Sémantique et Analyse de Programmes Parallèles

Eric Goubault
CEA/LIST/DTSI/SLA Laboratoire Sûreté du Logiciel

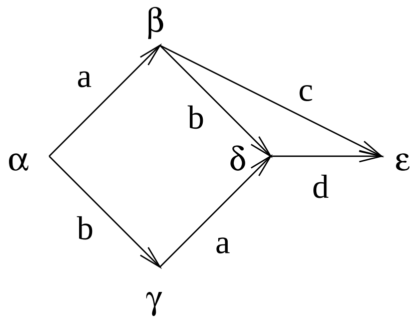
Rappels – Systèmes de transitions

Définition 1 *Un système de transitions est une structure*

$$(S, i, L, Tran)$$

- S *ensemble d'états et* i *état initial,*
- L *ensemble d'étiquettes, et*
- $Tran \subseteq S \times L \times S$ *la relation de transition*

Exemple



Catégorie

On en fait une catégorie avec pour morphismes les simulations (Glynn Winskel et al.).

Un système de transition T_1 simule un système de transitions T_0 à la condition que si T_0 peut effectuer une action a dans un certain contexte, alors T_1 peut effectuer a également dans un contexte similaire.

Un morphisme $f : T_0 \to T_1$ definit la façon dont les états et les transitions de T_0 sont reliés aux états et transitions de T_1.

Morphismes

Définition 2 *Soit* $T_0 = (S_0, i_0, L_0, Tran_0)$ *et* $T_1 = (S_1, i_1, L_1, Tran_1)$ *deux systèmes de transition. Un morphisme* $f : T_0 \to T_1$ *est un couple* $f = (\sigma, \lambda)$ *où,*

- $\sigma : S_0 \to S_1$,
- $\lambda : L_0 \to L_1$ *est tel que* $\sigma(i_0) = i_1$ *et*

$$(s, a, s') \in Tran_0 \Rightarrow (\sigma(s), \lambda(a), \sigma(s')) \in Tran_1$$

<u>Référence:</u> définition différente de celle de G. Winskel et al. (pas de morphismes “partiels”).

Morphismes partiels

Les morphismes partiels permettent à T_1 de ne pas effectuer d'action quand T_0 le fait.

Définition 3 *Soient* $T_0 = (S_0, i_0, L_0, Tran_0)$ *et* $T_1 = (S_1, i_1, L_1, Tran_1)$ *deux systèmes de transition. Un morphisme partiel* $f : T_0 \to T_1$ *est un couple* $f = (\sigma, \lambda)$ *où,*

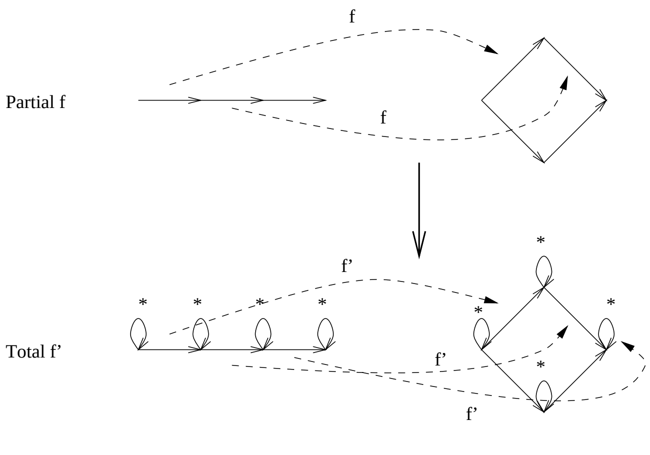
- $\sigma : S_0 \to S_1$,
- $\lambda : L_0 \to L_1$ *est une fonction* partielle. (σ, λ) *est tel que*
  - $\sigma(i_0) = i_1$,
  - $(s, a, s') \in Tran_0$ *et* $\lambda(a)$ *est défini implique* $(\sigma(s), \lambda(a), \sigma(s')) \in Tran_1$. *Sinon, si* $\lambda(a)$ *n'est pas défini alors* $\sigma(s) = \sigma(s')$.

