SLR204

Bases de la Vérification des Systèmes Répartis

Notes de cours sur les Systèmes de Transitions Etiquetés

Elie Najm

Elie.Najm@telecom-paristech.fr

46, Rue Barrault 75013 Paris

Janvier 2017

1. Introduction

Les systèmes de transition étiquetés (STE) constituent un domaine sémantique intéressant pour les langages de qui traitent explicitement de la concurrence, de la communication et de la coopération. Dans ces langages, le sens d'une spécification est donné par sa traduction dans un STE. Nous présentons les concepts théoriques d'une approche pour la vérification des systèmes répartis basée sur les STE. Nous explorons les propriétés des STE ainsi que des critères de comparaison et d'abstraction qui peuvent leur être associés.

2. Les systèmes de transitions étiquetés

2.1 Définition

Un système de transitions étiqueté (STE) est un quadruplet:

$$<q_0, Q, A_\tau, T>$$

où :

- Q est un ensemble d'états. Les symboles, q, q', q_0, q_1, ..., p, p', p_0, p_1, ... désignerons des éléments de cet ensemble,

- q_0 est un état particulier de Q, l'état initial, $p \xrightarrow[T]{a} p'$

- A_τ un ensemble d'actions. A_τ contient une action particulière, τ, qui désigne l'action interne (appelée aussi action silencieuse). Nous désignerons par A l'ensemble des actions différentes de τ. Donc $A =_{def} A_\tau - \{\tau\}$. Les actions de A sont dites observables. Les symboles a, a', a_0, a_1, ..., b, b', b_0, b_1, ... désignerons des éléments de A. Les symboles α, α', α_0, α_1, ..., β, α', α_0, α_1, ... désignerons des éléments de A_τ.

- T désigne l'ensemble des transitions. T est une relation ternaire : $T \subseteq Q \times A_\tau \times Q$. Nous écrirons $p \xrightarrow[T]{a} p'$ pour désigner le fait que la transition $(p, \alpha, p') \in T$ et lorsqu'il n'y a pas d'ambigüité, nous ne mentionnerons pas l'ensemble de référence T, et écrirons $p \xrightarrow{a} p'$

Les STE peuvent être représentés visuellement par des graphes où les états sont représentés par les nœuds, les transitions sont matérialisées par les arcs, et les actions sont des étiquettes apposées sur les arcs. Exemple :

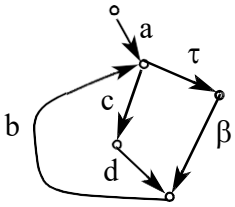

Fig. 1 : représentation graphique d'un STE

2.2 Rôle de l'action silencieuse

Les systèmes que nous considérons sont par nature non déterministes. Parmi les causes du non déterminisme : les pannes matériel ou logiciel de certains éléments, les erreurs de transmission, les ruptures de liaisons de communication, etc ... Ce non déterminisme doit être reflété dans les modèles analysés. Dans un STE, le non déterminisme peut être représenté de deux façons :

- par l'existence de deux (ou plus) transitions partant d'un même état, portant la même étiquette et ayant des états suivants avec des comportements différents, ou
- par l'existence d'une transition étiquetée par l'action silencieuse.

Le non déterminisme est nécessaire, par exemple, pour de la modélisation des médiums de communication non fiables. La modélisation de ce type de médium est très importante pour la vérification des protocoles de communication. Voici 4 exemples simples de modélisation en STE de médiums : fiable (a) et non fiables (b, c et d).

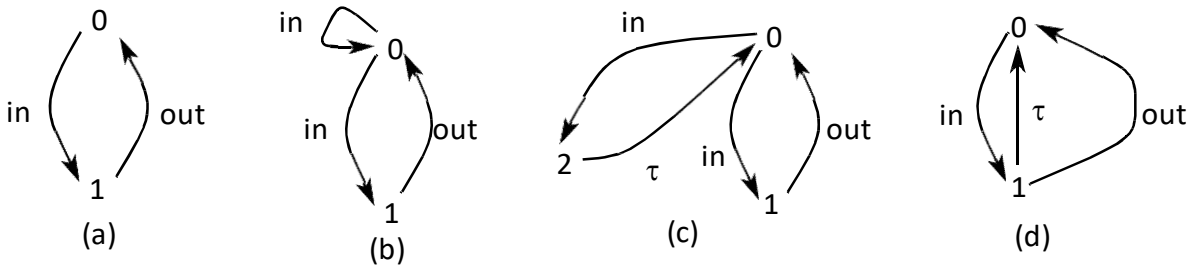

Fig. 2 : Représentations en STE de médiums fiable (a) et non fiables (b, c et d)

Le comportement du STE (a) est tel que chaque transition, in, est suivie par une transition out. Ce comportement représente bien un médium parfait (ici il s'agit d'une simplification, car le médium est considéré de capacité 1, çàd, un deuxième message ne peut être déposé dans le médium que si le message précédent a déjà été reçu). Les comportements décrits par les STE (b), (c) et (d) ont une propriété en commun : à un moment donné, la différence entre le nombre d'actions d'émission (in) et le nombre d'actions de réception (out) est un entier positif non borné. Cette différence représente le nombre de messages perdus. Ces trois STE, néanmoins, se distinguent par la représentation du moment de la perte du message. Dans le cas (b), les messages se perdent juste au moment du dépôt du message dans le médium, (transition $0 \xrightarrow{in} 0$). Idem pour (c) (transition $0 \xrightarrow{in} 2$), mais, dans ce cas, il existe un état (l'état 2) qui représente l'état du médium au moment où il vient de perdre le message, la transition silencieuse (çàd, celle étiquetée par τ) remet ce STE dans l'état où il peut accepter d'autres messages. Dans le STE (d), c'est la transition silencieuse qui, en se déclenchant, modélise, à la fois, la perte du message et le retour à l'état où d'autres messages peuvent être acceptés par le canal.

