_t)$ tel que $cible(ch) = q'$. Soit $q'' \in \mathbb{Q}'_t$.

Si $q'' \in \mathbb{Q}_t$, comme $\mathbb{G}_t$ est un raffinement de t, il existe $ch \in Path(\mathbb{G}_t)$ tel que $source(ch) = q''$ et $cible(ch) = q'$. De plus $\mathbb{G}'_t$ raffine $\mathbb{G}_t$. Donc si on prend $ch' \in Path(\mathbb{G}'_t)$ tel que ch' soit un chemin raffinant de ch, on a bien $source(ch') = q''$ et $cible(ch') = q'$.

Si $q'' \in \mathbb{Q}'_t \setminus \mathbb{Q}_t$, alors il existe $t' \in \mathbb{T}_t$ tel que $q'' \in \mathbb{Q}_{t'}$ où $\mathbb{G}_{t'} = (\mathbb{Q}_{t'}, q_{0_{t'}}, \mathbb{T}_{t'})$ est le raffinement de t' dans $\mathbb{G}'_t$. Par définition du raffinement d'une transition, on sait que pour tout $q'' \in \mathbb{Q}_{t'}$, il existe $ch' \in Path(\mathbb{G}_{t'})$ tel que $source(ch') = q''$ et $cible(ch') = cible(t')$. Or $cible(t') \in \mathbb{Q}_t$, donc d'après le point précédent, il existe $ch \in Path(\mathbb{G}'_t)$ tel que $source(ch) = cible(t')$ et $cible(ch) = q'$. On pose $ch'' = ch' \cdot ch$. On a alors $ch'' \in Path(\mathbb{G}'_t)$, $source(ch'') = source(ch') = q''$ et $cible(ch'') = cible(ch) = q'$.

Donc pour tout $q'' \in \mathbb{Q}'_t$, il existe $ch \in Path_{q''}(\mathbb{G}'_t)$ tel que $cible(ch) = q'$. Soit $q'' \in \mathbb{Q}'_t$.

• Montrons que pour tout chemin $ch = t_1 \dots t_n \in Path(\mathbb{G}'_t)$ tel que $source(ch) = q$ et $cible(ch) = q'$, il existe $k \in [\![1, n]\!]$ tel que t_k soit étiquetée par act, et pour tout $j \in [\![1, n]\!] \setminus \{k\}$, l'action de communication de t_j est τ ou utilise un nom de canal de $\mathcal{C}_2 \setminus \mathcal{C}_1$.

Soit $ch = t_1 \dots t_n \in Path(\mathbb{G}'_t)$ tel que $source(ch) = q$ et $cible(ch) = q'$. Alors il existe $ch' = t'_1 \dots t'_m \in Path(\mathbb{G}_t)$, $m \leq n$, tel que $ch = ch_1 \cdot \ldots \cdot ch_m$ soit un chemin raffinant de ch'. On a alors $source(ch') = q$ et $cible(ch') = q'$. Comme $\mathbb{G}_t$ raffine t, il existe $l \in [\![1, m]\!]$ tel que t'_l soit étiquetée par act. Alors, comme $ch_l \in Path(\mathbb{G}_{t'_l})$ est tel que $source(ch_l) = source(t'_l)$ et $cible(ch_l) = cible(t'_l)$ et comme $\mathbb{G}_{t'_l}$ raffine t'_l, il existe une transition de ch_l qui est étiquetée par act. Donc il existe $k \in [\![1, n]\!]$ tel que t_k soit étiquetée par act.

Soit $i \in [\![1, n]\!]$, $j \neq k$. Soit t_i une transition de ch. Si l'action de communication qui l'étiquète est différente de τ, comme elle provient du raffinement d'une transition t_j de

$\mathbb{T}_t$, on a deux possibilités : soit elle étiquetait t_j, alors elle utilise un nom de canal de $\mathcal{C}_2 \setminus \mathcal{C}_1$, soit elle a été créée par le raffinement, dans ce cas elle utilise un nom de canal de $\mathcal{C}_3 \setminus \mathcal{C}_2$. Or $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathcal{C}_3$, donc $(\mathcal{C}_2 \setminus \mathcal{C}_1) \subseteq (\mathcal{C}_3 \setminus \mathcal{C}_1)$ et $(\mathcal{C}_3 \setminus \mathcal{C}_2) \subseteq (\mathcal{C}_3 \setminus \mathcal{C}_1)$. Donc l'action de communication de t_i utilise un nom de canal de $\mathcal{C}_3 \setminus \mathcal{C}_1$.

Donc $\mathbb{G}'_t$ raffine t. □

Exemple 2.3.6 On prend $\mathscr{L}_{\mathcal{C}_1} = \mathscr{L}_{\mathcal{C}_{DAB}}$ et $\mathscr{L}_{\mathcal{C}_3} = \mathscr{L}_{\mathcal{C}_2}$ de l'exemple 2.3.1. De plus, on ajoute à $\mathscr{L}_{\mathcal{C}_1}$ les deux fonctions $cartevalide : int \to bool$ et $montantdispo : int \times int \to int$ pour former $\mathscr{L}_{\mathcal{C}_2}$. On raffine la fonction $autorisation$ à l'aide de $cartevalide$ et $montantdispo$, puis chacune de ces fonctions à l'aide des noms de canaux $ValiditeDate$ et $TotalCompte$ respectivement. L'EIOLTS obtenu est bien un raffinement de la transition initiale.

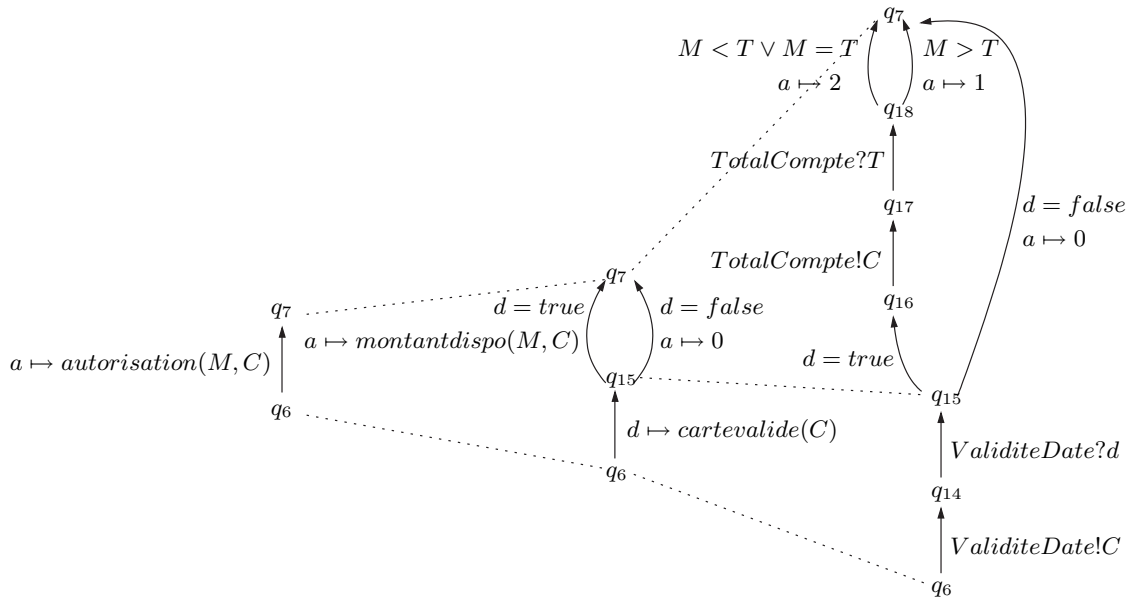


FIG. 2.6 – Transitivité du raffinement d'une transition

On peut maintenant démontrer la transitivité du raffinement syntaxique d'un EIOLTS à partir du théorème précédent.

Théorème 2.3.2 (Transitivité du raffinement syntaxique d'un EIOLTS) *Soit $\mathbb{G}_2$ un $\mathscr{L}_{\mathcal{C}_2}$-EIOLTS qui raffine $\mathbb{G}_1$ syntaxiquement. Soit $\mathbb{G}_3$ un $\mathscr{L}_{\mathcal{C}_3}$-EIOLTS qui raffine $\mathbb{G}_2$ syntaxiquement. Alors $\mathbb{G}_3$ est un raffinement syntaxique de $\mathbb{G}_1$.*

Démonstration. Soit $t \in \mathbb{T}_1$. Soit $\mathbb{G}_t = (\mathbb{Q}_t, q_{0_t}, \mathbb{T}_t)$ le raffinement de t dans $\mathbb{G}_2$. Soit $\mathbb{G}'_t = (\mathbb{Q}'_t, q'_{0_t}, \mathbb{T}'_t)$ le raffinement de $\mathbb{G}_t$ dans $\mathbb{G}_3$. D'une part, d'après le théorème précédent, on sait que $\mathbb{G}'_t$ raffine t. D'autre part, si on note, pour tout $t' \in \mathbb{T}_t$, $\mathbb{G}_{t'} = (\mathbb{Q}_{t'}, q_{0_{t'}}, \mathbb{T}_{t'})$ le raffinement de t' dans $\mathbb{G}'_t$, on a :

– $\mathbb{Q}'_t = \bigcup_{t' \in \mathbb{T}_t} \mathbb{Q}_{t'}$

– $q'_{0_t} = q_{0_1}$
– $\mathbb{T}'_t = \bigcup_{t' \in \mathbb{T}_t} \mathbb{T}_{t'}$

Montrons que $\mathbb{G}_3$ raffine $\mathbb{G}_1$. On note, pour tout $t \in \mathbb{T}_1$, $\mathbb{G}_t = (\mathbb{Q}_t, q_{0_t}, \mathbb{T}_t)$ le raffinement de t dans $\mathbb{G}_2$ et pour tout $t' \in \mathbb{T}_2$, $\mathbb{G}_{t'} = (\mathbb{Q}_{t'}, q_{0_{t'}}, \mathbb{T}_{t'})$ le raffinement de t' dans $\mathbb{G}_3$.

• $\mathbb{G}_3$ raffine $\mathbb{G}_2$, on a donc $\mathbb{Q}_3 = \bigcup_{t' \in \mathbb{T}_2} \mathbb{Q}_{t'}$. Or $\mathbb{G}_2$ raffine $\mathbb{G}_1$, on a donc $\mathbb{T}_2 = \bigcup_{t \in \mathbb{T}_1} \mathbb{T}_t$. On obtient alors $\mathbb{Q}_3 = \bigcup_{t \in \mathbb{T}_1} \bigcup_{t' \in \mathbb{T}_t} \mathbb{Q}_{t'}$. Or $\bigcup_{t' \in \mathbb{T}_t} \mathbb{Q}_{t'} = \mathbb{Q}'_t$, donc $\mathbb{Q}_3 = \bigcup_{t \in \mathbb{T}_1} \mathbb{Q}'_t$.

• $\mathbb{G}_3$ raffine $\mathbb{G}_2$, on a donc $q_{0_3} = q_{0_2}$. $\mathbb{G}_2$ raffine $\mathbb{G}_1$, donc $q_{0_2} = q_{0_1}$. D'où $q_{0_3} = q_{0_1}$.

• $\mathbb{G}_3$ raffine $\mathbb{G}_2$, on a donc $\mathbb{T}_3 = \bigcup_{t' \in \mathbb{T}_2} \mathbb{T}_{t'}$. Or $\mathbb{G}_2$ raffine $\mathbb{G}_1$, on a donc $\mathbb{T}_2 = \bigcup_{t \in \mathbb{T}_1} \mathbb{T}_t$. On obtient alors $\mathbb{T}_3 = \bigcup_{t \in \mathbb{T}_1} \bigcup_{t' \in \mathbb{T}_t} \mathbb{T}_{t'}$. Or $\bigcup_{t' \in \mathbb{T}_t} \mathbb{T}_{t'} = \mathbb{T}'_t$, donc $\mathbb{T}_3 = \bigcup_{t \in \mathbb{T}_1} \mathbb{T}'_t$.

Comme pour tout $t \in \mathbb{T}_1$, $\mathbb{G}'_t = (\mathbb{Q}'_t, q'_{0_t}, \mathbb{T}'_t)$ est le raffinement de t dans $\mathbb{G}_3$, des trois points précédents on déduit que $\mathbb{G}_3$ raffine $\mathbb{G}_1$. □

Conservation de la correction et de la complétude. On voudrait maintenant assurer la conservation des propriétés du raffinement par transitivité. En effet, on veut s'assurer par exemple que, si le raffinement à chaque étape est correct, le résultat de chaque étape de raffinement est un raffinement correct de l'EIOLTS initial. C'est ce qu'énonce le théorème suivant :

Théorème 2.3.3 (Conservation de la correction) *Soit un $\mathscr{L}_{\mathcal{C}_2}$-EIOLTS $\mathbb{G}_2$, raffinement correct de $\mathbb{G}_1$. Soit un $\mathscr{L}_{\mathcal{C}_3}$-EIOLTS $\mathbb{G}_3$, raffinement correct de $\mathbb{G}_2$. Alors $\mathbb{G}_3$ est un raffinement correct de $\mathbb{G}_1$.*

On introduit un lemme intermédiaire, qui énonce la propriété de conservation de la correction par transitivité pour une transition.