Transitions silencieuses

On complète un système de transition $T = (S, i, L, Tran)$ en $T_* = (S_*, i_*, L_*, Tran_*)$,

- $S_* = S$,
- $i_* = i$,
- $L_* = L \cup \{*\}$,
- $Tran_* = Tran \cup \{(s, *, s)/s \in S\}$.

Exemple



Produits Cartésien

Soient deux systèmes de transitions $T_0 = (S_0, i_0, L_0, Tran_0)$ et $T_1 = (S_1, i_1, L_1, Tran_1)$. Leur produit est $T_0 \times T_1 = (S, i, L, Tran)$ avec,

- $S = S_0 \times S_1$ et $i = (i_0, i_1)$. Soient $\rho_0 : S_0 \times S_1 \rightarrow S_0$ et $\rho_1 : S_0 \times S_1 \rightarrow S_1$ les projections canoniques,
- $L = L_0 \times_* L_1 = \{(a, *)/a \in L_0\} \cup \{(*, b)/b \in L_1\} \cup \{(a, b)/a \in L_0, b \in L_1\}$ avec des projections canoniques π_0 et π_1,
- $(s, a, s') \in Tran_*$ ssi $(\rho_0(s), \pi_0(a), \rho_0(s')) \in Tran_{0*}$ et $(\rho_1(s), \pi_1(a), \rho_1(s')) \in Tran_{1*}$

Autres Constructions catégoriques

Le coproduit de deux systèmes de transitions $T_0 = (S_0, i_0, L_0, Tran_0)$ et $T_1 = (S_1, i_1, L_1, Tran_1)$ est $T_0 \oplus T_1 = (S, i, L, Tran)$ avec,

- $S = (S_0 \cup S_1)/\{i_0 = i_1\}$, et $i = i_0 = i_1$,
- $L = L_0 \cup L_1$,
- $Tran = Tran_1 \cup Tran_2$.

Produits Synchronisés

Un cas particulier: produit fibré suivant.

Référence plus générale: Arnold et M.Nivat.

Soient $T_0 = (S_0, i_0, L_0, Tran_0)$ et $T_1 = (S_1, i_1, L_1, Tran_1)$ deux systèmes de transition. Leur produit synchronisé (à la CSP) est $T = (S, i, L, Tran)$ avec,

Définition

- $S = S_0 \times S_1$, $i = (i_0, i_1)$,
- $L = L_0 \times L_1$,
- $((s, t), a, (u, v)) \in Tran$ ssi
  - $a \in L_0 \cap L_1$ et $(s, a, u) \in Tran_0$ et $(t, a, v) \in Tran_1$,
  - $a \in L_0 \backslash L_1$ et $(s, a, u) \in Tran_0$ et $v = t$,
  - $a \in L_1 \backslash L_0$ et $u = s$ et $(t, a, v) \in Tran_1$.

Sémantique de processus parallèles

- Automate pour chaque processus séquentiel,
- produit synchronisé correspondant aux différentes primitives de communications utilisées.

Les états globaux du système de transition résultant P sont donc des n-uplets (n=nombre de processus communicants) $s = (s_1, \dots, s_n)$ avec s_i état local du processus P_i.

Jusqu'à une exponentielle de plus par rapport au cas séquentiel!

