Systèmes de transition asynchrone

Définition 1 *Un système de transition asynchrone est un quintuplet* $(S,i,E,I,Tran)$*,*

- $(S,i,E,Tran)$ *est un système de transition,*
- $I \subseteq E \times E$ *est une relation symétrique irréflexive (la relation d'"indépendance") telle que,*

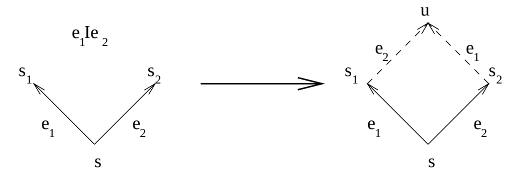
  (1) $e \in E \Rightarrow \exists s, s' \in S, (s,e,s') \in Tran$

  (2) $(s,e,s') \in Tran \wedge (s,e,s'') \in Tran \Rightarrow s' = s''$

  (3) $e_1 I e_2 \wedge (s,e_1,s_1) \in Tran \wedge (s,e_2,s_2) \in Tran \Rightarrow$
  $\exists u, (s_1,e_2,u) \in Tran \wedge (s_2,e_1,u) \in Tran$

  (4) $e_1 I e_2 \wedge (s,e_1,s_1) \in Tran \wedge (s_1,e_2,u) \in Tran \Rightarrow$
  $\exists s_2, (s,e_2,s_2) \in Tran \wedge (s_2,e_1,u) \in Tran$

Condition (3)



Modèles du Vrai Parallélisme

Eric Goubault
Commissariat à l'Energie Atomique, Saclay
email: Eric.Goubault@cea.fr
http://www.di.ens.fr/~goubault
http://www.enseignement.polytechnique.fr/profs/informatique/

Eric.Goubault

avec Pierre-Louis Curien (PPS)
James Leifer (INRIA Rocquencourt)
Jean-Jacques Lévy (INRIA Rocquencourt)
Catuscia Palamidessi (INRIA Futurs)

Exemple d'adjonction

Foncteur inclusion $ST \hookrightarrow TS$ adjoint à gauche de $Unfold : TS \to ST$: $Unfold(T = (S,i,L,Tran)) = (S',i',L,Tran')$ avec,

- S' est l'ensemble de toutes les suites de transitions finies $(t_1,\ldots,t_j,t_{j+1},\ldots,t_{n-1})$ t.q. $t_j = (s_{j-1},a_j,s_j)$ et $t_{j+1} = (s_j,a_{j+1},s_{j+1})$ $(1 < j < n)$
- $i' = ()$ la suite vide
- $Tran'$ consiste en tous les triplets (u,a,v) où $u, v \in S'$ et $u = (u_1,\ldots,u_k)$, $v = (u_1,\ldots,u_k,(s,a,s'))$ obtenu en ajoutant une transition a à u

(en fait, coréflexion, on oublie les boucles - application: préservation des produits)

Produit cartésien

Soient $T_0 = (S_0, i_0, E_0, I_0, Tran_0)$ et $T_1 = (S_1, i_1, E_1, I_1, Tran_1)$ deux systèmes de transitions asynchrones. Leur produit $T_0 \times T_1$ est $(S, i, E, I, Tran)$ où,

- le système de transitions sous-jacent $(S, i, E, Tran)$ est le produit cartésien de $(S_0, i_0, E_0, Tran_0)$ avec $(S_1, i_1, E_1, Tran_1)$
- aIb ssi ($p_0(a)$ et $p_0(b)$ sont définis, implique $p_0(a) I_0 p_0(b)$) et ($p_1(a)$ et $p_1(b)$ sont définis, implique $p_1(a) I_0 p_1(b)$)

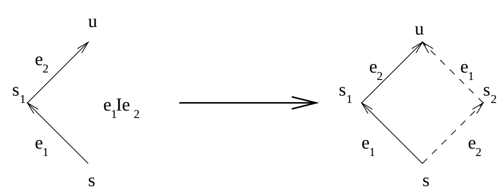
<u>Exercice</u>: montrer que cela est bien un produit cartésien