Lemme 1 (Conservation de la correction pour une transition) *Soit $t \in \mathbb{T}_1$. Soit un $\mathscr{L}_{\mathcal{C}_2}$-EIOLTS $\mathbb{G}_t$, raffinement correct de t. Soit un $\mathscr{L}_{\mathcal{C}_3}$-EIOLTS $\mathbb{G}'_t$, raffinement correct de $\mathbb{G}_t$. Alors $\mathbb{G}'_t$ est un raffinement correct de t.*

Démonstration. $\mathbb{G}_t = (\mathbb{Q}_t, q_{0_t}, \mathbb{T}_t)$ est un raffinement correct de t, c'est-à-dire que :

$$\rho_t(Sem_{\mathcal{M}_2}(\mathbb{G}_t)) \subseteq Sem_{\mathcal{M}_1}(t)$$

En d'autres termes, pour tout $ch' \in Path(\mathbb{G}_t)$ un chemin raffinant de t, pour tout $s' \in Sem_{\mathcal{M}_2}(ch')$, il existe $s \in Sem_{\mathcal{M}_1}(t)$ tel que $\rho_t^{2,1}(s') = s$.

$\mathbb{G}'_t$ est un raffinement correct de $\mathbb{G}_t$, c'est-à-dire que pour tout $t' \in \mathbb{T}_t$, si on note $\mathbb{G}_{t'}$ l'EIOLTS qui raffine t' dans $\mathbb{G}'_t$, on a :

$$\rho_{t'}(Sem_{\mathcal{M}_3}(\mathbb{G}_{t'})) \subseteq Sem_{\mathcal{M}_2}(t')$$

En d'autres termes, pour tout $t' \in \mathbb{T}_t$, pour tout $ch'' \in Path(\mathbb{G}_{t'})$ un chemin raffinant de t', pour tout $s'' \in Sem_{\mathcal{M}_3}(ch'')$, il existe $s' \in Sem_{\mathcal{M}_2}(t')$ tel que $\rho^{3,2}_{t'}(s'') = s'$.

Soit $ch'' = ch''_1 \cdot \ldots \cdot ch''_n \in Path(\mathbb{G}'_t)$ un chemin raffinant de t. Soit $s'' = s''_1 \centerdot \ldots \centerdot s''_n \in Sem_{\mathcal{M}_3}(ch'')$ où pour tout $1 \leq j \leq n$, $s''_j = (\nu^i_{1,j}, e_{1,j}, \nu^f_{1,j}) \ldots (\nu^i_{m_j,j}, e_{m_j,j}, \nu^f_{m_j,j})$.

Il existe $ch' = t'_1 \ldots t'_n \in Path(\mathbb{G}_t)$ chemin raffinant de t tel que ch'' soit un chemin raffinant de ch'. Comme $\mathbb{G}'_t$ est un raffinement correct de $\mathbb{G}_t$, on sait que pour tout $1 \leq j \leq n$, il existe $s'_j \in Sem_{\mathcal{M}_2}(t'_j)$ tel que $\rho^{3,2}_{t'_j}(s''_j) = s'_j$. On note :

$$\begin{aligned} s' &= s'_1 \ldots s'_n \\ &= \rho^{3,2}_{t'_1}(s''_1) \ldots \rho^{3,2}_{t'_n}(s''_n) \\ &= \left(\nu^i_{1,1|_{V_2}}, e_{k_1,1}, \nu^f_{m_1,1|_{V_2}}\right) \ldots \left(\nu^i_{1,n|_{V_2}}, e_{k_n,n}, \nu^f_{m_n,n|_{V_2}}\right) \end{aligned}$$

où pour tout $1 \leq j \leq n$, k_j désigne l'indice de la transition de ch''_j étiquetée par l'action de communication de t'_j. Comme $s'' \in Sem_{\mathcal{M}_3}(ch'')$, on sait que pour tout $1 \leq j \leq n-1$, $\nu^f_{m_j,j} = \nu^i_{1,j+1}$, donc $\nu^f_{m_j,j_{|_{V_2}}} = \nu^i_{1,j+1_{|_{V_2}}}$, donc on a bien $s' \in Sem_{\mathcal{M}_2}(ch')$.

Comme $\mathbb{G}_t$ est un raffinement correct de t, on sait qu'il existe $s \in Sem_{\mathcal{M}_1}(t)$ tel que $s = \rho^{2,1}_t(s') = \left(\nu^i_{1,1|_{V_2|V_1}}, e_{k_l,l}, \nu^f_{m_n,n|_{V_2|V_1}}\right) = \left(\nu^i_{1,1|_{V_1}}, e_{k_l,l}, \nu^f_{m_n,n|_{V_1}}\right)$, où l désigne l'indice de la transition de ch' étiquetée par l'action de communication de t.

Or $\rho^{3,1}_t(s'') = \left(\nu^i_{1,1|_{V_1}}, e_{k_l,l}, \nu^f_{m_n,n|_{V_1}}\right) = s$, donc il existe $s \in Sem_{\mathcal{M}_1}(t)$ tel que $\rho^{3,1}_t(s'') = s$. D'où $\mathbb{G}'_t$ est un raffinement correct de t. □

On peut maintenant démontrer le théorème 2.3.3 à l'aide du lemme précédent.

Démonstration. $\mathbb{G}_2$ est un raffinement correct de $\mathbb{G}_1$, c'est-à-dire que pour tout $t \in \mathbb{T}_1$, si on note $\mathbb{G}_t$ l'EIOLTS qui raffine t dans $\mathbb{G}_2$, $\mathbb{G}_t$ est un raffinement correct de t.

$\mathbb{G}_3$ est un raffinement correct de $\mathbb{G}_2$, donc en particulier, pour tout $t \in \mathbb{T}_1$, si on note $\mathbb{G}'_t$ l'EIOLTS qui raffine $\mathbb{G}_t$ dans $\mathbb{G}_3$, $\mathbb{G}'_t$ est un raffinement correct de $\mathbb{G}_t$.

D'après le lemme 1, $\mathbb{G}'_t$ est un raffinement correct de t, pour tout $t \in \mathbb{T}_1$. Donc $\mathbb{G}_3$ est un raffinement correct de $\mathbb{G}_1$. □

On va montrer la même propriété de conservation par transitivité pour la complétude, énoncée par le théorème suivant :

Théorème 2.3.4 (Conservation de la complétude) *Soit un $\mathscr{L}_{\mathcal{C}_2}$-EIOLTS $\mathbb{G}_2$, raffinement complet de $\mathbb{G}_1$. Soit un $\mathscr{L}_{\mathcal{C}_3}$-EIOLTS $\mathbb{G}_3$, raffinement complet de $\mathbb{G}_2$. Alors $\mathbb{G}_3$ est un raffinement complet de $\mathbb{G}_1$.*

On introduit un lemme intermédiaire, qui énonce la propriété de conservation de la complétude par transitivité pour une transition.

Lemme 2 (Conservation de la complétude) *Soit $t \in \mathbb{T}_1$. Soit un $\mathscr{L}_{\mathcal{C}_2}$-EIOLTS $\mathbb{G}_t$, raffinement complet de t. Soit un $\mathscr{L}_{\mathcal{C}_3}$-EIOLTS $\mathbb{G}'_t$, raffinement complet de $\mathbb{G}_t$. Alors $\mathbb{G}'_t$ est un raffinement complet de t.*

Démonstration. $\mathbb{G}_t = (\mathbb{Q}_t, q_{0_t}, \mathbb{T}_t)$ est un raffinement complet de t, c'est-à-dire que :

$$Sem_{\mathcal{M}_1}(t) \subseteq \rho_t(Sem_{\mathcal{M}_2}(\mathbb{G}_t))$$

En d'autres termes, pour tout $s \in Sem_{\mathcal{M}_1}(t)$, il existe $ch' \in Path(\mathbb{G}_t)$ un chemin raffinant de t, il existe $s' \in Sem_{\mathcal{M}_2}(ch')$ tel que $s = \rho_t^{2,1}(s')$.

$\mathbb{G}'_t$ est un raffinement complet de $\mathbb{G}_t$, c'est-à-dire que pour tout $t' \in \mathbb{T}_t$, si on note $\mathbb{G}_{t'}$ l'EIOLTS qui raffine t' dans $\mathbb{G}'_t$, on a :

$$Sem_{\mathcal{M}_2}(t') \subseteq \rho_{t'}(Sem_{\mathcal{M}_3}(\mathbb{G}_{t'}))$$

En d'autres termes, pour tout $t' \in \mathbb{T}_t$, pour tout $s' \in Sem_{\mathcal{M}_2}(t')$, il existe $ch'' \in Path(\mathbb{G}_{t'})$ un chemin raffinant de t', il existe $s'' \in Sem_{\mathcal{M}_3}(ch'')$ tel que $s' = \rho_{t'}^{3,2}(s'')$.

Soit $s = (\nu^i, e, \nu^f) \in Sem_{\mathcal{M}_1}(t)$. Comme $\mathbb{G}_t$ est un raffinement complet de t, il existe $ch' = t'_1 \dots t'_n \in Path(\mathbb{G}_t)$ un chemin raffinant de t, il existe $s' = s'_1 \dots s'_n \in Sem_{\mathcal{M}_2}(ch')$ tel que $s = \rho_t^{2,1}(s')$. On a donc $s' = (\nu_1^i, e_1, \nu_1^f) \dots (\nu_l^i, e_l, \nu_l^f) \dots (\nu_n^i, e_n, \nu_n^f)$ où l est l'indice de la transition de ch' étiquetée par l'action de communication de t, et tel que $\nu^i_{1_{|V_1}} = \nu^i$, $e_l = e$ et $\nu^f_{n_{|V_1}} = \nu^f$.

Comme $\mathbb{G}'_t$ est un raffinement complet de $\mathbb{G}_t$, pour tout $1 \leq j \leq n$, pour tout $s'_j \in Sem_{\mathcal{M}_2}(t'_j)$, il existe $ch''_j \in Path(\mathbb{G}_t)$ un chemin raffinant de t'_j, il existe $s''_j \in Sem_{\mathcal{M}_2}(ch''_j)$ tel que $s'_j = \rho_{t'_j}^{3,2}(s''_j)$. On a donc pour tout $1 \leq j \leq n$, $s''_j = (\nu^i_{1,j}, e_{1,j}, \nu^f_{1,j}) \dots (\nu^i_{k_j,j}, e_{k_j,j}, \nu^f_{k_j,j}) \dots (\nu^i_{m_j,j}, e_{m_j,j}, \nu^f_{m_j,j})$ où k_j désigne l'indice de la transition de ch''_j étiquetée par l'action de communication de t'_j, et tel que $\nu^i_{1,1_{|V_2}} = \nu^i_1$, $e_{k_l,l} = e_l = e$ et $\nu^f_{m_n,n_{|V_2}} = \nu^f_n$. On a donc $\nu^i_{1,1_{|V_2|V_1}} = \nu^i_{1_{|V_1}} = \nu^i$ et $\nu^f_{m_n,n_{|V_2|V_1}} = \nu^f_{n_{|V_1}} = \nu^f$.

On note $ch'' = ch''_1 \cdot \dots \cdot ch''_n$. D'une part $t'_1 \dots t'_n \in Path(\mathbb{G}_t)$, d'autre part ch''_j est un chemin raffinant de t'_j pour tout $1 \leq j \leq n$, donc $source(ch''_j) = source(t'_j)$ et $cible(ch''_j) = cible(t'_j)$, d'où $ch'' \in Path(\mathbb{G}'_t)$.

On note $s'' = s''_1 . \dots . s''_n$. D'une part $s''_j \in Sem_{\mathcal{M}_2}(ch''_j)$, pour tout $1 \leq j \leq n$. D'autre part, $\rho_{t'_1}^{3,2}(s''_1) \dots \rho_{t'_n}^{3,2}(s''_n) = s'_1 \dots s'_n \in Sem_{\mathcal{M}_2}(ch')$, donc $s'' \in Sem_{\mathcal{M}_3}(ch'')$.

De plus $\rho_t^{3,1}(s'') = (\nu^i_{1,1_{|V_1}}, e_{k_l,l}, \nu^f_{m_n,n_{|V_1}}) = (\nu^i, e, \nu^f) = s$.

Donc il existe $ch'' \in Path(\mathbb{G}'_t$, il existe $s'' \in Sem_{\mathcal{M}_3}(ch'')$ tel que $s = \rho_t^{3,1}(s'')$. D'où $\mathbb{G}'_t$ est un raffinement complet de t. □

On peut maintenant démontrer le théorème 2.3.4 à l'aide du lemme précédent.

Démonstration. $\mathbb{G}_2$ est un raffinement complet de $\mathbb{G}_1$, c'est-à-dire que pour tout $t \in \mathbb{T}_1$, si on note $\mathbb{G}_t$ l'EIOLTS qui raffine t dans $\mathbb{G}_2$, $\mathbb{G}_t$ est un raffinement complet de t.