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 $I_s \subseteq L \times L$.*

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,
  - $\lambda^*(\epsilon) = \epsilon$,
  - $\lambda^*(sa) = \begin{cases} \lambda^*(s)\lambda(a) & \text{si } \lambda(a) \text{ est defini} \\ \lambda^*(s) & \text{sinon} \end{cases}$

Systèmes de transition asynchrone

Définition 6 *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,*

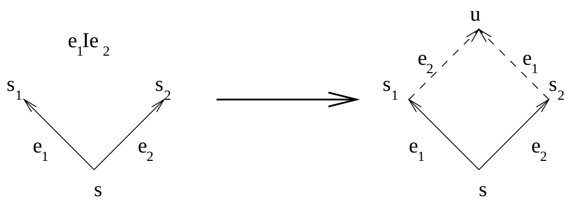
  (1) $e \in E \Rightarrow \exists s, s' \in S, (s, e, s') \in Tran$

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

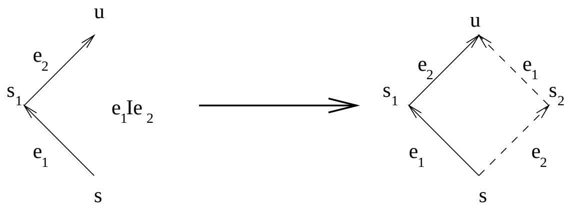
  (3) $e_1 I e_2 \wedge (s, e_1, s_1) \in Tran \wedge (s, e_2, s_2) \in Tran \Rightarrow \exists u, (s_1, e_2, u) \in Tran \wedge (s_2, e_1, u) \in Tran$

  (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)



Condition (4)



Morphismes

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

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

Automates de trace

Définition 7 *Un automate de trace est un triplet $A = (E, Q, T)$,*

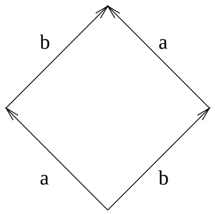
- *E est un alphabet "parallèle", c'est à dire un ensemble d'événements sur lequel on a une relation binaire symétrique, irreflexive $\|_E$ appelée relation de parallélisme*
- *Q ensemble d'états,*
- *$T \subseteq Q \times (E \cup \{\epsilon\}) \times Q$ ensemble de transitions.*

Automates de Trace

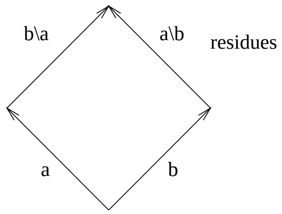
Tout cela est soumis aux conditions suivantes,

- $q \xrightarrow{\epsilon} r$ 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!

Système de transition concurrent

Définition 8 *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.

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) $id_q \uparrow t = id_r$,

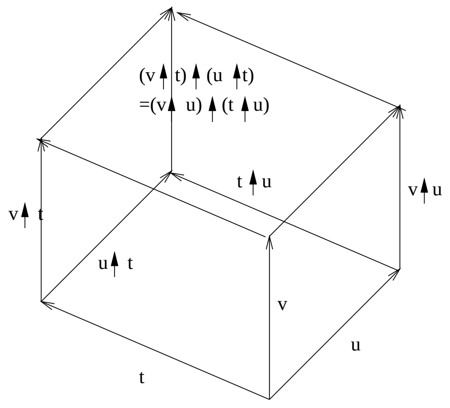
  (b) $t \uparrow id_q = t$,

  (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



Morphismes

Définition 9 *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* $\rho(t \uparrow u) = \rho(t) \uparrow' \rho(u)$.

Systèmes de transitions avec indépendance

Définition 10 *Un système de transition avec independance est une structure* $(S, s^I, L, Tran, I)$ *avec* $(S, s^I, L, Tran)$ *un système de transition et* $I \subseteq Tran^2$ *une relation irreflexive et symétrique telle que*

(i) $(s,a,s') \sim (s,a,s'') \Rightarrow s = s''$ *(condition (2) des systèmes de transition asynchrones),*

(ii) $(s,a,s')I(s,b,s'') \Rightarrow \exists u\,(s,a,s')I(s',b,u) \wedge (s,b,s'')I(s'',a,u)$ *(condition (3) des systèmes de transition asynchrones),*

(iii) $(s,a,s')I(s',b,u) \Rightarrow \exists s''(s,a,s')I(s,b,s'') \wedge (s,b,s'')I(s'',a,u)$ *(condition (4) des systèmes de transition asynchrones),*

(iv) $(s,a,s') \sim (s'',a,u)I(w,b,w') \Rightarrow (s,a,s')I(w,b,w')$.