Coproduit

Soient $T_0 = (S_0, i_0, E_0, I_0, Tran_0)$ et $T_1 = (S_1, i_1, E_1, I_1, Tran_1)$ deux systèmes de transitions asynchrones. Leur coproduit $T_0 + T_1$ est $(S, i, E, I, Tran)$ où,

- le système de transitions sous-jacent $(S, i, E, Tran)$ est le coproduit de $(S_0, i_0, E_0, Tran_0)$ avec $(S_1, i_1, E_1, Tran_1)$
- aIb ssi (il existe a_0 et b_0 tels que a=$j_0(a_0)$ et $b = j_0(b_0)$ et $a_0 I_0 b_0$) ou (il existe a_1 et b_1 tels que a=$j_1(a_1)$ et $b = j_1(b_1)$ et $a_1 I_0 b_1$)

Condition (4)



Morphismes

Ce sont les morphismes de systèmes de transition f préservant la relation d'indépendance:

$$aIb \Rightarrow f(a) I' f(b)$$

soumis aux conditions suivantes,

(1) pour tout $t \in A^{\#}$ et $u \in A^{\#}$,

  (a) $dom(t \uparrow u) = cod(u)$,

  (b) $cod(t \uparrow u) = cod(u \uparrow t)$.

(2) pour tout $t : q \to r \in A^{\#}$,

  (a) $id_q \uparrow t = id_r$,

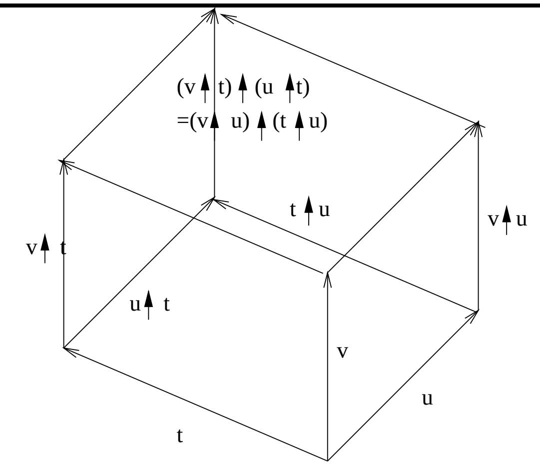
  (b) $t \uparrow id_q = t$,

  (c) $t \uparrow t = id_r$.

(3) pour toutes transitions coinitiales t, u, v dans $A^{\#}$,
$(v \uparrow t) \uparrow (u \uparrow t) = (v \uparrow u) \uparrow (t \uparrow u)$ (l'"axiome du cube")

(4) pour toutes transitions coinitiales t, u dans $A^{\#}$, si $t \uparrow u$ et $u \uparrow t$ sont toutes les deux des identités alors $t = u$.

L'axiome du cube



Système de transition concurrent

Définition 2 *Un système de transition concurrent (CTS) est une structure* $(G, \uparrow)$*,*

- $G = (O, A, dom, cod, id)$ *est un graphe "avec identités" c.a.d.,*
  - O *ensemble d'états,*
  - A *ensemble de transitions,*
  - $dom : A \to O$ *relie les transitions à leurs états initiaux,*
  - $cod : A \to O$ *relie les transitions à leurs états finaux,*
  - $id : O \to A$ *relie chaque* $s \in O$ *à une transition spéciale (les transitions "*" des systèmes de transition standards)* id_s *telle que* $dom(id_s) = s$ *et* $cod(id_s) = s$.