$\mathbb{G}_3$ est un raffinement complet de $\mathbb{G}_2$, donc en particulier, pour tout $t \in \mathbb{T}_1$, si on note $\mathbb{G}'_t$ l'EIOLTS qui raffine $\mathbb{G}_t$ dans $\mathbb{G}_3$, $\mathbb{G}'_t$ est un raffinement complet de $\mathbb{G}_t$.

D'après le lemme 2, $\mathbb{G}'_t$ est un raffinement complet de t, pour tout $t \in \mathbb{T}_1$. Donc $\mathbb{G}_3$ est un raffinement complet de $\mathbb{G}_1$. □

Exemple 2.3.7 Dans la figure 2.6, chaque raffinement étant correct et complet, l'EIOLTS final est bien un raffinement correct et complet de la transition initiale.

Chapitre 3

Un formalisme axiomatique dédié à la spécification de systèmes réactifs

Le formalisme dynamique que nous proposons de décrire ici est dédié à la spécification de systèmes réactifs. L'idée de ce formalisme est de permettre d'abstraire le comportement des systèmes réactifs en considérant les EIOLTS précédemment décrits, non plus comme des spécifications mathématiques d'un système, mais comme des réalisations possibles associées à une spécification. L'abstraction sera obtenue en permettant qu'à une spécification soit associée non pas un EIOLTS, mais un ensemble d'EIOLTS qui satisfont les propriétés de la spécification. Il nous faut alors introduire dans ce formalisme des notions ayant un sens pour les EIOLTS, comme la notion de canal, d'émission et de réception de messages, et ceci afin de permettre d'exprimer des propriétés sur les chemins d'un EIOLTS. De plus, ce formalisme sera du premier ordre afin de prendre en compte les données manipulées par les EIOLTS. Nous allons donc présenter les deux aspects qui caractérisent ce formalisme : une syntaxe où l'on définit les notions de signature, termes et formules, et une sémantique qui donnera un sens aux différents éléments de la signature dans le cadre des EIOLTS.

3.1 Syntaxe

3.1.1 Signatures dynamiques

Dans le formalisme que nous allons définir, la dynamique portera sur l'envoi et la réception de messages via des canaux de communication. Une signature dynamique sera donc une signature au sens de la définition 1.1.1, dans laquelle on aura ajouté des noms de canaux pour nous permettre de décrire les actions de communications.

Définition 3.1.1 (Signature dynamique) *Une* signature dynamique $\delta\Sigma$ *est un couple* $(\Sigma, \mathcal{C})$ *où* Σ *est une signature au sens de la définition* 1.1.1 *et* $\mathcal{C}$ *est un ensemble*

dont les éléments sont des noms de canaux.

Exemple 3.1.1 On reprend l'exemple du distributeur automatique de billets. On donne une signature dynamique de ce distributeur que l'on note $\delta\Sigma_{DAB} = (S, F, R, \mathcal{C})$ où :

$S = \{int, bool\}$

$F = \{+ : int \times int \rightarrow int,$
$\quad estvalide : int \times int \rightarrow bool,$
$\quad autorisation : int \times int \rightarrow bool\}$

$R = \{=_b : bool \times bool,$
$\quad =_i : int \times int,$
$\quad < : int \times int\}$

$\mathcal{C} = \{Carte, Code, Montant, Billets\}$

3.1.2 Termes dynamiques

On peut maintenant construire les termes et les formules dynamiques sur les signatures dynamiques que l'on vient de définir. Les termes dynamiques sont construits inductivement de la manière suivante :

Définition 3.1.2 (Termes dynamiques) *Soit $\delta\Sigma = (\Sigma, \mathcal{C})$ une signature dynamique. Soit V un ensemble de variables sur Σ. L'ensemble des* termes dynamiques avec variables*, noté $T_{\delta\Sigma}(V)$, est le plus petit ensemble tel que :*

- *$\_ \in T_{\delta\Sigma}(V)$;*
- *si $c \in \mathcal{C}$ et $x \in V$, alors $c?x \in T_{\delta\Sigma}(V)$;*
- *si $c \in \mathcal{C}$ et $t \in T_\Sigma(V)$, alors $c!t \in T_{\delta\Sigma}(V)$;*
- *si $t_1, t_2 \in T_{\delta\Sigma}(V)$, alors $t_1; t_2 \in T_{\delta\Sigma}(V)$.*

De plus ; est associative, et $\_$ est neutre pour ;.

Intuitivement, un terme de la forme $c?x$ dénote la réception d'une valeur pour la variable x par le canal c et un terme de la forme $c!t$ dénote l'émission du terme t par le canal c. Le terme $\_$ dénote l'action de ne rien faire sur le système et $t_1; t_2$ dénote l'ordre séquentiel des actions représentées par t_1 et t_2. Par la propriété d'associativité, un terme dynamique t est un enchaînement d'actions élémentaires $t = t_1; \ldots; t_n$ où pour tout $1 \leq i \leq n$, t_i dénote une réception ou une émission.

Exemple 3.1.2 On prend l'ensemble de variables suivant :

$$V = V_{bool} \amalg V_{int} = \{C, compt, code, M\}$$

On donne quelques exemples de termes dynamiques sur $\delta\Sigma_{DAB}$.
Le distributeur reçoit la carte de l'utilisateur, puis reçoit le code saisi par l'utilisateur :

$$Carte?C; Code?code$$

Le distributeur reçoit le montant saisi par l'utilisateur, puis rend la carte et donne les billets correspondant au montant à l'utilisateur :

$$Montant?M; Carte!C; Billets!M$$

3.1.3 Formules dynamiques

À partir de ces termes dynamiques, on peut maintenant construire les formules. On construit tout d'abord les formules qui vont exprimer des propriétés sur un chemin d'un EIOLTS. On veut par exemple être capable d'écrire la propriété : « *après avoir effectué une certaine suite d'actions élémentaires, l'EIOLTS considéré vérifie la propriété* φ », ou bien : « *il est certain qu'il arrivera un moment où l'EIOLTS vérifiera la propriété* φ », où φ est elle-même une propriété dynamique. Ce genre de propriétés se spécifie par des opérateurs temporels portant sur le futur. On construit donc l'ensemble des formules dynamiques sur les chemins inductivement de la manière suivante :

Définition 3.1.3 (Formules dynamiques sur les chemins) *Soit* $\delta\Sigma = (\Sigma, \mathcal{C})$ *une signature dynamique. Soit* V *un ensemble de variables sur* Σ*. L'ensemble des* formules dynamiques sur les chemins*, noté* $\delta Sen_P(\delta\Sigma)$*, est défini de la manière suivante :*

- *si* $\varphi \in Sen(\Sigma)$*, alors* $\varphi \in \delta Sen_P(\delta\Sigma)$ *;*
- *si* $t \in T_{\delta\Sigma}(V)$*, alors* $\|t\| \in \delta Sen_P(\delta\Sigma)$ *;*
- *si* $t \in T_{\delta\Sigma}(V)$ *et* $\varphi \in \delta Sen_P(\delta\Sigma)$*, alors* $[t]\varphi \in \delta Sen_P(\delta\Sigma)$ *;*
- *si* $\varphi \in \delta Sen_P(\delta\Sigma)$*, alors* $\mathbf{F}\varphi, \mathbf{G}\varphi \in \delta Sen_P(\delta\Sigma)$ *;*
- *si* $\varphi, \psi \in \delta Sen_P(\delta\Sigma)$*, alors* $\varphi \mathbf{U} \psi \in \delta Sen_P(\delta\Sigma)$ *;*
- *si* $\varphi \in \delta Sen_P(\delta\Sigma)$*, alors* $\neg\varphi \in \delta Sen_P(\delta\Sigma)$ *;*
- *si* $\varphi, \psi \in \delta Sen_P(\delta\Sigma)$*, alors* $\varphi \wedge \psi, \varphi \vee \psi, \varphi \Rightarrow \psi \in \delta Sen_P(\delta\Sigma)$*.*

Donnons une intuition en langage naturel des formules décrites dans cette définition, les connecteurs logiques ayant le sens usuel :

- $\|t\|$ signifie que l'enchaînement d'actions représenté par le terme dynamique t existe dans le chemin considéré ;
- $[t]\varphi$ signifie qu'à partir d'un état courant, après l'enchaînement d'actions représenté par le terme dynamique t, la propriété φ est vérifiée pour l'état atteint par cet enchaînement ;
- $\mathbf{F}\varphi$ signifie qu'il existe un état futur où φ sera vraie (**F**inally) ;
- $\mathbf{G}\varphi$ signifie que pour tous les états futurs, φ sera vraie (**G**lobally) ;
- $\varphi \mathbf{U} \psi$ signifie que φ est vérifiée jusqu'à temps que ψ le soit (**U**ntil).

Exemple 3.1.3 On donne quelques exemples de formules sur un chemin.

Le distributeur recevra le numéro de carte de l'utilisateur puis le code :

$$\|Carte?C; Code?code\|$$

Le distributeur reçoit la carte et le code de l'utilisateur puis, si le code est valide, l'utilisateur pourra saisir un montant :

$$[Carte?C; Code?code](estvalide(code, C) =_b true \Rightarrow \|Montant?M\|)$$

Si le code saisi par l'utilisateur n'est pas valide et que le compteur passe à 3, alors l'utilisateur ne doit pas récupérer sa carte, c'est-à-dire que le distributeur ne doit pas renvoyer de carte avant qu'un nouvel utilisateur n'en ait introduit une :

$$(estvalide(code) =_b false \wedge compt + 1 =_i 3) \Rightarrow (\neg\|Carte!C; Carte?C\| \wedge \|Carte?C\|)$$

La valeur du compteur sera toujours comprise entre 0 et 3 :

$$\mathbf{G}\big(\neg(compt < 0) \wedge (compt < 3 \vee compt =_i 3)\big)$$

On a maintenant envie d'exprimer des propriétés sur l'ensemble des chemins qui ont pour origine le point de contrôle initial de l'EIOLTS qu'on considère, c'est-à-dire les exécutions réelles du système. On veut pouvoir dire que tous ces chemins vérifient une propriété φ, ou bien qu'il existe un de ces chemins où la propriété φ est vraie, φ étant une propriété exprimée par une formule dynamique sur les chemins de la définition précédente. On ajoute aussi aux formules déjà définies une formule qui permet de comparer deux chemins. On construit donc l'ensemble des formules dynamiques de la manière suivante :

Définition 3.1.4 (Formules dynamiques) *Soit $\delta\Sigma = (\Sigma, \mathcal{C})$ une signature dynamique. Soit V un ensemble de variables sur Σ. L'ensemble des* formules dynamiques*, noté $\delta Sen(\delta\Sigma)$, est défini de la manière suivante :*

- *si $\varphi \in \delta Sen_P(\delta\Sigma)$, alors $\mathbf{A}\varphi, \mathbf{E}\varphi \in \delta Sen(\delta\Sigma)$;*
- *si $t_1, t_2 \in T_{\delta\Sigma}(V)$, alors $t_1 \equiv t_2 \in \delta Sen(\delta\Sigma)$;*
- *si $\varphi \in \delta Sen(\delta\Sigma)$, alors $\neg\varphi \in \delta Sen(\delta\Sigma)$;*
- *si $\varphi, \psi \in \delta Sen(\delta\Sigma)$, alors $\varphi \wedge \psi, \varphi \vee \psi, \varphi \Rightarrow \psi \in \delta Sen(\delta\Sigma)$.*

De manière intuitive, $\mathbf{A}\varphi$ signifie que tous les chemins qui partent de l'état considéré vérifient φ (**A**lways), et $\mathbf{E}\varphi$ signifie qu'il existe un de ces chemins qui vérifie φ (**E**ventually). La formule $t_1 \equiv t_2$ exprime le fait que les deux suites d'actions représentées par t_1 et t_2 amènent le système dans le même état.