Il est à noter que l'action τ peut revêtir une autre signification : un STE peut modéliser le comportement d'un système à un certain niveau d'abstraction, et τ représente les actions internes à ce système (par exemple, une interaction entre des composants de ce système), et dont le contenu n'est pas pertinent pour le niveau d'abstraction considéré.

2.3 Abstraction des transitions silencieuses

Considérons un STE, nous nous intéressons à définir, entre les états de ce STE, une nouvelle relation de transition (notée : $\Rightarrow$) qui fait abstraction de l'action silencieuse. Nous considérons un nouveau symbole $\varepsilon \notin A_\tau$. Nous noterons par A_ε l'ensemble $A \cup \{\varepsilon\}$.

La relation $\Rightarrow$ est définie sur Q x A_ε x Q comme suit :

(i) soit p et p' deux états, alors $p \overset{\varepsilon}{\Longrightarrow} q$ si et seulement si, soit p = q, soit il existe une chaîne de transitions silencieuses, commençant par l'état p et finissant par l'état q. Plus formellement :

$$p \overset{\varepsilon}{\Longrightarrow} q \overset{\text{def}}{=} \begin{cases} p = q \\ \text{ou} \\ \exists\, p_1, \ldots, p_n \;\; \text{avec} \;\; p_1 = p \;,\; p_n = q \;\; \text{et} \; \forall\, i \; (1 \leq i < n) : p_i \xrightarrow{\tau} p_{i+1} \end{cases}$$

(ii) soit p et q deux états et a ∈ A, alors $p \overset{a}{\Longrightarrow} q$ si, et seulement si, il existe une chaîne de transitions, toutes silencieuses sauf une étiquetée par a, et qui commence en p et finit en p'. Plus formellement :

$$p \overset{a}{\Longrightarrow} q \overset{\text{def}}{=} \exists\, p', q' \;\; \text{tels que} \;\; p \overset{\varepsilon}{\Longrightarrow} p' \,, p' \xrightarrow{a} q' \,, q' \overset{\varepsilon}{\Longrightarrow} p$$

D'après ce qui précède, il est aisé de constater que $\forall\, p : p \overset{\varepsilon}{\Longrightarrow} p$

2.4 Action possible en un état

Soit un état p et une action observable a. L'action a est dite possible en p si et seulement si ∃ q tel que $p \overset{a}{\Longrightarrow} q$. Nous écrirons $p \overset{a}{\Longrightarrow}$ pour signifier que a est possible en p.

2.5 Relation de transition étiquetée par les séquences d'actions

Nous étendons la définition de $\Rightarrow$ aux séquences d'actions. Nous noterons par A* l'ensemble des séquences finies d'éléments de A (ensemble appelé aussi le langage des mots finis sur l'alphabet A). Un élément σ de A* est donc soit la séquence vide, notée ε, soit une séquence de la forme $a_1 \ldots a_n$ où ($n \geq 1$) et $\forall\, i : a_i \in A$. Nous construisons une nouvelle relation notée $\Rightarrow^*$ et définie sur Q x A* x Q comme suit.

- Pour σ = ε: $p \xrightarrow{\varepsilon}{}^{*} q \overset{\text{def}}{=} p \xrightarrow{\varepsilon} q$
- Pour $\sigma = a_1 \ldots a_n$: $p \xrightarrow{\sigma}{}^{*} q \;\; \overset{\text{def}}{=} \;\; \exists \;\; p_1{=}p, \ldots, p_n{=}q$, tels que $\forall\, i : p_i \xrightarrow{a_i} p_{i+1}$

2.6 Traces possibles à partir d'un état

Soit un état p et $\sigma \in A^*$.

Nous dirons que σ est une trace possible de p si et seulement si : $\exists$ q avec : $p \xRightarrow{\sigma}{}^{*} q$

Nous écrirons $p \xRightarrow{\sigma}{}^{*}$ pour signifier que σ est une trace possible de p

2.7 Accessibilité

Nous définissons la relation d'accessibilité, notée $\Longrightarrow^{*}$, entre les états d'un STE, comme suit :

$p \Longrightarrow^{*} q \stackrel{def}{=} \exists \sigma \in A^* \; p \xRightarrow{\sigma}{}^{*} q$

Lorsque la relation $\Longrightarrow^{*}$ est établie entre p et q, ou, en d'autres termes, si $p \Longrightarrow^{*} q$, on dira que q est accessible à partir de p.