Suite

où $\sim$ est la plus petite equivalence sur les transitions contenant la relation R définie par,

$$(s,a,s')R(s'',a,u) \Leftrightarrow \begin{cases} (s,a,s')I(s,b,s'') & et \\ (s,a,s')I(s',b,u) & et \\ (s,b,s'')I(s'',a,u) & \end{cases}$$

Morphismes

Ce sont des paires de fonctions (σ, λ) qui forment des morphismes de systèmes de transition plus,

- $(s,a,s') \in Tran \Rightarrow (\sigma(s), \lambda(a), \sigma(s')) \in Tran'$
- $(s,a,s')I(\overline{s}, b, \overline{s}') \Rightarrow (\sigma(s), \lambda(a), \sigma(s'))I(\sigma(\overline{s}), \lambda(b), \sigma(\overline{s}'))$

Réseaux de Pétri

Définition 11 *Un réseau de Pétri* $N = (P, T, pre, post)$ *est composé de,*

- P *ensemble de places,*
- T *ensemble de transitions,*
- $pre : T \rightarrow \tilde{P}$ *est la fonction de pré-condition, où* $\tilde{P}$ *est l'ensemble des multi-ensembles de* P*,*
- $post : T \rightarrow \tilde{P}$ *est la fonction de post-condition.*

On appelle marquage tout multi-ensemble de places

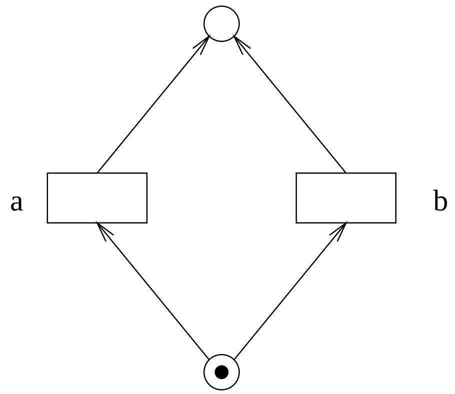
Relation de transition induite

La fonction de pré-condition décrit comment les transitions "consomment" les ressources et la fonction de post-condition montre comment les transitions "créent" des nouvelles resources.
Cela permet de définit une relation de transition entre les marquages.

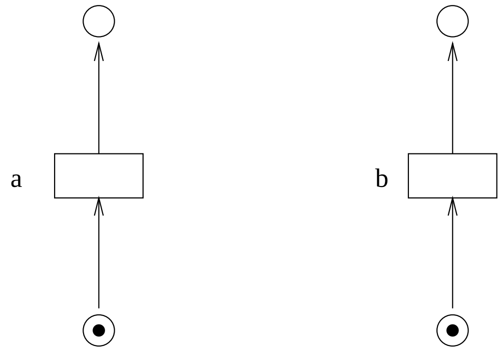
Si M et M' sont deux marquages d'un réseau N, et t est une transition de N on écrit $M[t\rangle M'$ pour "t fait une transition de M à M'" ssi

$$\exists M'' \in \tilde{M}, M = M'' + pre(t) \text{ et } post(t) + M'' = M'$$

Exemple: Exclusion Mutuelle entre a et b



Exemple: Exécution parallèle de a et de b



Structures d'événements

Définition 12 *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).*

Morphismes

Définition 13 *Soit $S = \{E, \leq, \#\}$ et $S' = \{E', \leq', \#'\}$ des structures d'événements premières. Un morphisme de structures d'événements 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 $e_0 = e_1$.*