Exemple 3.1.4 On donne une spécification (incomplète) de $\mathbb{G}_{DAB}$:

Le distributeur doit demander à l'utilisateur sa carte, puis son code :

$$\mathbf{A}\ \|Carte?C; Code?code\|$$

Le distributeur ne doit pas refuser systématiquement tous les codes saisis :

$$\mathbf{EF}\ estvalide(code, C) =_b true$$

Le distributeur ne doit pas accepter systématiquement tous les codes saisis :

$$\mathbf{EF}\ estvalide(code, C) =_b false$$

Si après la réception de la carte et la saisie du code, le code est valide, alors le distributeur doit demander à l'utilisateur de saisir un montant :

$$\mathbf{A}\ [Carte?C; Code?code](estvalide(code, C) =_b true \Rightarrow \|Montant?M\|)$$

Si après la réception de la carte et la saisie du code, le code est erroné mais que l'utilisateur n'a pas encore fait ses trois essais, le distributeur doit lui demander de resaisir son code :

$$\mathbf{A}\ [Carte?C; Code?code]\big((estvalide(code, C) =_b false \wedge compt+1 < 3) \Rightarrow \|Code?code\|\big)$$

Si le code est erroné et que l'utilisateur en est à son troisième essai, le distributeur ne doit pas rendre sa carte à l'utilisateur et doit demander à recevoir une nouvelle carte :

$$\mathbf{A}\ [Carte?C; Code?code]\big((estvalide(code, C) =_b false \wedge compt + 1 =_i 3) \Rightarrow$$

$$(\neg\|Carte!C; Carte?C\| \wedge \|Carte?C\|)\big)$$

Le distributeur doit attribuer toutes les autorisations possibles :

$$\mathbf{EF}\ autorisation(M, C) =_i 0$$

$$\mathbf{EF}\ autorisation(M, C) =_i 1$$

$$\mathbf{EF}\ autorisation(M, C) =_i 2$$

Si après la réception de la carte et la saisie du montant, le distributeur attribue l'autorisation 0, l'utilisateur ne doit pas récupérer sa carte ni recevoir l'argent qu'il a demandé, et le distributeur doit demander à recevoir une nouvelle carte :

$$\mathbf{A}\ [Carte?C; Montant?M]\big(autorisation(M, C) =_i 0 \Rightarrow$$

$$\big((\neg\|Carte!C; Carte?C\| \wedge \neg\|Billets!M; Carte?C\|) \wedge \|Carte?C\|\big)\big)$$

Si l'autorisation donnée est égale à 1, le distributeur ne doit pas donner son argent à l'utilisateur, mais doit lui rendre sa carte avant de demander à en recevoir une nouvelle :

$$\mathbf{A}\ [Carte?C; Montant?M]\big(autorisation(M, C) =_i 1 \Rightarrow$$

$$(\neg\|Billets!M; Carte?C\| \wedge \|Carte!C; Carte?C\|)\big)$$

Si l'autorisation donnée est égale à 2, le distributeur doit rendre sa carte à l'utilisateur, puis lui donner l'argent qu'il a demandé, puis demander à recevoir une nouvelle carte :

$$\mathbf{A}\ [Carte?C; Montant?M](autorisation(M, C) =_i 2 \Rightarrow$$

$$\|Carte!C; Billets!M; Carte?C\|)$$

La valeur du compteur doit toujours être comprise entre 0 et 3 :

$$\mathbf{AG}\ \big(\neg(compt < 0) \wedge (compt < 3 \vee compt =_i 3)\big)$$

Le compteur doit pouvoir prendre toutes les valeurs de 0 à 3 :

$$\mathbf{EF}\ compt =_i 0$$

$$\mathbf{EF}\ compt =_i 1$$

$$\mathbf{EF}\ compt =_i 2$$

$$\mathbf{EF}\ compt =_i 3$$

3.2 Sémantique

Dans cette section, on va donner un sens mathématique aux différentes constructions symboliques de la syntaxe, termes et formules, en commençant par donner un sens aux éléments sur lesquels elles sont construites. On va donc commencer par associer un modèle à une signature dynamique.

3.2.1 Modèle dynamique

Comme on l'a dit plus haut, on a défini ce formalisme pour pouvoir écrire des propriétés qui nous permettent de décrire un ensemble d'EIOLTS. On veut donc que les EIOLTS, décrits à la section précédente comme des spécifications, deviennent des modèles de notre formalisme. On associera alors à une signature dynamique un modèle dynamique composé :

- d'un EIOLTS qui partage le vocabulaire de la signature dynamique (les propriétés que l'on va pouvoir écrire ne porteront donc que sur un sous-ensemble du vocabulaire de la signature de l'EIOLTS, ce qui permet d'anticiper le fait que les raffinements de cet EIOLTS doivent aussi être des modèles) ;
- d'un modèle du premier ordre associé à la signature du premier ordre contenue dans la signature dynamique (on interprétera alors l'EIOLTS dans un modèle du premier ordre qui s'oubliera sur ce modèle).

Définition 3.2.1 (Modèle dynamique) *Soit $\delta\Sigma = (\Sigma, \mathcal{C})$ une signature dynamique. Un* modèle dynamique *associé à $\delta\Sigma$ est un couple $(\mathbb{G}, \mathcal{M})$ où :*

- *$\mathbb{G}$ est un $\mathscr{L}_{\mathcal{C}_1}$-EIOLTS, où $\mathscr{L}_{\mathcal{C}_1} = (\Sigma_1, V_1, \mathcal{C}_1)$ est telle que $\Sigma \subseteq \Sigma_1$ et $\mathcal{C} \subseteq \mathcal{C}_1$;*
- *$\mathcal{M}$ est un Σ-modèle au sens de la définition* 1.1.1.

Exemple 3.2.1 Un modèle associé à $\delta\Sigma_{DAB}$ est, par exemple, le couple formé de l'EIOLTS $\mathbb{G}_{DAB}$ de l'exemple 2.1.2 et du modèle $\mathcal{M}$ qu'on a associé à $\mathscr{L}_{\mathcal{C}_{DAB}}$ dans l'exemple 2.2.1.

3.2.2 Interprétation des termes dynamiques

Un terme dynamique représente une action qui permet à l'EIOLTS de passer d'un état à un autre. Sémantiquement, les états sont vus comme des interprétations des variables du système à un moment donné de l'exécution. Un terme dynamique exprime donc le passage d'une interprétation des variables à une autre. On associe alors à chaque terme dynamique une relation binaire dans $\mathcal{M}^V \times \mathcal{M}^V$ qui donne un sens à ce terme.

Définition 3.2.2 (Interprétation des termes dynamiques) *Soit $\delta\Sigma = (\Sigma, \mathcal{C})$ une signature dynamique. Soit V un ensemble de variables sur Σ. Soit $\mathcal{M}$ un Σ-modèle. Soit $t \in T_{\delta\Sigma}(V)$. L'interprétation de t sur $\mathcal{M}$, notée $[\![t]\!]_{\mathcal{M}}$, est la relation binaire sur $\mathcal{M}^V \times \mathcal{M}^V$ définie sur la structure de t de la manière suivante :*

- $(\nu, \nu') \in [\![\_]\!]_{\mathcal{M}}$ *ssi* $\nu' = \nu$;
- $(\nu, \nu') \in [\![c?x]\!]_{\mathcal{M}}$ *ssi pour tout* $y \in V$, $y \neq x$, $\nu'(y) = \nu(y)$;
- $(\nu, \nu') \in [\![c!t]\!]_{\mathcal{M}}$ *ssi* $\nu' = \nu$;
- $(\nu, \nu') \in [\![t_1; t_2]\!]_{\mathcal{M}}$ *ssi il existe* $\nu'' \in \mathcal{M}^V$ *tel que* $(\nu, \nu'') \in [\![t_1]\!]_{\mathcal{M}}$ *et* $(\nu'', \nu') \in [\![t_2]\!]_{\mathcal{M}}$.

3.2.3 Satisfaction des formules dynamiques

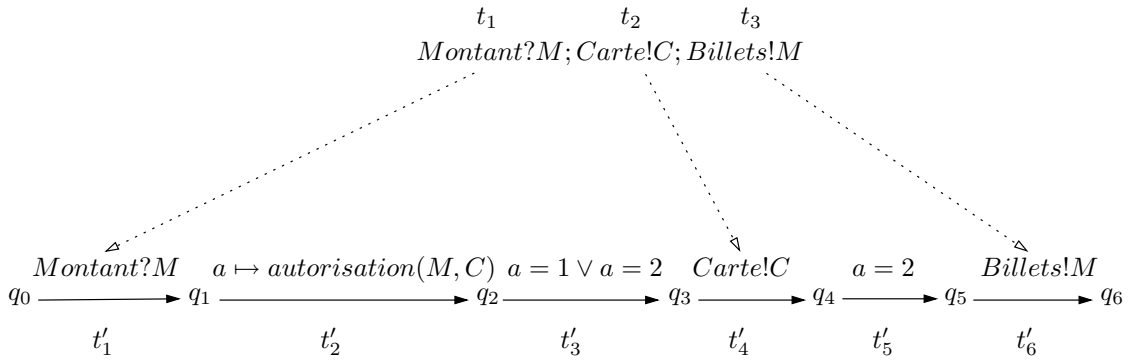
On se donne ici une signature dynamique $\delta\Sigma = (\Sigma, \mathcal{C})$ et un ensemble de variables V sur Σ. On se donne également un modèle associé à $\delta\Sigma$, c'est-à-dire un $\mathscr{L}_{\mathcal{C}_1}$-EIOLTS $\mathbb{G} = (\mathbb{Q}, q_0, \mathbb{T})$ où $\mathscr{L}_{\mathcal{C}_1} = (\Sigma_1, V_1, \mathcal{C}_1)$ telle que $\Sigma \subseteq \Sigma_1$, $V \subseteq V_1$ et $\mathcal{C} \subseteq \mathcal{C}_1$, et un Σ-modèle $\mathcal{M}$.

On va commencer par définir ce qu'est la satisfaction, par un chemin de l'EIOLTS, d'une formule dynamique qui exprime une propriété sur ce chemin. On se retrouve confronté aux formules de la forme $[t]\varphi$, qui expriment le fait qu'après avoir effectué l'action représentée par t, la suite du chemin satisfait φ. Comment définir ce qu'est l'action représentée par t dans le chemin ? On ne veut pas imposer que les termes dynamiques qui composent t représentent des actions strictement consécutives, car on veut être capable de tester une même formule sur un automate et son raffinement. On veut juste être capable de retrouver, dans le même ordre, chacune des actions élémentaires dans le chemin. On définit donc une application de plongement d'un terme dynamique dans un chemin, qui consiste à mettre en relation chacun des termes avec une transition du chemin, dans le même ordre que celui où ils apparaissent dans le terme.

Définition 3.2.3 (Plongement d'un terme dynamique dans un chemin) *Soit un terme dynamique $t = t_1; \ldots; t_n \in T_{\delta\Sigma}(V)$. Soit $ch = t'_1 \ldots t'_m \in Path(\mathbb{G})$, $m \geq n$, où pour tout $i \in [\![1, m]\!]$, $t'_i = (q_i, act_i, \varphi_i, \delta_i, q'_i)$. On dit que t* est plongé dans *ch ssi il existe une application $\lambda : [\![1, n]\!] \to [\![1, m]\!]$ strictement croissante telle que :*

- $\lambda(1) = 1$;
- $\lambda(n) = m$;
- *pour tout* $1 \leq i \leq n$, $t_i = act_{\lambda(i)}$.

Exemple 3.2.2 On prend le terme $t = t_1; t_2; t_3 = Montant?M; Carte!C; Billets!M$ et le chemin $ch = t'_1 t'_2 t'_3 t'_4 t'_5 t'_6$ suivant. t est plongé dans ch par l'application $\lambda : [\![1, 3]\!] \to [\![1, 6]\!]$ telle que $\lambda(1) = 1$, $\lambda(2) = 4$ et $\lambda(3) = 6$.



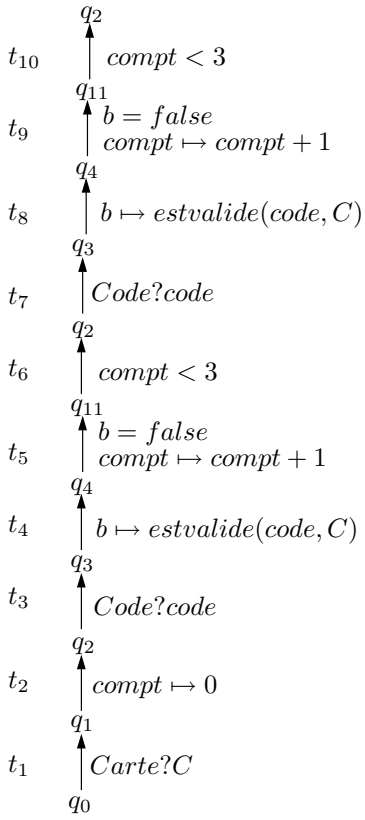
On peut maintenant définir la satisfaction, par un chemin, d'une formule dynamique sur un chemin.