Suite

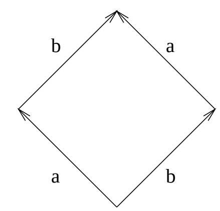
- $\uparrow$: $Coin(G^{\#}) \to A^{\#}$ est *l'opération résidu* qui vérifie,
  - $G^{\#}$ est le graphe augmenté $(O^{\#}, A^{\#}, dom, cod, id)$ avec,
    - $O^{\#} = O \cup \{\Omega\}$ (Ω n'appartient pas à O),
    - $A^{\#} = A \cup \{\omega_q / q \in O^{\#}\}$, $dom(\omega_q) = q$, $cod(\omega_q) = \Omega$.
  - $Coin(X)$ (où X est un graphe) est l'ensemble des transitions *coinitiales*, c.a.d. l'ensemble des couples (t, u) de transitions t, u de X qui ont les mêmes états de départ.

Automates de Trace

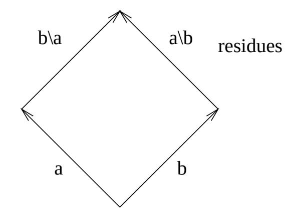
Tout cela est soumis aux conditions suivantes,

- $q \xrightarrow{\epsilon} r$ ssi $q = r$,
- si $q \xrightarrow{a} r$ et $q \xrightarrow{a} r'$ alors $r = r'$ (similaire à la condition (2) des systèmes de transition asynchrones),
- pour tous les états q et événements a, b, si $a \|_E b$, $q \xrightarrow{a} r$ et $q \xrightarrow{b} s$ alors pour un état p il existe des transitions $s \xrightarrow{a} p$ et $r \xrightarrow{b} p$ (similaire à la condition (3) des systèmes de transition asynchrones).

Equivalence par permutation et résidus



Permutation



Strong confluence

Idée similaire!

Morphismes

Définition 3 *Un morphisme de systèmes de transition concurrent est une paire de fonctions* $\rho = (\rho_O, \rho_A) : (O, A, dom, cod, id, \uparrow) \to (O', A', dom', cod', id', \uparrow')$ *telle que,*

- $\rho_O : O \to O'$ *et* $\rho_A : A \to A'$ *sont des fonctions avec* $dom' \circ \rho_A = \rho_O \circ dom$, $cod' \circ \rho_A = \rho_O \circ cod$ *et* $\rho_A \circ id = id' \circ \rho_O$
- *Si* t, u *sont des transitions propres chérentes de* C *alors* $\rho(t \uparrow u) = \rho(t) \uparrow' \rho(u)$.

Automates de trace

Un automate de trace est un triplet $A = (E, Q, T)$,

- E est un alphabet "parallèle", c'est à dire un ensemble d'événements sur lequel on a une relation binaire symétrique, irreflexive $\|_E$ appelée relation de parallélisme
- Q ensemble d'états,
- $T \subseteq Q \times (E \cup \{\epsilon\}) \times Q$ ensemble de transitions.

Equivalence de traces

Soit (M, L, I) un langage de trace. Pour $s, t \in M$ on définit $\cong$ comme la plus petite relation d'équivalence t.q.

$$sabt \cong sbat$$

si aIb, pour tout $sabt, sbat \in M$.

On appelle $\{s\}_{\cong}$ une trace de M.

Un exemple

- Un distributeur VM de café (c) ou de thé (t) pour un coût de c_2
- Un autre distributeur VM' de thé, pour un coût moindre c_1, mais qui peut tomber en panne (b)
- Un consommateur C qui veut prendre un café pour c_2 ou un thé pour c_1

Langages de traces

- (H, L) avec H un sous-ensemble non vide de chaines de L^*, clos par préfixes
- $(H, \epsilon, L, Tran) = ls(H, L)$ est un arbre de synchronisation ($(h, a, h') \in Tran$ ssi $h' = ha$)
- Inversement $sl(T = (S, i, L, Tran) = (H, L)$ (T est un arbre de synchronisation) est un langage de traces ($h \in L^*$ est dans H ssi il existe une suite de transitions dans T de la forme $i \rightarrow^{a_1} s_1 \rightarrow^{a_2} s_2 \ldots \rightarrow^{a_n} s_n$)

Reflexion (ou on oublie la structure de branchement - sl adjoint à gauche de ls)