2.8 Blocages

Nous avons besoin des notations suivantes pour définir la notion de blocage :

- $p \xrightarrow{\alpha}$ $=_{def} \exists$ p' tel que $p \xrightarrow{\alpha} p'$
- $p \not\xrightarrow{\alpha}$ $=_{def} \nexists$ p' tel que $p \xrightarrow{\alpha} p'$
- $p \not\longrightarrow$ $=_{def} \nexists$ p' , $\nexists$ $\alpha \in A_\tau$, tels que $p \xrightarrow{\alpha} p'$

Il est aisé, en imitant les définitions données ci-dessus pour la relation $\longrightarrow$, de définir les notations : $p \xRightarrow{a}\!\!\!\!\!/\;$ et $p \xRightarrow{\sigma}{}^{*}\!\!\!\!\!/\;$

Définition : blocage total (deadlock)

Un STE possède un blocage total lorsque : $\exists$ p, accessible à partir de q_0, $p \not\longrightarrow$

Définition : blocage vivant (livelock)

Un STE possède un blocage vivant ssi $\exists$ ($n \geq 1$), $\exists$ p_1, ..., $p_n \in Q$ et accessibles à partir de q_0 avec :

$$\begin{cases} \forall\, i, \forall\, a \in A, \quad p_i \xRightarrow{a}\!\!\!\!\!/\; \\ \text{et} \\ \forall\, i, \exists\, j, \text{ tel que } \quad p_i \xrightarrow{\tau} p_j \end{cases}$$

En d'autres termes, un STE possède un blocage vivant lorsqu'il existe une boucle de transitions silencieuses qui relie des états qui ne possèdent pas d'autres transitions sortantes que les transitions silencieuses.

3. Equivalences entre systèmes de transition étiquetés

Vérifier un système revient à examiner la concordance entre différentes descriptions données pour ce système. En général, ces descriptions peuvent prendre plusieurs formes et être faites dans des notations différentes. Exemples de descriptions : énoncés de propriétés (dans une logique temporelle), modèles de comportement (spécifiés dans un langage tel CCS+ ou SDL) à un niveau de détail plus ou moins important. Nous nous plaçons ici dans le cas où nous avons, pour un même système à l'étude, des modèles de comportement différents, donnés sous forme de STE, et nous cherchons à définir des critères permettant de conclure sur la concordance entre ces modèles. Le cas typique est, par exemple, l'examen de la concordance entre un protocole et le service qu'il est censé fournir. Plusieurs critères de comparaison existent avec chacun son pouvoir discriminatoire. Nous allons les explorer dans les paragraphes qui suivent.

Soit deux STE $S_1 = <q_1, Q_1, A_{1\tau}, T_1>$ et $S_2 = <q_2, Q_2, A_{2\tau}, T_2>$. Notons que les deux ensembles des actions de ces deux STE utilisent le même symbole pour l'action silencieuse, τ. Il est aisé de ramener la comparaison de ces deux STE à la comparaison entre deux états d'un même STE. En effet, si $Q_1 \cap Q_2 = \varnothing$, nous pouvons former deux nouveaux STE comme suit. Notons $A_\tau = (A_1 \cup A_2) \cup \{\tau\}$, $Q = Q_1 \cup Q_2$, $T = T_1 \cup T_2$. Nous définissons : $S'_1 = <q_1, Q, A_\tau, T>$ et $S'_2 = <q_2, Q, A_\tau, T>$. Il est aisé de constater que, en prenant $i, j \in \{1, 2\}$:

pour $(p,p') \in Q_i \times Q_i$ l'on a $\quad p \xrightarrow[T]{\alpha} p' \quad$ ssi $\quad p \xrightarrow[T_i]{\alpha} p'$

et

pour $i \neq j$, $\forall (p,p') \in Q_i \times Q_j$: $\quad \nexists \ \alpha \in A_\tau$ tel que $\quad p \xrightarrow[T]{\alpha} p'$

Sachant que la condition $Q_1 \cap Q_2 = \varnothing$ est toujours réalisable (il suffit de renommer de façon disjointe les états des deux STE pour que cette condition soit satisfaite), nous pouvons donc conclure que l'étude de deux STE se ramène toujours à l'étude de deux états d'un même STE. Par la suite, donc, nous considérons un seul STE de référence et nous baserons notre discussion sur le comportement associé à ses états.

3.1 Equivalences de traces

Un premier critère de comparaison naturel entre états d'un STE consiste à examiner les séquences d'actions observables à partir de chacun de ces deux états. Il s'agit de l'équivalence dite de traces : deux états q_1 et q_2 sont trace-équivalents, noté $q_1 \sim_T q_2$, ssi ils ont les mêmes séquences d'actions observables tirables à partir de chacun de ces deux états. Plus formellement :

$$q_1 \sim_T q_2 =_{def} \forall\, \sigma \in A^* \quad q_1 \xRightarrow{\sigma *} \ <=> \ q_2 \xRightarrow{\sigma *}$$

La relation $\sim_T$ ne garantit pas l'équivalence de blocage. Voici (fig. 3) 4 STE trace-équivalents, dont trois contiennent un blocage accessible de l'état initial alors que le quatrième est sans blocage.