Définition 3.2.4 (Satisfaction d'une formule dynamique par un chemin) *Soit un chemin $ch = t_1 \dots t_n \in Path(\mathbb{G})$. Soit $\nu : V \to \mathcal{M}$. On note $Sem^{\nu}_{\mathcal{M}}(ch)$ l'ensemble $\{s \in Sem_{\mathcal{M}}(ch) \mid \nu^i_1{}_{|_{V_1}} = \nu\}$. Soit $\varphi \in \delta Sen_P(\delta\Sigma)$. On définit la* satisfaction de φ par le modèle $(\mathbb{G}, \mathcal{M})$, pour le chemin ch et l'interprétation ν, *notée $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$, de la manière suivante :*

- *si $\varphi \in Sen(\Sigma)$, alors $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ ssi $\mathcal{M} \vDash_{\nu} \varphi$ au sens de la définition* 1.2.3 *;*
- *si φ est de la forme $\|t\|$, alors $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ ssi il existe $k_1, k_2 \in [\![1, n]\!]$, $k_1 \leq k_2$, tels que t est plongé dans $t_{k_1} \dots t_{k_2}$;*
- *si φ est de la forme $[t]\psi$, alors $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ ssi, s'il existe $k_1, k_2 \in [\![1, n]\!]$, $k_1 \leq k_2$, tels que t est plongé dans $t_{k_1} \dots t_{k_2-1}$, alors si on note $ch_{k_2} = t_{k_2} \dots t_n$, pour tout $s \in Sem^{\nu}_{\mathcal{M}}(ch)$, $(\mathbb{G}, \mathcal{M}) \vDash_{ch_{k_2}, \nu^i_{k_2}} \psi$;*
- *si φ est de la forme $\mathbf{F}\psi$, alors $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ ssi il existe $k \in [\![1, n]\!]$ tel que si on note $ch_k = t_k \dots t_n$, pour tout $s \in Sem^{\nu}_{\mathcal{M}}(ch)$, $(\mathbb{G}, \mathcal{M}) \vDash_{ch_k, \nu^i_k} \psi$;*
- *si φ est de la forme $\mathbf{G}\psi$, alors $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ ssi pour tout $k \in [\![1, n]\!]$, si on note $ch_k = t_k \dots t_n$, pour tout $s \in Sem^{\nu}_{\mathcal{M}}(ch)$, $(\mathbb{G}, \mathcal{M}) \vDash_{ch_k, \nu^i_k} \psi$;*
- *si φ est de la forme $\psi \mathbf{U} \chi$, alors $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ ssi $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \chi$ ou bien s'il existe $k \in [\![2, n]\!]$ tel que si on note pour tout $j \in [\![1, k]\!]$, $ch_j = t_j \dots t_n$, pour tout $s \in Sem^{\nu}_{\mathcal{M}}(ch)$*
  - *pour tout $l \in [\![1, k-1]\!]$, $(\mathbb{G}, \mathcal{M}) \vDash_{ch_l, \nu^i_l} \psi$;*
  - *$(\mathbb{G}, \mathcal{M}) \vDash_{ch_k, \nu^i_k} \chi$.*
- *si φ est de la forme $\neg\psi$, alors $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ ssi $(\mathbb{G}, \mathcal{M}) \nvDash_{ch,\nu} \psi$;*
- *si φ est de la forme $\psi \wedge \chi$, alors $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ ssi $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \psi$ et $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \chi$;*
- *si φ est de la forme $\psi \vee \chi$, alors $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ ssi $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \psi$ ou $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \chi$;*

– *si* φ *est de la forme* $\psi \Rightarrow \chi$, *alors* $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ *ssi, si* $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \psi$ *alors* $(\mathbb{G}, \mathcal{M}) \vDash_{ch,\nu} \chi$.

Exemple 3.2.3 • On considère le chemin $ch = t_1t_2t_3t_4t_5t_6t_7t_8t_9t_{10}$ suivant :



Soit $\nu : V \to \mathcal{M}$ une interprétation des variables, où $\mathcal{M}$ est le Σ-modèle défini précédemment.

On va montrer que $(\mathbb{G}_{DAB}, \mathcal{M})$ satisfait, pour le chemin ch et l'interprétation ν, la formule φ suivante :

$$\mathbf{F}\ compt =_i 2$$

Les éléments de la sémantique de ce chemin s'écrivent de la manière suivante, où $v_C, v^1_{code}, v^2_{code} \in \mathbb{N}$ tels que $v^1_{code} \neq v^2_{code}$:

$$(\nu, Carte?v_C, \nu_2)(\nu_2, \tau, \nu_3)(\nu_3, Code?v^1_{code}, \nu_4)(\nu_4, \tau, \nu_5)(\nu_5, \tau, \nu_6)(\nu_6, \tau, \nu_7)$$

$$(\nu_7, Code?v^2_{code}, \nu_8)(\nu_8, \tau, \nu_9)(\nu_9, \tau, \nu_{10})(\nu_{10}, \tau, \nu_{11})$$

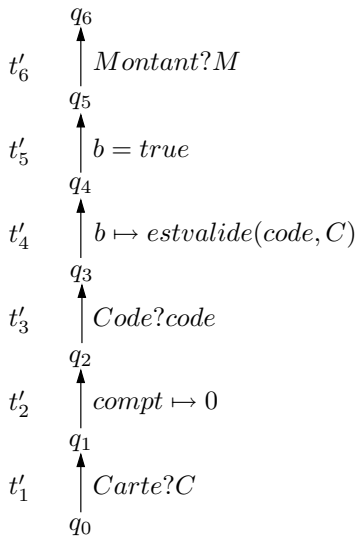
tels que $\nu_2(C) = v_C$, $\nu_3(compt) = 0$, $\nu_4(code) = v^1_{code}$, $\nu_5(b) = \mathbf{faux}$, $\nu_6(compt) = 1$, $\nu_8(code) = v^2_{code}$, $\nu_9(b) = \mathbf{faux}$, $\nu_{10}(compt) = 2$.

Montrer que $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{ch,\nu} \varphi$ est équivalent à montrer qu'il existe $k \in [\![1, 10]\!]$ tel que si on note $ch_k = t_k \dots t_{10}$, on a $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{ch_k,\nu_k} compt =_i 2$.

On prend $k = 10$. Il faut maintenant vérifier que $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{t_{10},\nu_{10}} compt =_i 2$, c'est-à-dire $\mathcal{M} \vDash_{\nu_{10}} compt =_i 2$. Or $\nu_{10}(compt) = 2$, donc $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{t_{10},\nu_{10}} compt =_i 2$.

D'où $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{ch,\nu} \mathbf{F}\ compt =_i 2$.

• On considère le chemin $ch' = t'_1 t'_2 t'_3 t'_4 t'_5 t'_6$ suivant :



Soit $\nu' : V \to \mathcal{M}$ une interprétation des variables, où $\mathcal{M}$ est le Σ-modèle défini précédemment.

On va montrer que $(\mathbb{G}_{DAB}, \mathcal{M})$ satisfait, pour le chemin ch' et l'interprétation ν', la formule φ suivante :

$$[Carte?C; Code?code](estvalide(code, C) = true \Rightarrow \|Montant?M\|)$$

Les éléments de la sémantique de ce chemin s'écrivent de la manière suivante, où $v_C, v_{code}, v_M \in \mathbb{N}$:

$$(\nu', Carte?v_C, \nu'_2)(\nu'_2, \tau, \nu'_3)(\nu'_3, Code?v_{code}, \nu'_4)(\nu'_4, \tau, \nu'_5)(\nu'_5, \tau, \nu'_6)(\nu'_6, Montant?v_M, \nu'_7)$$

tels que $\nu'_2(C) = v_C$, $\nu'_3(compt) = 0$, $\nu'_4(code) = v_{code}$, $\nu'_5(b) = \mathbf{vrai}$, $\nu'_7(M) = v_M$.

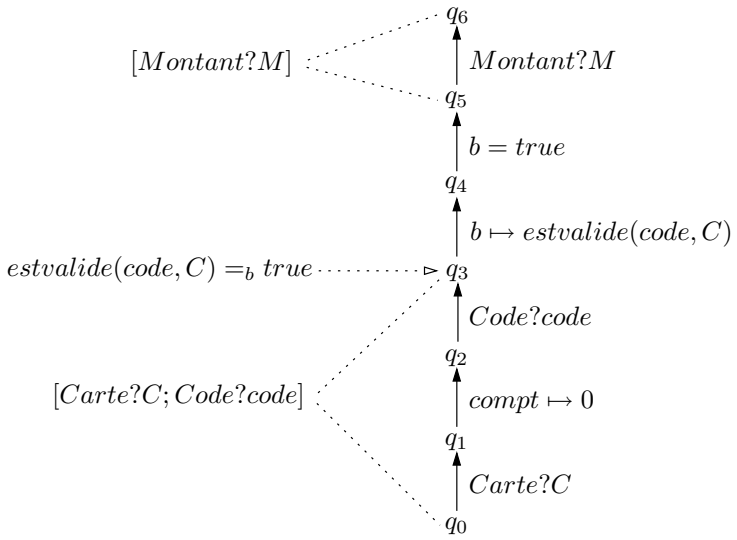
Montrer que $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{ch',\nu'} \varphi$ est équivalent à montrer que, s'il existe $k_1, k_2 \in [\![1, 6]\!]$ tels que $Carte?C; Code?code$ soit plongé dans $t'_{k_1} \dots t'_{k_2}$, alors si on note $ch'_{k_2+1} = t'_{k_2+1} \dots t'_6$, on a $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{ch'_{k_2+1}, \nu'_{k_2+1}} estvalide(code, C) = true \Rightarrow \|Montant?M\|$.

On prend $k_1 = 1$ et $k_2 = 3$. On peut plonger $Carte?C; Code?code$ dans $t'_1 t'_2 t'_3$ par l'application $\lambda : [\![1, 2]\!] \to [\![1, 3]\!]$ telle que $\lambda(1) = 1$ et $\lambda(2) = 3$.

On doit maintenant vérifier que $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{t'_4 t'_5 t'_6, \nu'_4} estvalide(code, C) = true \Rightarrow \|Montant?M\|$. On suppose que $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{t'_4 t'_5 t'_6, \nu'_4} estvalide(code, C) = true$. Montrons qu'il existe $k_3, k_4 \in [\![4, 6]\!]$ tels que $Montant?M$ soit plongé dans $t'_{k_3} \dots t'_{k_4}$.

On prend $k_3 = k_4 = 6$. $Montant?M$ étiquète la transition t'_6, donc

$$(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{t'_4 t'_5 t'_6, \nu'_4} estvalide(code, C) = true \Rightarrow \|Montant?M\|$$



D'où $(\mathbb{G}_{DAB}, \mathcal{M}) \models_{ch',\nu'} [Carte?C; Code?code](estvalide(code, C) = true \Rightarrow \|Montant?M\|)$.

Comme on ne dispose pas de la notion de chemin infini, on a besoin, pour la satisfaction des formules temporelles, de pouvoir étendre les chemins, de manière à disposer de chemins arbitrairement grands.

Définition 3.2.5 (Ensemble des extensions d'un chemin) *Soit* $ch \in Path(\mathbb{G})$. *On définit l'*ensemble des extensions *de* ch *dans* $\mathbb{G}$, *noté* $Ext_{\mathbb{G}}(ch)$, *par :*

$$Ext_{\mathbb{G}}(ch) = \{ch' \in Path(\mathbb{G}) \mid \exists ch'' \in Path(\mathbb{G}), ch' = ch \cdot ch''\}$$

On peut maintenant définir la satisfaction de toutes les formules par un EIOLTS donné, à partir de la définition 3.2.4.

Définition 3.2.6 (Satisfaction d'une formule dynamique) *Soit* $\varphi \in \delta Sen(\delta\Sigma)$. *On définit la* satisfaction de φ par le modèle $(\mathbb{G}, \mathcal{M})$, *notée* $(\mathbb{G}, \mathcal{M}) \models \varphi$, *de la manière suivante :*

- *si* φ *est de la forme* $\mathbf{A}\psi$, *alors* $(\mathbb{G}, \mathcal{M}) \models \varphi$ *ssi pour tout* $\nu : V \to \mathcal{M}$, *pour tout* $ch \in Path_{q_0}(\mathbb{G})$, *il existe* $ch' \in Ext_{\mathbb{G}}(ch)$ *tel que* $(\mathbb{G}, \mathcal{M}) \models_{ch',\nu} \psi$;
- *si* φ *est de la forme* $\mathbf{E}\psi$, *alors* $(\mathbb{G}, \mathcal{M}) \models \varphi$ *ssi pour tout* $\nu : V \to \mathcal{M}$, *il existe* $ch \in Path_{q_0}(\mathbb{G})$ *tel que* $(\mathbb{G}, \mathcal{M}) \models_{ch,\nu} \psi$;
- *si* φ *est de la forme* $t \equiv t'$, *alors* $(\mathbb{G}, \mathcal{M}) \models \varphi$ *ssi il existe* $ch = t_1 \dots t_n, ch' = t'_1 \dots t'_m \in Path_{q_0}(\mathbb{G})$ *tels qu'il existe* $k_1, k_2 \in [\![1, n]\!]$ *et* $k'_1, k'_2 \in [\![1, m]\!]$ *tels que* t *est plongé dans* $t_{k_1} \dots t_{k_2}$ *et* t' *est plongé dans* $t'_{k'_1} \dots t'_{k'_2}$, *et si*

$$\left\{ (\nu^i_{k_1}|_V, \nu^f_{k_2}|_V) \mid s \in Sem_{\mathcal{M}}(ch) \right\} = \left\{ (\nu^i_{k'_1}|_V, \nu^f_{k'_2}|_V) \mid s \in Sem_{\mathcal{M}}(ch') \right\}$$

- *si* φ *est de la forme* $\neg\psi$, *alors* $(\mathbb{G}, \mathcal{M}) \models \varphi$ *ssi* $(\mathbb{G}, \mathcal{M}) \not\models \psi$;

– *si* φ *est de la forme* $\psi \wedge \chi$, *alors* $(\mathbb{G}, \mathcal{M}) \vDash \varphi$ *ssi* $(\mathbb{G}, \mathcal{M}) \vDash \psi$ *et* $(\mathbb{G}, \mathcal{M}) \vDash \chi$;
– *si* φ *est de la forme* $\psi \vee \chi$, *alors* $(\mathbb{G}, \mathcal{M}) \vDash \varphi$ *ssi* $(\mathbb{G}, \mathcal{M}) \vDash \psi$ *ou* $(\mathbb{G}, \mathcal{M}) \vDash \chi$;
– *si* φ *est de la forme* $\psi \Rightarrow \chi$, *alors* $(\mathbb{G}, \mathcal{M}) \vDash \varphi$ *ssi, si* $(\mathbb{G}, \mathcal{M}) \vDash \psi$ *alors* $(\mathbb{G}, \mathcal{M}) \vDash \chi$.