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 $e' \in x$

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,

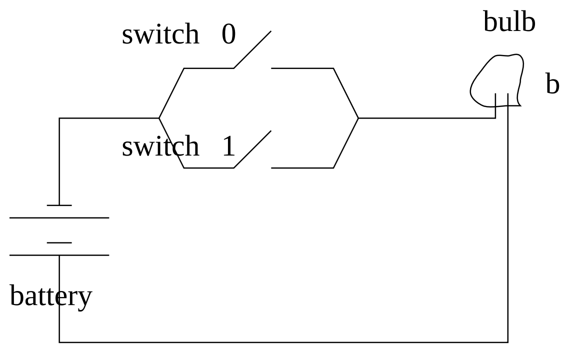
(i) $e \notin c$,

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

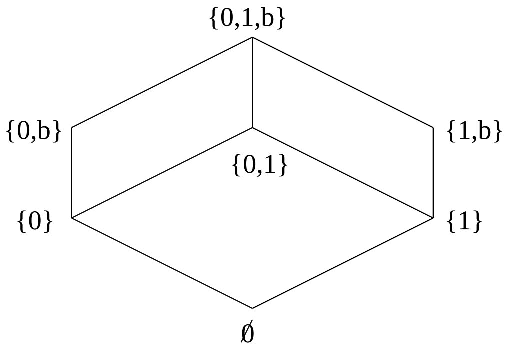
(iii) $e' \in E$ and $e'\#e$ implique $e' \notin c$

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 chaînes $e_1 \ldots e_n$.

Exemple: un interrupteur parallèle



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.

HDA – formalisation

(ensembles cubiques...)

- Modèle basé sur l'aglomération de points, segments, carrés, . . . ,hyperc
- les "collages" sont spécifiés par des opérateurs bords: d^0, l'opérateur bord "début" et d^1 l'opérateur bord "fin" (généralisation de ceux pour les automates standards).

Dimension un

On considère un segment,

$$0 \xrightarrow{I} 1$$

l'objet de dimension un I a comme bord début $d^0(I) = 0$, et comme bord fin $d^1(I) = 1$.

Dimension 2

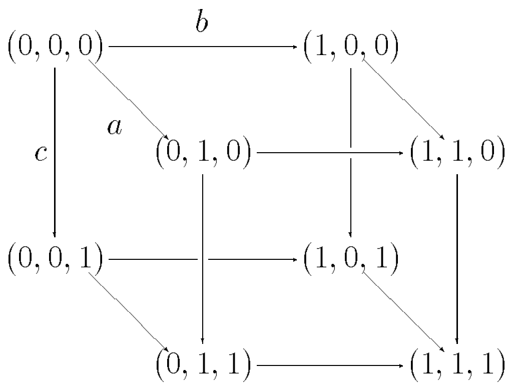
$$
\begin{tikzcd}
(0,0) \arrow[r, "a"] \arrow[d, "b"'] \arrow[dr, phantom, "A"] & (0,1) \arrow[d, "b'"'] \\
(1,0) \arrow[r, "a'"] & (1,1)
\end{tikzcd}
$$

L'objet de dimension 2 “interieur du carré” A a deux bords début à l'ordre près que l'on se fixe sur $\{a, b\}$, $d_0^0(A) = a$ et $d_1^0(A) = b$ comme de l'état $(0,0)$ on peut faire a et b. De même il a deux bords fin $d_0^1(A) = a'$ car $d_1^1(A) = b'$. On remarque que l'on a

$$d^0(d_1^0(A)) = (0,0) = d^0(d_0^0(A))$$
$$d^1(d_1^0(A)) = (1,0) = d^0(d_0^1(A))$$

Dimension 3

Cela se généralise aisément au cube.