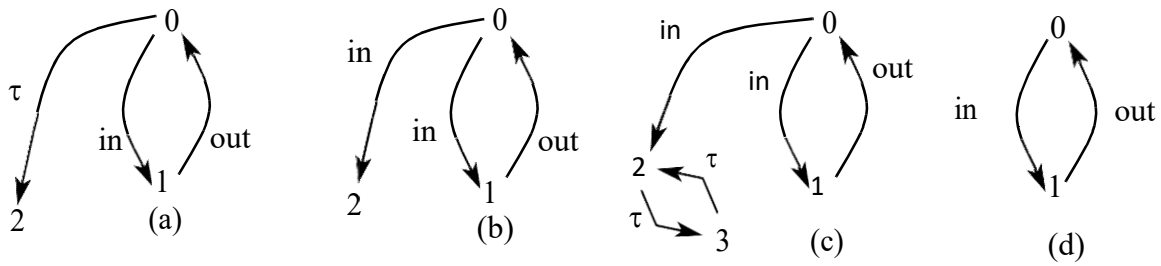

Fig. 3 : quatre STE trace-équivalents
(a), (b) et (c) sont bloquants alors que (d) est sans blocage

3.2 L'équivalence de traces maximales

Une manière de remédier à la faiblesse de la définition précédente consiste à comparer non seulement les traces, mais aussi les traces maximales, çàd celles qui débouchent sur des états bloquants (blocage total ou vivant). Si nous notons par A^ω l'ensemble des séquences infinies d'actions de A, avant de définir cette nouvelle relation d'équivalence, nous avons besoin d'étendre la définition de $\Rightarrow$ aux séquences infinies d'actions. Nous construisons une nouvelle relation, notée $\Rightarrow^\omega$ et définie sur Q x A^ω comme suit (où $\sigma = a_1 \ldots a_n \ldots$) :

$$p \xRightarrow{\sigma}{}^{\omega} =_{def} \exists \text{ une suite infinie } p_1=p, p_2, \ldots, p_n, \ldots \text{ telle que } \forall\, i : p_i \xRightarrow{a_i} p_{i+1}$$

Nous définissons maintenant l'équivalence (il est aisé de prouver que c'est une équivalence) de trace-maximales, notée $\sim_M$, comme suit :

$$q_1 \sim_M q_2 =_{def} \begin{cases} \forall\, \sigma \in A^\omega : \; q_1 \xRightarrow{\sigma}{}^{\omega} \iff q_2 \xRightarrow{\sigma}{}^{\omega} \\ \text{et} \\ \forall\, \sigma \in \Lambda^* : \; (\exists\, p_1 : q_1 \xRightarrow{\sigma}{}^{*} p_1 \text{ et } p_1 \xRightarrow{a} \forall\, a \in \Lambda) \\ \qquad\qquad \iff \\ \qquad\qquad (\exists\, p_2 : q_2 \xRightarrow{\sigma}{}^{*} p_2 \text{ et } p_2 \xRightarrow{a} \forall\, a \in A) \end{cases}$$

Cette équivalence préserve la propriété de blocage : si $q_1 \sim_M q_2$ et si p_1 est un état de blocage accessible de q_1, alors il existe un état de blocage p_2 accessible de q_2.

Néanmoins, la trace-maximale équivalence possède un inconvénient : elle n'est pas compositionnelle. En effet, il existe plusieurs opérations de composition de STE et il est souhaitable que la relation d'équivalence choisie soit préservée par ces opérations. Plus concrètement, si op est une opération binaire sur les STE, l'on souhaite, pour la relation de comparaison choisie ~ , qu'elle vérifie la propriété :

$$p_1 \sim p_2 \Rightarrow \forall\, q : (q \text{ op } p_1) \sim (q \text{ op } p_2)$$

Si l'on considère l'opérateur de restriction (vu dans le cours sur CCS+) il est facile de constater, sur un exemple (fig 4), que la relation $\sim_M$ n'est pas compositionnelle pour cet opérateur.

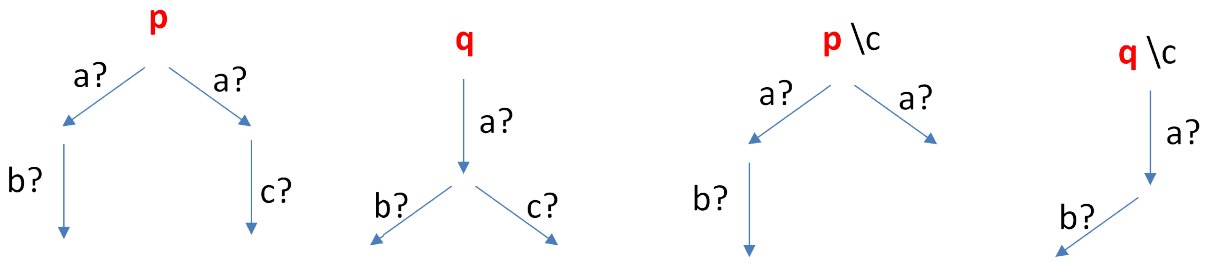

Fig. 4 : p ~M q mais ¬ p \c ~M q \c

En effet, les deux STE p et q de la figure 4 ont le même ensemble de traces maximales constitué de 2 traces: {a? b?, a? c ?}. Alors que l'ensemble des traces maximales de p\c et q\c sont différents. Les traces maximales de p\c sont {a? b?, a?} et les traces maximales de q\c sont {a? b?}.