Exemple 3.2.4 • On va montrer que $(\mathbb{G}_{DAB}, \mathcal{M})$ satisfait la formule φ suivante :

$$\mathbf{EF}\ compt =_i 2$$

Montrer que $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash \varphi$ est équivalent à montrer que pour tout $\nu : V \to \mathcal{M}$, il existe $ch \in Path_{q_0}(\mathbb{G})$ tel que $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{ch,\nu} \mathbf{F}\ compt =_i 2$. Or on a exhibé au premier point de l'exemple 3.2.3 un chemin tel que $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{ch,\nu} \mathbf{F}\ compt =_i 2$ pour tout $\nu : V \to \mathcal{M}$.

Donc $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash \mathbf{EF}\ compt =_i 2$.

• On va montrer que $(\mathbb{G}_{DAB}, \mathcal{M})$ satisfait la formule φ suivante :

$$\mathbf{A}\ [Carte?C; Code?code](estvalide(code, C) = true \Rightarrow \|Montant?M\|)$$

Montrer que $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash \varphi$ est équivalent à montrer que pour tout $\nu : V \to \mathcal{M}$, pour tout chemin ch d'origine q_0 suffisamment long, $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{ch,\nu} [Carte?C; Code?code](estvalide(code, C) = true \Rightarrow \|Montant?M\|)$.

Après la réception de la carte et la saisie du code, si $estvalide(code, C)$ rend la valeur $false$, alors la formule est vérifiée ($A \Rightarrow B$ est équivalent à $\neg A \vee B$). Donc pour tous les chemins ch pour lesquels $estvalide(code, C)$ rend la valeur $false$, $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{ch,\nu} [Carte?C; Code?code](estvalide(code, C) = true \Rightarrow \|Montant?M\|)$.

Après la réception de la carte et la saisie du code, si $estvalide(code, C)$ rend la valeur $true$, alors le seul chemin possible est le chemin ch' du deuxième point de l'exemple 3.2.3. On a montré dans cet exemple que pour tout $\nu' : V \to \mathcal{M}$, $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash_{ch',\nu'} [Carte?C; Code?code](estvalide(code, C) = true \Rightarrow \|Montant?M\|)$.

Donc $(\mathbb{G}_{DAB}, \mathcal{M}) \vDash \mathbf{A}\ [Carte?C; Code?code](estvalide(code, C) = true \Rightarrow \|Montant?M\|)$.

3.3 Conservation des propriétés dynamiques au travers du raffinement

On se donne ici deux EIOLTS-signatures $\mathscr{L}_{\mathcal{C}_1}$ et $\mathscr{L}_{\mathcal{C}_2}$ telles que $\mathscr{L}_{\mathcal{C}_1} \subseteq \mathscr{L}_{\mathcal{C}_2}$, ainsi qu'un $\mathscr{L}_{\mathcal{C}_1}$-EIOLTS $\mathbb{G}_1 = (\mathbb{Q}_1, q_{0_1}, \mathbb{T}_1)$. On se donne également un Σ_2-modèle $\mathcal{M}_2$ et on note $\mathcal{M}_1$ l'oubli de $\mathcal{M}_2$ sur Σ_1, et $\mathcal{M}$ l'oubli de $\mathcal{M}_1$ sur Σ.

On va avoir besoin ici d'une notion de correction plus forte que celle définie à la section précédente. En effet, on veut que le raffinement d'un EIOLTS vérifie les mêmes propriétés dynamiques que cet EIOLTS. Or ces propriétés sont des formules portant sur les variables de V, donc pour que l'EIOLTS raffiné satisfasse ces formules, il faut qu'il conserve le comportement de l'EIOLTS initial sur les variables de V_1. On a alors la définition suivante de la correction forte pour le raffinement d'une transition :

Définition 3.3.1 (Correction forte du raffinement d'une transition) *Soit* $t \in \mathbb{T}_1$. *Soit* $\mathbb{G}_t$ *un* $\mathscr{L}_{\mathcal{C}_2}$*-EIOLTS qui raffine* t. *On dit que* $\mathbb{G}_t$ *est un* raffinement correct fortement *ssi il est correct au sens de la définition* 2.3.5 *et si, pour tout* $ch = t_1 \dots t_n \in Path(\mathbb{G}_t)$ *chemin raffinant de* t, *pour tout* $s \in Sem_{\mathcal{M}_2}(ch)$, *il existe* $(\nu, e, \nu') \in Sem_{\mathcal{M}_1}(t)$ *tel que :*

- *pour tout* $j \in [\![1, k-1]\!]$, $\nu = \nu^i_j{}_{|V_1} = \nu^f_j{}_{|V_1} = \nu^i_k{}_{|V_1}$;
- *pour tout* $j \in [\![k+1, n]\!]$, $\nu^f_k{}_{|V_1} = \nu^i_j{}_{|V_1} = \nu^f_j{}_{|V_1} = \nu'$,

où k *désigne l'indice tel que* t_k *soit étiquetée par l'action de communication de* t.

On impose donc, en plus de la correction au sens de la définition 2.3.5, que toutes les interprétations des variables de ch avant t_k, ainsi que l'interprétation initiale de t_k, soient égales à ν sur les variables de V_1, et de la même manière, que toutes les interprétations des variables de ch après t_k, ainsi que l'interprétation finale de t_k, soient égales à ν' sur les variables de V_1. Cela nous assure que le raffinement n'induit pas de comportements nouveaux sur les variables de V_1.

On étend cette définition au raffinement d'EIOLTS de la manière suivante :

Définition 3.3.2 (Correction forte du raffinement d'un EIOLTS) *Soit* $\mathbb{G}_2$ *un raffinement syntaxique de* $\mathbb{G}_1$. *On dit que* $\mathbb{G}_2$ *est un* raffinement correct fortement *de* $\mathbb{G}_1$ *ssi pour tout* $t \in \mathbb{T}_1$, *l'EIOLTS inclus dans* $\mathbb{G}_2$ *qui raffine* t *est un raffinement correct fortement de* t.

On peut maintenant, à l'aide de cette notion de correction forte, énoncer le théorème suivant, qui assure la conservation des propriétés sur les chemins au travers du raffinement.

Théorème 3.3.1 (Conservation des propriétés sur les chemins) *Soit* $\mathbb{G}_2$ *un raffinement correct fortement de* $\mathbb{G}_1$. *Soit* $ch \in Path(\mathbb{G}_1)$. *Soit* $ch' \in Path(\mathbb{G}_2)$ *un chemin raffinant de* ch. *Pour toute formule* $\varphi \in \delta Sen_P(\delta\Sigma)$ *(exceptées* $\neg\varphi$ *et* $\varphi \Rightarrow \psi$*), pour tout* $\nu : V \to \mathcal{M}$, *si* $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \varphi$, *alors* $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \varphi$.

Pour démontrer ce théorème, on a besoin d'un lemme intermédiaire, qui assure que le plongement d'un terme dynamique dans un chemin est conservé au travers du raffinement.

Lemme 3 *Soit* $\mathbb{G}_2$ *un raffinement syntaxique de* $\mathbb{G}_1$. *Soient* $ch \in Path(\mathbb{G}_1)$ *et* $ch' = t_1 \dots t_n \in Path(\mathbb{G}_2)$ *un chemin raffinant de* ch. *Soit* $t \in T_{\delta\Sigma}(V)$ *tel que* t *soit plongé dans* ch. *Alors il existe* $k_1, k_2 \in [\![1, n]\!]$ *tels que* t *soit plongé dans* $t_{k_1} \dots t_{k_2}$.

Démonstration. Soient $ch = t_1 \dots t_n \in Path(\mathbb{G}_1)$ et $t = t'_1 \dots t'_p \in T_{\delta\Sigma}(V)$, $p \leq n$, tel que t soit plongé dans ch. Alors il existe $\lambda : [\![1, p]\!] \to [\![1, n]\!]$ strictement croissante telle que $\lambda(1) = 1$, $\lambda(p) = n$, et pour tout $i \in [\![1, p]\!]$, $t'_i = act_{\lambda(i)}$, où act_j est l'action de communication qui étiquète t_j.

Soit $ch' = ch_1 \cdot \ldots \cdot ch_n \in Path(\mathbb{G}_2)$ un chemin raffinant de ch. Pour tout $i \in [\![1, n]\!]$, on note $ch_i = t_i^1 \ldots t_i^{l_i}$ et $M_i = \sum_{j=1}^{i} l_j$. Pour tout $i \in [\![1, n]\!]$, ch_i est un chemin raffinant de t_i, donc il existe $k_i \in [\![1, l_i]\!]$ tel que $t_i^{k_i}$ soit étiquetée par act_i. On prend $K_1 = k_1$ et $K_2 = M_{n-1} + k_n$. On pose $\lambda' : [\![1, p]\!] \to [\![K_1, K_2]\!]$ telle que $\lambda'(1) = K_1$ et pout tout $i \in [\![2, p]\!]$, $\lambda'(i) = M_{\lambda(i)-1} + k_{\lambda(i)}$. Il est évident que λ' est strictement croissante et que pour tout $i \in [\![1, p]\!]$, $t'_i = act_{\lambda'(i)}$. Donc t est plongé dans $t_{K_1} \ldots t_{K_2}$. □

On peut maintenant démontrer le théorème 3.3.1 à l'aide du lemme précédent.

Démonstration. On note $ch = t_1 \ldots t_n$, $ch' = ch_1 \cdot \ldots \cdot ch_n$ et pour tout $j \in [\![1, n]\!]$, on note $ch_j = {t'_j}^1 \ldots {t'_j}^{l_j}$ et $M_j = \sum_{i=1}^{j} l_i$. On a alors $ch' = t'_1 \ldots t'_{M_n}$.

Soit $\varphi \in \delta Sen_P(\delta\Sigma)$. Soit $\nu : V \to \mathcal{M}$.

On suppose que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \varphi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \varphi$ par récurrence sur la structure des formules.

Cas de base.

• $\varphi \in Sen(\Sigma)$. On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \varphi$, donc $\mathcal{M} \vDash_\nu \varphi$. On a donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \varphi$.

• φ est de la forme $\|t\|$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \|t\|$, c'est-à-dire qu'il existe $k'_1, k'_2 \in [\![1, M_n]\!]$, tels que t soit plongé dans $t'_{k'_1} \ldots t'_{k'_2}$.

On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \|t\|$, c'est-à-dire qu'il existe $k_1, k_2 \in [\![1, n]\!]$, tels que t soit plongé dans $t_{k_1} \ldots t_{k_2}$. D'après le lemme 3, comme t est plongé dans $t_{k_1} \ldots t_{k_2}$, il existe $k'_1, k'_2 \in [\![M_{k_1-1} + 1, M_{k_2}]\!] \subseteq [\![1, M_n]\!]$, tels que t est plongé dans $t'_{k'_1} \ldots t'_{k'_2}$.

Donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \|t\|$.

Récurrence. On suppose que la propriété est vraie pour toute formule de $\delta Sen_P(\delta\Sigma)$ de taille inférieure ou égale à N. On va montrer qu'elle est vraie pour toute formule de taille $N + 1$.

• φ est de la forme $[t]\psi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} [t]\psi$ c'est-à-dire que s'il existe $k'_1, k'_2 \in [\![1, M_n]\!]$, tels que t soit plongé dans $t'_{k'_1} \ldots t'_{k'_2}$, alors pour tout $s' \in Sem^\nu_{\mathcal{M}_2}(ch')$, si on note $ch'_{k'_2+1} = t'_{k'_2+1} \ldots t'_{M_n}$, on a $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{k'_2+1}, \nu^i_{k'_2+1}|_V} \psi$.

On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} [t]\psi$, c'est-à-dire que s'il existe $k_1, k_2 \in [\![1, n]\!]$ tels que t soit plongé dans $t_{k_1} \ldots t_{k_2}$, alors pour tout $s \in Sem^\nu_{\mathcal{M}_1}(ch)$, si on note $ch_{k_2+1} = t_{k_2+1} \ldots t_n$, on a $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch_{k_2+1}, \mu^i_{k_2+1}|_V} \psi$. On suppose qu'il existe de tels k_1 et k_2.

D'une part, d'après le lemme 3, comme t est plongé dans $t_{k_1} \ldots t_{k_2}$, il existe $k'_1, k'_2 \in [\![M_{k_1-1} + 1, M_{k_2}]\!] \subseteq [\![1, M_n]\!]$, tels que t est plongé dans $t'_{k'_1} \ldots t'_{k'_2}$.