Le cube

Soient A, B et C les faces (respectivement)

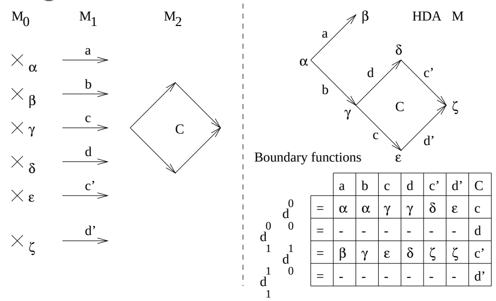
$$((0,0,0),(1,0,0),(0,0,1),(1,0,1))$$
$$((0,0,0),(0,1,0),(0,0,1),(0,1,1))$$
$$((0,0,0),(1,0,0),(0,1,0),(1,1,0))$$

Soient A', B' et C' les faces parallèles à A, B et C respectivement.

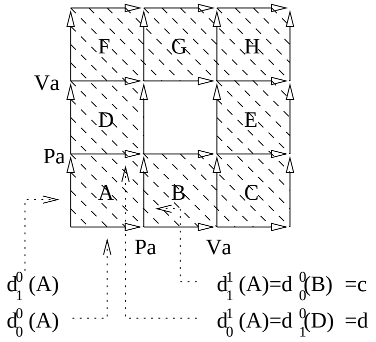
On a $d_0^0(D) = A$, $d_1^0(D) = B$, $d_2^0(D) = C$ et $d_0^1(D) = A'$, $d_1^1(D) = B'$, $d_2^1(D) = C'$. Alors $d_0^0(A) = b$, $d_1^0(A) = c$, $d_0^0(B) = a$, $d_1^0(A) = c$, $d_0^0(C) = a$, $d_1^0(C) = b$. On verifie que $d_i^0(d_j^0(D)) = d_{j-1}^0(d_i^0(D))$ pour tous les $i < j$.

Généralisation

Généralisation de ces relations aux hypercubes.
On amalgame avec des relations sur les bords:

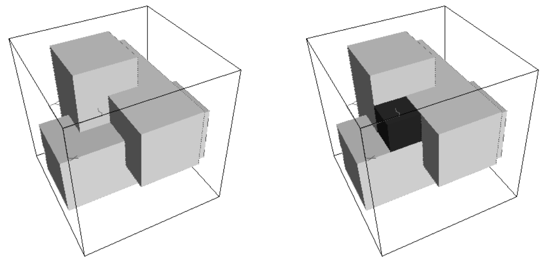


		a	b	c	d	c'	d'	C
d_0^0	=	α	α	γ	γ	δ	ε	c
d_1^0	=	-	-	-	-	-	-	d
d_0^1	=	β	γ	ε	δ	ζ	ζ	c'
d_1^1	=	-	-	-	-	-	-	d'



$$\begin{tikzcd}
F \arrow[r] & G \arrow[r] & H \\
D \arrow[u] & & E \arrow[u] \\
A \arrow[u, "(d)"] \arrow[r, "(c)"] & B \arrow[r] & C \arrow[u]
\end{tikzcd}$$

Autre exemple



Sémantique générale

Deux phases:
1): arbre syntaxique de chaque processus commence par une composition séquentielle de P et de V avec un autre terme.
Ces processus sont: $X_i = Q_i a_i.Y_i$, $1 \leq i \leq k$, où Q_i est P ou V, $a_i \in \mathcal{O}$ et Y_i est un processus. La sémantique de $X_1 \mid \cdots \mid X_k$ dans l'environnement ρ est $[\![X_1 \mid \cdots \mid X_k]\!]\rho$.

Première phase

Si pour tout $a \in \mathcal{O}$, $\rho(a) \geq 0$,

$$[\![X_1 \mid \cdots \mid X_k]\!]\rho = (X_1 \mid \cdots \mid X_k, \rho) + [\![Y_1 \mid X_2 \mid \cdots \mid X_k]\!]\rho_1 + \cdots + [\![X_1 \mid \cdots \mid X_{k-1} \mid Y_k]\!]\rho_k$$

où ρ_i, $1 \leq i \leq k$ tel que $\rho_i(b) = \rho(b)$ pour tout $b \in \mathcal{O}$, $b \neq a_i$, et $\rho_i(a_i) = \rho(a_i) - 1$ si $Q_i = P$ ou $\rho_i(a_i) = \rho(a_i) + 1$ si $Q_i = V$. Si il y a un $a \in \mathcal{O}$, $\rho(a) < 0$,

$$[\![X_1 \mid \cdots \mid X_k]\!]\rho = [\![Y_1 \mid X_2 \mid \cdots \mid X_k]\!]\rho_1 + \cdots + [\![X_1 \mid \cdots \mid X_{k-1} \mid Y_k]\!]\rho_k$$

avec les mêmes environnements ρ_i, $1 \leq i \leq k$.