Il faut donc choisir une autre relation de comparaison qui garantisse la propriété de compositionnalité pour les opérations importantes sur les STE (mise en parallèle, restriction, renommage des actions, choix, ...).

3.3 L'équivalence de test

Si l'on examine l'exemple de la figure 4, l'on constate que p et q, bien que produisant les mêmes traces maximales, n'ont pas cette propriété après avoir accompli chacun une action similaire. Par exemple, après l'action a?, p ne réussit pas à faire l'action b? à tous les coups alors que q, lui, réussit. Donc, un critère de comparaison plus fin consiste à non seulement examiner les traces maximales, mais à considérer aussi les capacités de réussite et d'échec après l'accomplissement d'une séquence d'action. Une des équivalences qui met en œuvre ce type de critères est l'équivalence de test.

Soit L un ensemble d'actions, $L \subseteq A$, et p un état du STE considéré. On définit la relation p doit L comme suit :

$$p \text{ doit } L \quad \text{ssi} \quad \forall p' \text{ tel que : } p \xRightarrow{\varepsilon} p', \exists a \in L \text{ tel que } p' \xRightarrow{a}$$

Un test est un couple (σ, L) où $\sigma \in A^*$ et $L \subseteq A$. Si p est un état, l'on définit la relation p réussit (σ, L) comme suit :

$$p \text{ réussit } (\sigma, L) \quad \text{ssi} \quad \forall p' : p \xRightarrow{\sigma} p' \text{ alors } p' \text{ doit } L$$

Nous pouvons maintenant définir l'équivalence de test (que nous notons $\sim_t$) :

$$q_1 \sim_t q_2 \quad =_{def} \quad \forall (\sigma, L) : q_1 \text{ réussit } (\sigma, L) \text{ ssi } q_2 \text{ réussit } (\sigma, L)$$

La Figure 5 donne deux exemples de STE test équivalents. Alors que si nous reconsidérons les exemples de la figure 4, nous avons que $\neg(p \sim_t q)$. Pour s'en convaincre, il suffit d'appliquer le test (a?, {b?}) à p et q. Nous avons que (q réussit (a?, {b?})) mais $\neg$ (p réussit (a?, {b?})).

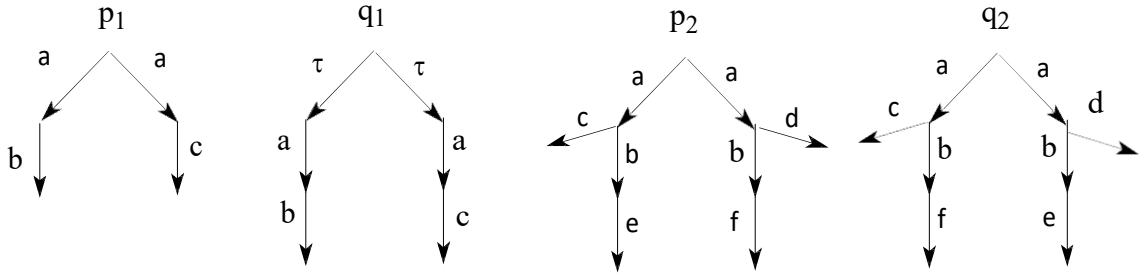

Fig. 5 : $p_1 \sim_t q_1$ et $p_2 \sim_t q_2$

L'on peut aisément prouver que $\sim_t$ est effectivement une équivalence et qu'elle est compositionnelle avec un bon nombre d'opérateurs sur les STE (et notamment ceux de CCS+). L'équivalence de test est donc un bon critère de comparaison entre STE, suffisamment discriminatoire tout étant compositionnelle. L'équivalence de test, néanmoins, a deux inconvénients : (i) la complexité de l'algorithme de décision, car il faut construire et explorer l'ensemble des tests, et (ii) l'information apportée par l'algorithme de décision ne renseigne que sur les deux états considérés et pas sur l'équivalence ou la non équivalence des autres états du STE. Or, cette connaissance est nécessaire pour réduire les STE à des STE équivalents ayant un plus petit nombre d'états (en regroupant les états équivalents dans un même état).

2.4 Les relations de simulation et de bisimulation

Nous explorons maintenant une famille de relations entre STE qui impliquent tous les états accessibles. Nous aborderons d'abord des relations qui ne font pas abstraction de l'action silencieuse τ.