D'autre part, on sait que pour tout $s \in Sem^\nu_{\mathcal{M}_1}(ch)$, $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch_{k_2+1}, \mu^i_{k_2+1}|_V} \psi$. L'indice $k'_2 \in [\![M_{k_2-1} + 1, M_{k_2}]\!]$ est l'indice de la transition de ch' étiquetée par l'action de communication de t_{k_2}. Comme $\mathbb{G}_2$ est un raffinement correct fortement de $\mathbb{G}_1$, on sait

que pour tout $s' \in Sem^{\nu}_{\mathcal{M}_2}(ch')$, il existe $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$ telle que $\nu^i_{k'_2+1}{}_{|V_1} = \nu^f_{M_{k_2}}{}_{|V_1} = \nu^i_{M_{k_2}+1}{}_{|V_1} = \mu^i_{k_2+1}$. Donc $\nu^i_{k'_2+1}{}_{|V} = \nu^i_{M_{k_2}+1}{}_{|V} = \mu^i_{k_2+1}{}_{|V}$.

Par hypothèse de récurrence, si on note $ch'_{M_{k_2}+1} = t'_{M_{k_2}+1} \dots t'_{M_n}$, comme $ch'_{M_{k-2}+1}$ est un chemin raffinant de ch_{k_2+1}, on a $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{M_{k_2}+1}, \nu^i_{M_{k_2}+1}{}_{|V}} \psi$. Donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{k'_2+1}, \nu^i_{k'_2+1}{}_{|V}} \psi$.

D'où $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} [t]\psi$.

• φ est de la forme $\mathbf{F}\psi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \mathbf{F}\psi$ c'est-à-dire qu'il existe $k' \in [\![1, M_n]\!]$ tel que pour tout $s' \in Sem^{\nu}_{\mathcal{M}_2}(ch')$, si on note $ch'_{k'} = t'_{k'} \dots t'_{M_n}$, on a $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{k'}, \nu^i_{k'}{}_{|V}} \psi$.

On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \mathbf{F}\psi$, donc il existe $k \in [\![1, n]\!]$ tel que pour tout $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$, si on note $ch_k = t_k \dots t_n$, on a $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch_k, \mu^i_k{}_{|V}} \psi$. On prend $k' = M_{k-1}+1$. Alors $ch_k \cdot \dots \cdot ch_n = t'_{k'} \dots t'_{M_n}$ est un chemin raffinant de $t_k \dots t_n$. D'autre part, comme $\mathbb{G}_2$ est un raffinement correct fortement de $\mathbb{G}_1$, c'est également un raffinement correct, donc pour tout $s' \in Sem^{\nu}_{\mathcal{M}_2}(ch')$, il existe $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$ telle que $\nu^i_{k'}{}_{|V_1} = \mu^i_k$, donc $\nu^i_{k'}{}_{|V} = \mu^i_k{}_{|V}$. Donc par hypothèse de récurrence, on a $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{k'}, \nu^i_{k'}{}_{|V}} \psi$.

D'où $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \mathbf{F}\psi$.

• φ est de la forme $\mathbf{G}\psi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \mathbf{G}\psi$ c'est-à-dire que pour tout $k' \in [\![1, M_n]\!]$, pour tout $s' \in Sem^{\nu}_{\mathcal{M}_2}(ch')$, si on note $ch'_{k'} = t'_{k'} \dots t'_{M_n}$, on a $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{k'}, \nu^i_{k'}{}_{|V}} \psi$.

Soit $k' \in [\![1, M_n]\!]$. Comme $k' \in [\![1, M_n]\!]$, il existe $k \in [\![1, n]\!]$ tel que $k' \in [\![M_{k-1} + 1, M_k]\!]$. On note l' l'indice de la transition de ch' étiquetée par l'action de communication de t_k.

- $k' \leq l'$: on sait que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \mathbf{G}\psi$, donc en particulier pour tout $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$, si on note $ch_k = t_k \dots t_n$, on a $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch_k, \mu^i_k{}_{|V}} \psi$. Comme $\mathbb{G}_2$ est un raffinement correct fortement de $\mathbb{G}_1$, on sait que pour tout $s' \in Sem^{\nu}_{\mathcal{M}_2}(ch')$, il existe $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$ telle que $\nu^i_{k'}{}_{|V_1} = \nu^i_{M_{k-1}+1}{}_{|V_1} = \mu^i_k$. Donc $\nu^i_{k'}{}_{|V} = \nu^i_{M_{k-1}+1}{}_{|V} = \mu^i_k{}_{|V}$. Par hypothèse de récurrence, si on note $ch'_{M_{k-1}+1} = t'_{M_{k-1}+1} \dots t'_{M_n}$, comme $ch'_{M_{k-1}+1}$ est un chemin raffinant de ch_k, on a $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{M_{k-1}+1}, \nu^i_{M_{k-1}+1}{}_{|V}} \psi$. Donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{k'}, \nu^i_{k'}{}_{|V}} \psi$.
- $k' > l'$: on sait que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \mathbf{G}\psi$, donc en particulier pour tout $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$, si on note $ch_{k+1} = t_{k+1} \dots t_n$, on a $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch_{k+1}, \mu^i_{k+1}{}_{|V}} \psi$. Comme $\mathbb{G}_2$ est un raffinement correct fortement de $\mathbb{G}_1$, on sait que pour tout $s' \in Sem^{\nu}_{\mathcal{M}_2}(ch')$, il existe $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$ telle que $\nu^i_{k'}{}_{|V_1} = \nu^f_{M_k}{}_{|V_1} = \nu^i_{M_k+1}{}_{|V_1} =$

μ^i_{k+1}. Donc $\nu^i_{k'}{}_{|V} = \nu^i_{M_k+1}{}_{|V} = \mu^i_{k+1}{}_{|V}$. Par hypothèse de récurrence, si on note $ch'_{M_k+1} = t'_{M_k+1} \dots t'_{M_n}$, comme ch'_{M_k+1} est un chemin raffinant de ch_{k+1}, on a $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{M_k+1}, \nu^i_{M_k+1}{}_{|V}} \psi$. Donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{k'}, \nu^i_{k'}{}_{|V}} \psi$.

D'où $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch', \nu} \mathbf{G}\psi$.

• φ est de la forme $\psi \mathbf{U} \chi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch', \nu} \psi \mathbf{U} \chi$ c'est-à-dire que :

– soit $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch', \nu} \chi$

– soit il existe $k' \in [\![2, M_n]\!]$ tel que si on note pour tout $j' \in [\![1, k']\!]$, $ch'_{j'} = t'_{j'} \dots t'_{M_n}$, pour tout $s' \in Sem^{\nu}_{\mathcal{M}_2}(ch')$,
  – pour tout $l' \in [\![1, k'-1]\!]$, $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{l'}, \nu^i_{l'}{}_{|V}} \psi$
  – $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{k'}, \nu^i_{k'}{}_{|V}} \chi$

On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch, \nu} \psi \mathbf{U} \chi$. Si $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch, \nu} \chi$, alors par hypothèse de récurrence, $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch', \nu} \chi$. Sinon, il existe $k \in [\![2, n]\!]$ tel que si on note pour tout $j \in [\![1, k]\!]$, $ch_j = t_j \dots t_n$, pour tout $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$,

– pour tout $l \in [\![1, k-1]\!]$, $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch_l, \mu^i_l{}_{|V}} \psi$

– $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch_k, \mu^i_k{}_{|V}} \chi$

On note $p' \in [\![M_{k-2}+1, M_{k-1}]\!]$ l'indice de la transition de ch' étiquetée par l'action de communication de t_{k-1}. On prend $k' = p'+1$. Soit $l' \in [\![1, k'-1]\!]$. Comme $l' \in [\![1, k'-1]\!]$, il existe $l \in [\![1, k-1]\!]$ tel que $l' \in [\![M_{l-1}+1, M_l]\!]$. On note $j' \in [\![M_{l-1}+1, M_l]\!]$ l'indice de la transition de ch' étiquetée par l'action de communication de t_l.

– $l' \leq j'$: comme $l \in [\![1, k-1]\!]$, on sait que pour tout $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$, $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch_l, \mu^i_l{}_{|V}} \psi$. Comme $\mathbb{G}_2$ est un raffinement correct fortement de $\mathbb{G}_1$, on sait que pour tout $s' \in Sem^{\nu}_{\mathcal{M}_2}(ch')$, il existe $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$ telle que $\nu^i_{l'}{}_{|V_1} = \nu^i_{M_{l-1}+1}{}_{|V_1} = \mu^i_l$. Donc $\nu^i_{l'}{}_{|V} = \nu^i_{M_{l-1}+1}{}_{|V} = \mu^i_l{}_{|V}$. Par hypothèse de récurrence, si on note $ch'_{M_{l-1}+1} = t'_{M_{l-1}+1} \dots t'_{M_n}$, comme $ch'_{M_{l-1}+1}$ est un chemin raffinant de ch_l, on a $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{M_{l-1}+1}, \nu^i_{M_{l-1}+1}{}_{|V}} \psi$. Donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{l'}, \nu^i_{l'}{}_{|V}} \psi$.

– $l' > j'$ pour $j' \neq p'$: comme $l \in [\![1, k-1]\!]$, on sait que pour tout $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$, si on note $ch_{l+1} = t_{l+1} \dots t_n$, on a $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch_{l+1}, \mu^i_{l+1}{}_{|V}} \psi$. Comme $\mathbb{G}_2$ est un raffinement correct fortement de $\mathbb{G}_1$, on sait que pour tout $s' \in Sem^{\nu}_{\mathcal{M}_2}(ch')$, il existe $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$ telle que $\nu^i_{l'}{}_{|V_1} = \nu^f_{M_l}{}_{|V_1} = \nu^i_{M_l+1}{}_{|V_1} = \mu^i_{l+1}$. Donc $\nu^i_{l'}{}_{|V} = \nu^i_{M_l+1}{}_{|V} = \mu^i_{l+1}{}_{|V}$. Par hypothèse de récurrence, si on note $ch'_{M_l+1} = t'_{M_l+1} \dots t'_{M_n}$, comme ch'_{M_l+1} est un chemin raffinant de ch_{l+1}, on a $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{M_l+1}, \nu^i_{M_l+1}{}_{|V}} \psi$. Donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{l'}, \nu^i_{l'}{}_{|V}} \psi$.

Donc pour tout $l' \in [\![1, k'-1]\!]$, $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{l'}, \nu^i_{l'}} \psi$.

On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch_k, \mu^i_k|_V} \chi$. Comme $\mathbb{G}_2$ est un raffinement correct fortement de $\mathbb{G}_1$, on sait que pour tout $s' \in Sem^{\nu}_{\mathcal{M}_2}(ch')$, il existe $s \in Sem^{\nu}_{\mathcal{M}_1}(ch)$ telle que $\nu^i_{k'|_{V_1}} = \nu^f_{M_{k-1}|_{V_1}} = \nu^i_{M_{k-1}+1|_{V_1}} = \nu^i_k$. Donc $\nu^i_{k'|_V} = \nu^i_{M_{k-1}+1|_V} = \nu^i_{k|_V}$. Par hypothèse de récurrence, si on note $ch'_{M_{k-1}+1} = t'_{M_{k-1}+1} \dots t'_{M_n}$, comme $ch'_{M_{k-1}+1}$ est un chemin raffinant de ch_k, on a $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{M_{k-1}+1}, \nu^i_{M_{k-1}+1}|_V} \chi$. Donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch'_{k'}, \nu^i_{k'|_V}} \chi$.

D'où $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi \mathbf{U} \chi$.

• φ est de la forme $\psi \wedge \chi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi \wedge \chi$, c'est-à-dire que $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi$ et $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \chi$.

On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \psi \wedge \chi$, c'est-à-dire que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \psi$ et $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \chi$. Par hypothèse de récurrence, comme $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \psi$, alors $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi$, et comme $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \chi$, alors $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \chi$. Donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi \wedge \chi$.

• φ est de la forme $\psi \vee \chi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi \vee \chi$, c'est-à-dire que $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi$ ou $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \chi$.

On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \psi \vee \chi$, c'est-à-dire que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \psi$ ou $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \chi$. Si $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \psi$, alors par hypothèse de récurrence, $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi$, donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi \vee \chi$. Si $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \chi$, alors par hypothèse de récurrence, $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \chi$, donc $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi \vee \chi$. □

Il ne reste plus à présent qu'à généraliser le théorème précédent à toutes les formules que peut satisfaire un EIOLTS.

Théorème 3.3.2 (Conservation des propriétés au travers du raffinement) *Pour tous $\delta\Sigma$-modèles $(\mathbb{G}_1, \mathcal{M})$ et $(\mathbb{G}_2, \mathcal{M})$ tels que $\mathbb{G}_2$ est un raffinement de $\mathbb{G}_1$ correct fortement, pour toute formule $\varphi \in \delta Sen(\delta\Sigma)$ (exceptée $t \equiv t'$), si $(\mathbb{G}_1, \mathcal{M}) \vDash \varphi$, alors $(\mathbb{G}_2, \mathcal{M}) \vDash \varphi$.*

Démonstration. Soit $\varphi \in \delta Sen(\delta\Sigma)$.