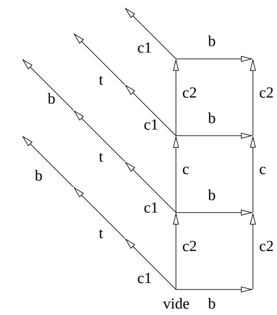
Traces de Mazurkiewicz non-généralisées

- (M, L, I) avec L un ensemble, $I \subseteq L \times L$ symétrique irréflexive (relation d'indépendance) et M un sous-ensemble non vide de L^* t.q.:
  - $sa \in M$ implique $s \in M$ pour tout $s \in L^*$, $a \in L$
  - $sabt \in M$ et aIb implique $sbat \in M$ pour tout $s, t \in L^*$, $a, b \in L$
  - $sa \in M$ et $sb \in M$ et aIb implique $sab \in M$ pour tout $s \in L^*$, $a, b \in L$

Langage de trace

- On a au moins les traces $\{\epsilon, c_1, c_2, b, c_1 t, c_2 c, c_2 b, b c_2, \ldots\}$
- On définit I comme étant la plus petite relation déquivalence telle que bIc et bIc_2

Ordre partiel sur les traces



(ne voit pas que le système peut avoir un point mort après que C fasse c_2! - il faudrait différencier les 2 types de c_2)

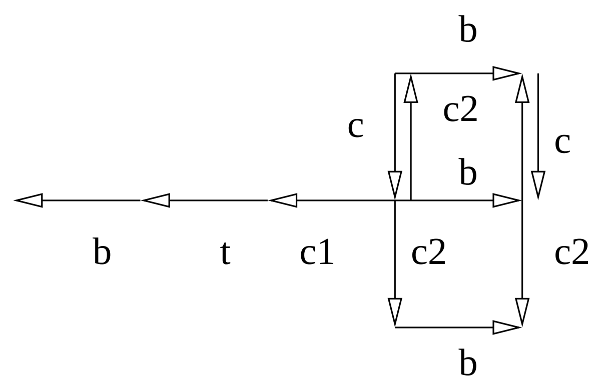
Un exemple

$$SYS = (VM \mid VM' \mid C) \backslash \{b, c, c_1, c_2, t\}$$

- $VM = c_2?.c!.VM + c_2?.t!.VM$
- $c1?.t!.VM' + b.nil$
- $c_2!.c?.C + c_1!.t?.nil$

Système de transition

En suppose la synchronisation possible sur $a?$ et $a!$ donnant l'action observable a:



Suite

tel que, si on définit $\cong$ comme la plus petite relation d'équivalence sur L^* telle que $sabu \cong sbau$ si aI_sb, et,

- pour tous $s \in M$, I_s est symétrique et irréflexive,
- (I est cohérente) $s \cong s'$ implique $I_s = I_{s'}$,
- (M est I-fermé) aI_sb implique $sab \in M$,
- (I est cohérente)
  (i) aI_sb et $aI_{sb}c$ et $cI_{sa}b$ implique $aI_{sc}b$,
  (ii) aI_sc et cI_sb implique (aI_sb ssi $aI_{sc}b$).

Morphismes

Définition 5 *Soient* (M, I, L) *et* (M', I', L') *deux langages de trace généralisés. Une fonction partielle* $\lambda : L \to L'$ *est un morphisme de* (M, I, L) *vers* (M', I', L') *ssi,*

- *λ preserve les mots: $s \in M$ implique $\lambda^*(s) \in M'$,*
- *λ respecte la relation d'indépendance:*
  aI_sb et $\lambda(a)$, $\lambda(b)$ definis implique $\lambda(a)I'_{\lambda^*(s)}\lambda(b)$ où λ^* est une extension de λ sur les mots définie comme suit,
  - $\lambda^*(\epsilon) = \epsilon$,
  - $\lambda^*(sa) = \begin{cases} \lambda^*(s)\lambda(a) & \textit{si } \lambda(a) \textit{ est defini} \\ \lambda^*(s) & \textit{sinon} \end{cases}$