2.4.1 simulation et équivalence de simulation

Une première relation que nous considérons est une relation non symétrique appelée simulation. Soit une relation $S \subseteq Q \times Q$. S est dite une simulation ssi elle satisfait la propriété suivante :

$q\ S\ p \Rightarrow \forall\ \alpha \in A_\tau$ et $\forall p' \in Q$ tels que $p \xrightarrow{\alpha} p'$ alors $\exists\ q'$ tel que $\xrightarrow{\alpha} q'$ et $p'\ S\ q'$

Cette propriété se représente graphiquement :

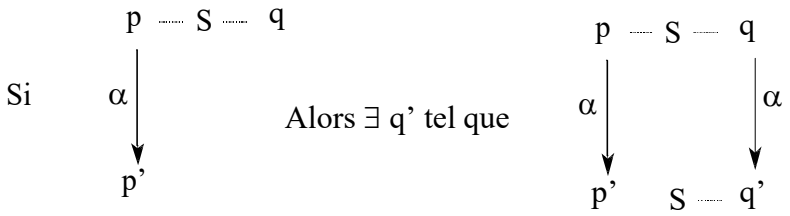

Fig. 6 : Propriété caractéristique des relations de simulation

Nous dirons que q <u>simule</u> p ssi il existe une relation de simulation S avec q S p.
Nous voyons que lorsque q <u>simule</u> p alors toute transition de p est imitée par une transition de q et les états suivants sont de nouveaux en relation de simulation. L'on peut prouver que <u>simule</u> est un préordre (réflexive et transitive) sur Q. L'on peut dériver de <u>simule</u> une relation d'équivalence que nous noterons $\sim_S$ de la façon suivante :

$$p \sim_S q \quad \text{ssi} \quad p \text{ simule } q \quad \text{et} \quad q \text{ simule } p$$

et, dans ce cas, nous dirons que p et q sont similaires.

2.4.2 bisimulation

Bien que la relation $\sim_S$ permette de mettre en relation les états de Q de façon récurrente, $\sim_S$ n'est pas satisfaisante car elle ne préserve pas les blocages. Voici un exemple (Fig. 7) de deux STE similaires mais, néanmoins, l'un est avec blocage et l'autre sans.

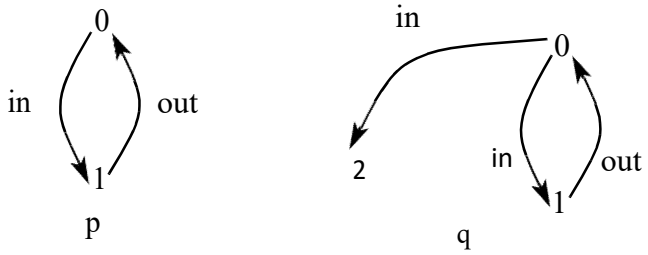

Fig. 7 : p et q sont similaires mais q possède un blocage et p n'en possède pas

Il est donc utile de définir une nouvelle relation symétrique de simulation qui soit plus forte. Nous définissons d'abord une propriété qui caractérise une famille de relations dite de bisimulation. Une relation $B \subseteq Q \times Q$ est une bisimulation ssi B et B^{-1} sont des simulations. De façon plus explicite, nous dirons qu'une relation B est une bisimulation ssi elle satisfait la propriété suivante :

p B q =>

$\forall \alpha \in A_\tau$ et $\forall p' \in Q$ tels que $p \xrightarrow{\alpha} p'$ alors $\exists$ q' tel que $q \xrightarrow{\alpha} q'$ et p' B q'

et symétriquement

$\forall \alpha \in A_\tau$ et $\forall q' \in Q$ tels que $q \xrightarrow{\alpha} q'$ alors $\exists$ p' tel que $p \xrightarrow{\alpha} p'$ et p' B q'

Cette propriété se représente graphiquement (Fig 9) :

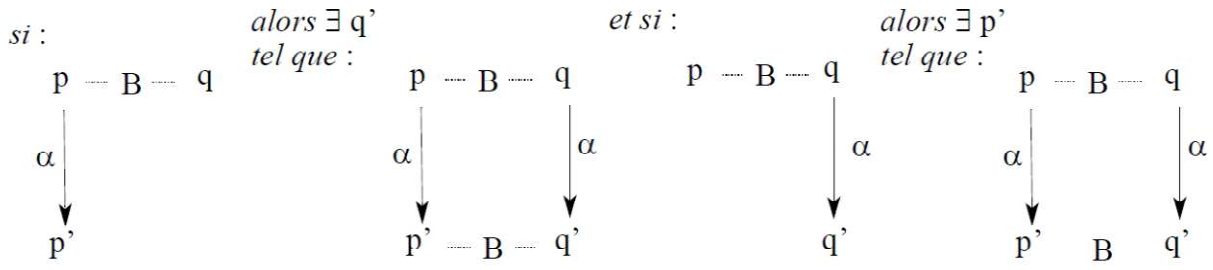

Fig. 9 : Propriété caractéristique des relations de bisimulation

Nous pouvons définir la relation de bisimulation :

p ~ q ssi ∃ une relation de bisimulation B avec p B q

En d'autres termes, p et q sont bisimilaires ssi il est possible de construire une relation de bisimulation entre p et ses états dérivés d'une part et q et ses états dérivés d'autres part. Voici (Fig10) un exemple de deux STE bisimilaires :