On suppose que $(\mathbb{G}_1, \mathcal{M}) \vDash \varphi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash \varphi$ par récurrence sur la structure des formules.

Cas de base.

• φ est de la forme $\mathbf{A}\psi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash \mathbf{A}\psi$, c'est-à-dire que pour tout $\nu : V \to \mathcal{M}$, pour tout $ch' \in Path_{q_0}(\mathbb{G}_2)$, il existe $Ch' \in Ext_{\mathbb{G}_2}(ch')$ tel que $(\mathbb{G}_2, \mathcal{M}) \vDash_{Ch',\nu} \psi$.

Soit $\nu : V \to \mathcal{M}$. Soit $ch' = t'_1 \dots t'_m \in Path_{q_0}(\mathbb{G}_2)$. Alors il existe $ch = t_1 \dots t_n \in Path_{q_0}(\mathbb{G}_1)$ tel que ch' s'écrive $ch_1 \cdot \dots \cdot ch_n$ où $ch_1 \cdot \dots \cdot ch_{n-1}$ est un chemin raffinant de $t_1 \dots t_{n-1}$ et $ch_n \in Path(\mathbb{G}_{t_n})$ est tel que $source(ch_n) = source(t_n)$. On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash \mathbf{A}\psi$ donc en particulier il existe $Ch = t_1 \dots t_n \dots t_N \in Ext_{\mathbb{G}_1}(ch)$ tel que $(\mathbb{G}_1, \mathcal{M}) \vDash_{Ch,\nu} \psi$. On pose $Ch' = ch_1 \cdot \dots \cdot ch_{n-1} \cdot ch'_n \cdot ch_{n+1} \cdot \dots \cdot ch_N$ où $ch'_n = ch_n \cdot ch''_n$, tel que Ch' soit un chemin raffinant de Ch. On a bien $Ch' \in Ext_{\mathbb{G}_2}(ch')$ d'une part, et d'autre part, d'après le théorème 3.3.1, $(\mathbb{G}_2, \mathcal{M}) \vDash_{Ch',\nu} \psi$. D'où $(\mathbb{G}_2, \mathcal{M}) \vDash \mathbf{A}\psi$.

• φ est de la forme $\mathbf{E}\psi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash \mathbf{E}\psi$, c'est-à-dire que pour tout $\nu : V \to \mathcal{M}$, il existe $ch' \in Path_{q_0}(\mathbb{G}_2)$ tel que $(\mathbb{G}_2, \mathcal{M}) \vDash \psi$.

Soit $\nu : V \to \mathcal{M}$. On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash \mathbf{E}\psi$, donc il existe $ch \in Path_{q_0}(\mathbb{G}_1)$ tel que $(\mathbb{G}_1, \mathcal{M}) \vDash_{ch,\nu} \psi$. On pose ch' un chemin raffinant de ch. D'après le théorème 3.3.1, $(\mathbb{G}_2, \mathcal{M}) \vDash_{ch',\nu} \psi$. D'où $(\mathbb{G}_2, \mathcal{M}) \vDash \mathbf{E}\psi$.

Récurrence. On suppose que la propriété est vraie pour toute formule de $\delta Sen(\delta\Sigma)$ de taille inférieure ou égale à N. On va montrer qu'elle est vraie pour toute formule de taille $N + 1$.

• φ est de la forme $\psi \wedge \chi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash \psi \wedge \chi$, c'est-à-dire que $(\mathbb{G}_2, \mathcal{M}) \vDash \psi$ et $(\mathbb{G}_2, \mathcal{M}) \vDash \chi$.

On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash \psi \wedge \chi$, c'est-à-dire que $(\mathbb{G}_1, \mathcal{M}) \vDash \psi$ et $(\mathbb{G}_1, \mathcal{M}) \vDash \chi$. Par hypothèse de récurrence, comme $(\mathbb{G}_1, \mathcal{M}) \vDash \psi$, alors $(\mathbb{G}_2, \mathcal{M}) \vDash \psi$, et comme $(\mathbb{G}_1, \mathcal{M}) \vDash \chi$, alors $(\mathbb{G}_2, \mathcal{M}) \vDash \chi$. Donc $(\mathbb{G}_2, \mathcal{M}) \vDash \psi \wedge \chi$.

• φ est de la forme $\psi \vee \chi$. Montrons que $(\mathbb{G}_2, \mathcal{M}) \vDash \psi \vee \chi$, c'est-à-dire que $(\mathbb{G}_2, \mathcal{M}) \vDash \psi$ ou $(\mathbb{G}_2, \mathcal{M}) \vDash \chi$.

On sait que $(\mathbb{G}_1, \mathcal{M}) \vDash \psi \vee \chi$, c'est-à-dire que $(\mathbb{G}_1, \mathcal{M}) \vDash \psi$ ou $(\mathbb{G}_1, \mathcal{M}) \vDash \chi$. Si $(\mathbb{G}_1, \mathcal{M}) \vDash \psi$, alors par hypothèse de récurrence, $(\mathbb{G}_2, \mathcal{M}) \vDash \psi$, donc $(\mathbb{G}_2, \mathcal{M}) \vDash \psi \vee \chi$. Si $(\mathbb{G}_1, \mathcal{M}) \vDash \chi$, alors par hypothèse de récurrence, $(\mathbb{G}_2, \mathcal{M}) \vDash \chi$, donc $(\mathbb{G}_2, \mathcal{M}) \vDash \psi \vee \chi$. □

Ainsi ce théorème montre que la classe des EIOLTS est stable par la relation de raffinement correct fortement.

Conclusion

Dans ce rapport, nous avons défini dans un premier temps une théorie du raffinement pour les EIOLTS. Nous avons montré que ce raffinement possède la propriété d'être transitif, et que la correction et la complétude du raffinement sont conservées par transitivité. Ces résultats nous permettent de construire incrémentalement, à partir de la spécification initiale, l'EIOLTS représentant l'implantation réelle du système, tout en assurant à chaque étape la conservation de la conformité.

Dans un deuxième temps, nous avons défini un formalisme axiomatique dédié à la spécification des systèmes réactifs. Ce formalisme nous permet d'abstraire les comportements des systèmes réactifs en considérant les EIOLTS, non plus comme des spécifications du système, mais comme des modèles. Nous avons alors montré que la classe des modèles associés à une spécification dans ce formalisme est stable par relation de raffinement correct fortement.

Nous avons également, dans les deux dernières semaines de ce stage, effectué un travail qui n'a pas pu être inclus dans ce rapport. Nous avons en effet, en utilisant l'outil AGATHA du CEA, défini un algorithme qui vérifie qu'un EIOLTS est un raffinement correct d'un autre. AGATHA est un outil d'aide à la validation de spécifications écrites dans un langage se traduisant en EIOLTS. Il permet de construire une structure arborescente, appelé arbre d'exécution symbolique, représentant, de façon abstraite, tous les comportements possibles décrits par la spécification. Chaque chemin de cet arbre dénote un ensemble de chemins numériques caractérisés par des contraintes déduites de la spécification. Cette construction est obtenue par exécution symbolique, combinée avec des techniques de réduction à la volée qui évitent le calcul de comportements redondants. Il a été prouvé dans [RGLG03], que l'arbre d'exécution symbolique est équivalent comportementalement à l'EIOLTS initial (ils sont bisimilaires), ce qui permet de raisonner sur cet arbre aussi bien que sur l'EIOLTS qu'il représente. Pour comparer un EIOLTS donné $\mathbb{G}_1$ et l'EIOLTS $\mathbb{G}_2$ dont on veut vérifier qu'il est un raffinement correct de $\mathbb{G}_1$, notre algorithme utilise donc les arbres générés par AGATHA à partir de $\mathbb{G}_1$ et $\mathbb{G}_2$, qu'on note $Ag(\mathbb{G}_1)$ et $Ag(\mathbb{G}_2)$. On vérifie ensuite que pour chaque chemin de $Ag(\mathbb{G}_2)$ liant deux points de contrôle du vocabulaire de $Ag(\mathbb{G}_1)$, il existe une transition dans $Ag(\mathbb{G}_1)$ que ce chemin raffine. Plus précisément, on vérifie que les chemins numériques dénotés par ce chemin symbolique sont bien des comportements décrits par la transition initiale (cf la définition de la correction du raffinement d'une transition 2.3.5).

Une première extension de ce travail serait d'étendre le raffinement aux noms de canaux. En effet, on impose, dans la définition du raffinement d'une transition, de retrouver l'action de communication de la transition dans chacun des chemins de l'EIOLTS qui la raffine. Or on voudrait aussi pouvoir « éclater » les noms de canaux. Par exemple, on voudrait pouvoir raffiner la transition, étiquetée par $Billet!M$, qui envoie les billets correspondant au montant demandé en un chemin qui calcule le nombre n de billets de 10 et le nombre m de billets de 20 et qui envoie sur deux canaux différents ces résultats, en étiquetant deux transitions par $Billets10!n$ et $Billets20!m$. Cela permettrait d'enrichir le raffinement.

La perspective directe de ce travail serait d'étendre ces résultats à la composition d'EIOLTS. En effet, un système réactif est en général modélisé par un ensemble d'EIOLTS communicants entre eux, soit de manière synchrone par rendez-vous, soit de manière asynchrone par files d'attente. Il faudrait alors assurer que raffiner le système revient à raffiner chacun de ses modules. En d'autres termes, il faudrait assurer que composer les raffinements de chacun des EIOLTS est équivalent à raffiner la composition des EIOLTS initiaux, tout en assurant la conservation de la correction et de la complétude. Du point de vue de notre formalisme, il faudrait déterminer quelles sont les propriétés qui peuvent être conservées par composition. En effet, lorsqu'on compose plusieurs EIOLTS, les communications que l'on synchronise disparaissent, les propriétés de plongement d'un terme dynamique dans un chemin peuvent alors ne plus être vérifiées. Il faudrait donc étudier la nature des propriétés que la composition peut conserver, et les conditions de cette conservation.

Le travail de ce stage s'inscrivait initialement dans le cadre d'un projet RNRT (Réseau National de Recherche en Télécommunications) sur la Spécification et le Test, Abstraits et Compositionnels, de Systèmes (STACS). Ce projet s'effectue en collaboration entre le LaMI, le CEA, Thalès Communications et Ligeron SA. Son objectif est de « développer une méthodologie formelle de test utilisant des mécanismes de structuration et d'abstraction dans le but de rendre faisable le passage à l'échelle de ces techniques de vérification ainsi que de diminuer le coût de ces tests et de permettre leur réutilisation ». Le travail présenté dans ce rapport s'incrit dans ce projet dans le sens où la définition d'un formalisme axiomatique dont la sémantique est fondée sur les EIOLTS serait un point de départ pour faire du test de propriétés dynamiques. Actuellement, les travaux sur le test de systèmes réactifs portent essentiellement sur le test de conformité de l'implantation du système par rapport à sa spécification, les deux étant définies par des automates [CA97] [Jér01]. Les techniques de test de propriétés ont essentiellement été définies dans le cadre de formalisme statique de type spécifications algébriques [BGM91][ALG02].

Bibliographie

[AGM92] S. Abramsky, Dov M. Gabbay T. S. E. Maibaum, *Handbook of logic in computer science*, Vol. 2, chapitre Modal and temporal logics, pages 504–526, Oxford University Press, 1992.

[ABP02] M. Aiguier, F. Barbier, P. Poizat. *A logic for mixed specifications*, Rapport de Recherche n° 73-2002, LaMI 2002.

[ALG02] A. Arnould, P. Le Gall. *Test de conformité : une approche algébrique*, Technique et science informatique, Vol. 21, n° 9, 2002.

[BGM91] G. Bernot, M.-C. Gaudel, B. Marre. *Software testing based on formal specifications : a theory and a tool*. Software Engineering Journal, pages 387–405, 1991.

[CA97] A. Cavalli, R. Anido. *Verification and testing techniques based on the finite state machine model*, Rapport de recherche n° 97, 1997.

[Jér01] T. Jéron. *Le test de conformité : état de l'art*, Rapport pour l'AAE (Architecture Électronique Embarquée), 2001.

[Lyn88] N. A. Lynch. *I/O automata : a model for discrete event systems*, Proceeding of 22^{nd} Conference on Information Sciences and Systems, 1988.

[Pha94] M. Phalippou. *Relations d'implantations et hypothèses de test sur les automates à entrée et sorties*, Thèse de Doctorat, Université de Bordeaux, 1994.

[RGLG03] N. Rapin, Ch. Gaston, A. Lapitre, J.-P. Gallois. *Behavioral unfolding of formal specifications based on communicating automata*, Workshop on Automated Technology for Verification and Analysis, 2003.

[Tre95] J. Tretmans. *Testing labelled transition systems with inputs and outputs*, 8^{th} International Workshop on Protocols Test Systems, 1995.