Propriétés

- Soit (M, L, I) un langage de trace avec équivalence de trace $\cong$
- Si $su \in M$ et $s \cong s'$ alors $s'u \in M$ et $su \cong s'u$
- La relation $\leq$ ($s \leq t$ si $\exists u$, $su \cong t$) est un préordre
- Son quotient $\leq / \cong$ est un ordre partiel de traces

Les traces de Mazurkiewicz généralisées

Définition 4 *Un language de traces généralisé est un triplet* (M, I, L) *où,*

- *L ensemble de symboles,*
- *$M \subseteq L^*$,*
- *$I : M \to 2^{L \times L}$ est une fonction qui associe à chaque $s \in M$ une relation $I_s \subseteq L \times L$.*

Structures d'événements

Une structure d'événements première est une structure $(E, \leq, \#)$ où E est un ensemble d'*événements* partiellement ordonné par $\leq$ appelé *relation de dépendance causale* et où $\# \subseteq E \times E$ est une relation symétrique irreflexive, la *relation de conflit* satisfaisant,

- $\{e'/e' \leq e\}$ est fini (axiome des "causes finies"),
- $e\#e'$ and $e' \leq e''$ implique $e\#e''$ (le conflit est héréditaire).

On écrit e co e' ssi $\neg(e \leq e'$ ou $e' \leq e$ ou $e\#e')$.

Morphismes

Définition 6 *Soit $S = \{E, \leq, \#\}$ et $S' = \{E', \leq', \#'\}$ des structures d'événements premières. Un morphisme de structures d'événe–ments de S vers S' est une fonction partielle $f : E \to E'$ telle que,*

- *si $f(e)$ est définie alors $\{e'/e' \leq f(e)\} \subseteq f(\{e''/e'' \leq e\}$,*
- *si $f(e_0)$ et $f(e_1)$ sont tous les deux définis $f(e_0)\#f(e_1)$ ou $f(e_0) = f(e_1)$ implique $e_0\#e_1$ ou $e_0 = e_1$.*

NB: si f est définie, alors f préserve co.

Relations systèmes de transitions asynchrones/langages de traces

- On restreint d'abord la catégorie des systèmes de transitions asynchrones à TL_0 pour ne plus contenir que les langages de traces (M, E, I) tels que
$$\forall e \in E, \exists s, se \in M$$
- Soit (M, E, I) un langage de traces dans TL_0 avec équivalence $\sim$. On définit
$$tla(M, E, I) = (S, i, E, I, Tran)$$
  - $S = M/\sim$ avec $i = \{e\}_\sim$
  - $(t, e, t') \in Tran$ ssi $\exists s, se \in M$ such that $t = \{s\}_\sim \& t' = \{se\}_\sim$
- Soit $\eta : (M, E, I) \to (M', E', I')$ un morphisme de langages de traces. On définit $tla(\eta) = (\sigma, \eta)$ avec $\sigma(\{s\}_\sim = \{\hat{\eta}(s)\}_\sim$.

Relations systèmes de transitions asynchrones/langages de traces

- Soit $T = (S, i, E, I, Tran)$ un système de transitions asynchrones. On définit
$$atl(T) = (Seq, E, I)$$
  - Seq est l'ensemble de toutes les chaines d'évenements $e_1 e_2 \ldots e_n$ pour lesquelles il y a des transitions $(i_1, e1, s_1), (s_1, e_2, s_2), \ldots, (s_{n-1}, e_n, s_n) \in Tran$
- Soit $(\sigma, \eta) : T \to T'$ un morphisme de systèmes de transitions asynchrones. On définit $atl(\sigma, \eta) = \eta$.

tla et atl forment une adjonction...