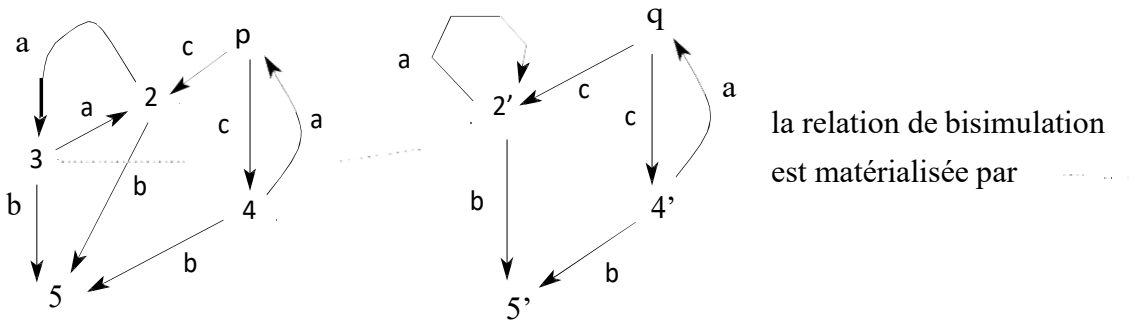

Fig. 10 : exemple de bisimulation: p ~ q

2.4.3 un algorithme de construction de bisimulations

L'algorithme suivant permet de partitionner les états d'un STE avec chaque partie contenant des états bisimilaires. La partition de départ de cet algorithme est la partition qui est composée d'une seule partie qui est l'ensemble Q. Cet algorithme procède par étapes, et dans chaque étape la partition de l'étape précédente est raffinée jusqu'à l'obtention d'une partition non raffinable.

Soit $\Pi^j_1, ..., \Pi^j_{nj}$ la partition de Q obtenue à l'étape j. Cette partition contient n_j parties disjointes. En utilisant cette notation, la partition de départ de l'algorithme s'écrit $\Pi^1_1 = Q$. Nous noterons $\Pi^j(p)$ la partie unique de la partition j à laquelle appartient l'état p. Le raffinement de la partition j, et donc l'obtention de la partition j+1, s'obtient en examinant les couples d'états de chaque partie Π^j_i pour décider s'ils sont dans une même partie de la partition j+1. Deux états p et q appartenant à une partie Π^j_i appartiennent à une même partie de la partition j+1 ssi :

$$\forall\, \alpha \in A_\tau \text{ et } \forall p' \in Q \text{ tels que } \quad p \xrightarrow{\alpha} p' \text{ alors } \exists\, q' \text{ tel que } \quad q \xrightarrow{\alpha} q' \text{ et } q' \in \Pi^j(p')$$

et symétriquement

$$\forall\, \alpha \in A_\tau \text{ et } \forall q' \in Q \text{ tels que } \quad q \xrightarrow{\alpha} q' \text{ alors } \exists\, p' \text{ tel que } \quad p \xrightarrow{\alpha} p' \text{ et } p' \in \Pi^j(q')$$

Le critère d'arrêt pour cet algorithme est lorsque deux partitions successives sont égales. Ceci se produit forcément lorsque le nombre des états Q est fini.

Appliquons cet algorithme aux états du STE p dans lafigure10.

Nous avons : $\Pi^1_1 = \{p, 2, 3, 4, 5\}$

L'examen des couples d'états permet d'en dériver la nouvelle partition :

$\Pi^2_1 = \{p\}$, $\Pi^2_2 = \{2, 3, 4\}$, $\Pi^2_3 = \{5\}$

Un examen des couples d'états de chaque partie de la partition précédente permet d'obtenir la partition :

$\Pi^3_1 = \{p\}$, $\Pi^3_2 = \{2, 3\}$, $\Pi^3_3 = \{4\}$, $\Pi^3_4 = \{5\}$

Un examen des couples d'états de chaque partie de cette dernière partition permet d'obtenir une partition identique. Donc l'algorithme s'arrête et nous pouvons conclure que les seuls états bisimilaires de Q sont 2 et 3 car ce sont les seuls états qui se retrouvent dans une même partie.

2.4.4 réduction des STE

L'algorithme précédent permet aussi de trouver un STE minimal (en nombre d'états) bisimilaire à un STE donné. L'on procède de la façon suivante :

- chaque partie de la partition finale, $\Pi^j_1, ..., \Pi^j_n$, est considérée comme un état,
- $\Pi^j_i \xrightarrow{\alpha} \Pi^j_k$ ssi $\exists\, p \in \Pi^j_i\, , \exists\, q \in \Pi^j_k$ tel que : $p \xrightarrow{\alpha} q$

Exemple : la réduction du STE p de la figure 10 donne le STE q de cette même figure.

Nous passons maintenant à une relation du type bisimulation mais qui permet de faire abstraction des transitions silencieuses τ.

2.4.5 bisimulationfaible

Une relation F ⊆ Q x Q est une bisimulation faible ssi F satisfait la propriété suivante :

p F q =>

$\forall u \in A_\varepsilon$ et $\forall p' \in Q$ tels que $p \overset{u}{\Longrightarrow} p'$ alors $\exists q'$ tel que $q \overset{u}{\Longrightarrow} q'$ et p' F q'

et symétriquement

$\forall u \in A_\varepsilon$ et $\forall q' \in Q$ tels que $q \overset{u}{\Longrightarrow} q'$ alors $\exists p'$ tel que $p \overset{u}{\Longrightarrow} p'$ et p' F q'

Cette propriété se représente graphiquement (Fig 11) :

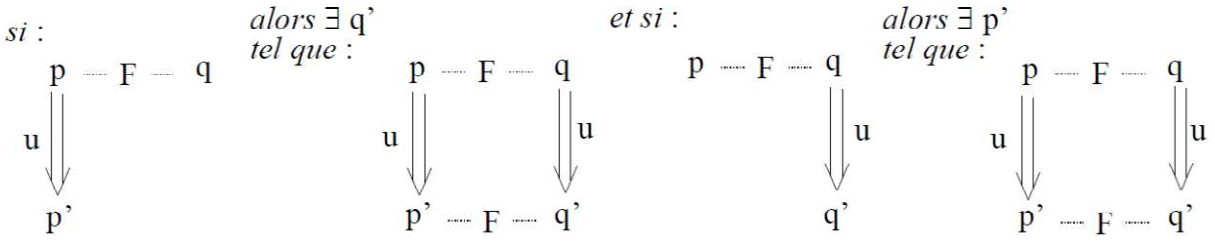

Fig. 11 : Propriété caractéristique des relations de bisimulation faible

Nous pouvons définir la relation de bisimulation faible (notée ≈):

$p \approx q$ ssi ∃ une relation de bisimulation faible F avec p F q

Il est possible de définir un algorithme de regroupement d'états faiblement bisisimilaires de façon identique au cas de la bisimulation forte, en remplaçant partout la flèche simple par la double flèche. De même, il est possible de réduire un STE en un STE minimal et faiblement bisimilaire en procédant de la façon suivante :

- chaque partie de la partition finale, Π^j_1, ..., Π^j_n, est considérée comme un état,
- pour a ∈ A, $\Pi^j_i \xrightarrow{a} \Pi^j_k$ ssi ∃ p ∈ Π^j_i , ∃ q ∈ Π^j_k tel que : $p \overset{a}{\Longrightarrow} q$
- l'on a $\Pi^j_i \xrightarrow{\tau} \Pi^j_k$ ssi ∃ p ∈ Π^j_i , ∃ q ∈ Π^j_k, q ≠ p, et tel que : $p \overset{\varepsilon}{\Longrightarrow} q$

2.4.6 propriétés des bisimulations fortes et faibles

Les deux bisimulations sont des relations d'équivalence et elles sont compositionnelles avec un bon nombre d'opérateurs sur les STE (et notament ceux de CCS+). Il est néanmoins important de noter que la bisimulation faible n'est pas compositionnelle avec l'opérateur de choix (+).

Exemple :

$(\tau\ ;\ a!;\ stop\ ) \approx (a!;\ stop)$ mais $\neg\ (\tau;\ a!;\ stop\ )\ (+)\ (b!;\ stop) \approx (a!;\ stop)(+)\ (b!;\ stop)$

Nous avons par ailleurs les implications suivantes entre les relations : $\sim\ \subset\ \approx\ \subset\ \sim_t$
Dans les exemples de la figure 5 nous avions $p_2 \sim_t q_2$ mais il est aisé de montrer que nous avons $\neg\ (p_2 \sim q_2)$ et $\neg\ (p_2 \approx q_2)$. Par ailleurs, la bisimulation faible ne préserve pas la propriété dite de divergence. Un STE est divergent s'il possède une séquence infinie d'actions internes. Deux STE peuvent être faiblement bisimilaires, l'un pouvant être divergent et l'autre pas (exemple en figure 12).

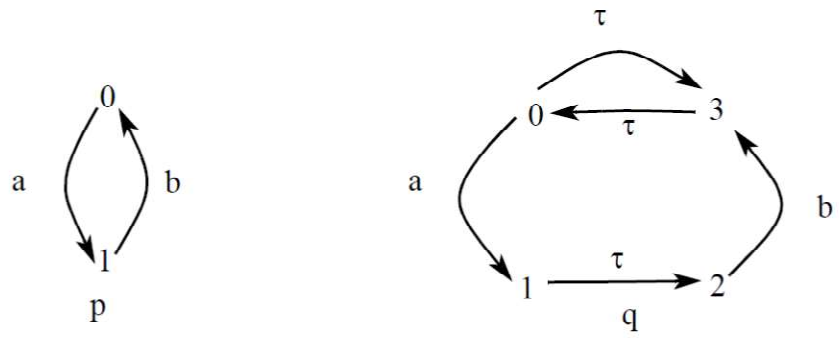

Fig. 12 : p et q sont faiblement bisimilaires avec q divergent et p non divergent

4. Références

R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.

Robin Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.

Rocco De Nicola, Matthew Hennessy: *Testing Equivalences for Processes*. Theor. Comput. Sci. 34: 83-133 (1984)

D. M. R. Park. *Concurrency and automata on infinite sequences.* In P. Deussen, editor: *5th GI Conference*, LNCS 104, Springer-Verlag, pp. 167-183, 1981.

Hubert Garavel, Frédéric Lang, Radu Mateescu. *An overview of CADP 2001.* European Association for Software Science and Technology (EASST) Newsletter volume 4, pages 13-24, August 2002. Also available as INRIA Technical Report RT-254.

httsp://www.inrialpes.fr/vasy/cadp/