Propriétés

- $e \leq e'$ ssi $\forall x \in \mathcal{D}^0(E, \leq, \#)$, $e' \in x$ implique $e \in x$
- $e \# e'$ ssi $\forall x \in \mathcal{D}^0(E, \leq, \#)$, $e \in x$ implique $e' \notin x$
- e co e' ssi $\exists x, x' \in \mathcal{D}^0(E, \leq, \#)$, $e \in x$ et $e' \notin x$ et $e \notin x'$ et $e' \notin x'$ et $x \cup x' \in \mathcal{D}^0(E, \#, \leq)$.

Relation avec les systèmes de transition

Soit $e \in E$ et $c \in \mathcal{D}(E, \leq, \#)$ alors e est permis en la configuration c, ce que l'on écrit par $c \vdash e$ si,

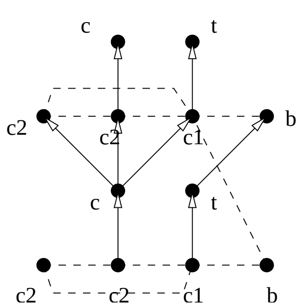
(i) $e \notin c$,

(ii) $\{e' / e' \leq e \wedge e' \neq e\} \subseteq c$,

(iii) $e' \in E$ and $e' \# e$ implique $e' \notin c$

Les configurations finies sont des traces quand on ordonne ses éléments avec la dépendance causale. $\{e_1 < e_2 < \ldots < e_n\}$ est une *garantie* pour c ssi $\{e_1, \ldots, e_{i-1}\} \vdash e_i$ pour $i = 1, \ldots, n$. On écrit aussi les garanties comme des chaines $e_1 \ldots e_n$.

Exemple



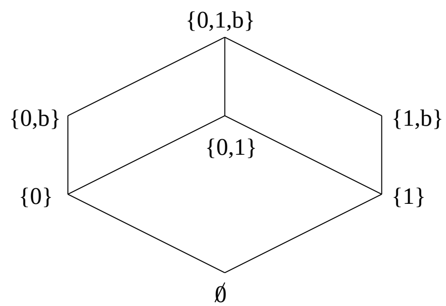
Ensemble de configurations

Soit $(E, \leq, \#)$ une structure d'événements. $\mathcal{D}(E, \leq, \#)$ est l'ensemble des configurations de E, sous-ensembles $x \subseteq E$ de E qui sont,

- sans conflit: pour tous $e, e' \in x$, e n'est pas en conflit avec e',
- fermés vers le bas: pour tous e, e', $e \in x$ et $e' \leq e$ implique $e' \in x$

On définit la configuration particulière $\lceil e \rceil = \{e' \in E \mid e' \leq e\}$. On définit $\mathcal{D}^0$ les configuration *finies*.

Exemple: les configurations correspondantes



Propriétés catégoriques

Référence: Voir G. Winskel et al. (Models of Concurrency, Handbook in Logic in Computer Science, Vol.3).

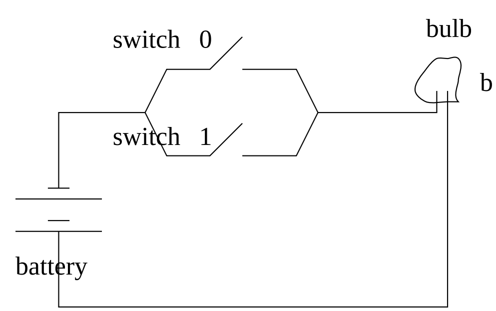
Avec ces définitions, il est possible d'avoir des algèbres de modèles avec des équivalents pour les produits synchronisés etc.

Relation avec les systèmes de transition

Soient $x, x' \in \mathcal{D}(E, \leq, \#)$. On écrit

$$x \rightarrow^e x' \Leftrightarrow e \notin x \& x' = x \cup \{e\}$$

Exemple: un interrupteur parallèle



Exercices

- Produit cartésien de deux structures d'événements? (difficile)
- Coproduit? (facile)
- La traduction d'un système d'événement vers un système de transitions est claire (configurations...), dans l'autre sens?