Deuxième phase

(Elimination des variables de processus)

$$[\![X_1 \mid \cdots \mid Y.Y_i \mid \cdots \mid X_k]\!]\rho = [\![X_1 \mid \cdots \mid Proc_Y.Y_i \mid \cdots \mid X_k]\!]\rho$$

(Elimination de plus)

$$[\![X_1 \mid \cdots \mid Y_i + Z_i \mid \cdots \mid X_k]\!]\rho = [\![X_1 \mid \cdots \mid Y_i \mid \cdots \mid X_k]\!]\rho +_i [\![X_1 \mid \cdots \mid Z_i \mid \cdots \mid X_k]\!]\rho$$

Le plus de l'equation ici est une union en collant la face $(X_1 \mid \cdots \mid Y_i \mid \cdots X_k, \rho, i)$ avec la face $(X_1 \mid \cdots \mid Z_i \mid \cdots X_k, \rho, i)$.

Sémantique duale

Nouvelle sémantique pour PV:
Soit $T = X_1 \mid \cdots \mid X_n$ un terme pur tel que tous ses sous-termes sont purs. $X_i(j)$ est la $j$ème lettre de la chaîne X_i. On suppose que la longueur des chaînes X_i $(1 \leq i \leq n)$ sont les entiers l_i, on a

$$[k_1, r_1] \times \cdots \times [k_n, r_n] \in [\![X_1 \mid \cdots \mid X_n]\!]_2$$

s'il y a une partition de $\{1, \cdots, n\}$ en $U \cup V$ avec $card(U) = s(a) + 1$ pour un objet a avec, $X_i(k_i) = Pa$, $X_i(r_i) = Va$ pour $i \in U$ et $k_j = 0$, $r_j = l_j$ avec $j \in V$.

Et les invariants?

Il faut examiner les propriétés du modèle (produit synchronisé?).

On va lister les principales propriétés d'un de ces modèles

Définition

Définition 14 *Un HDA semi-régulier est une collection d'ensembles M_n $(n \in \mathbb{N})$ et de fonctions*

$$M_n \underset{d_j^1}{\overset{d_i^0}{\rightrightarrows}} M_{n-1}$$

pour tout $n \in \mathbb{N}$ et $0 \leq i, j \leq n-1$, telles que $d_i^k \circ d_j^l = d_{j-1}^l \circ d_i^k$ $(i < j, k, l = 0, 1)$ et $\forall n, m\, n \neq m,\ \ M_n \cap M_m = 0$.

Les éléments x de M_n $(\dim x = n)$ sont appelés des n-transitions (ou états si $n = 0$).

Morphismes

Définition 15 *Soient M et N deux HDA semi-réguliers, et f une famille $f_n : M_n \rightarrow N_n$ de fonctions. f est un morphisme de HDA semi-réguliers ssi*

$$f_n \circ d_i^0 = d_i^0 \circ f_{n+1}$$
$$f_n \circ d_i^1 = d_i^1 \circ f_{n+1}$$

pour tout $n \in \mathbb{N}$.

Cela définit la catégorie Υ_{sr} de HDA semi-réguliers.

Troncation

Foncteur troncation $T_n : \Upsilon_{sr} \rightarrow \Upsilon_{sr}^n$ defini par, $T_n(M)_m = M_m$ ssi $m \leq n$ et $T_n(M)_m = \emptyset$ si $m > n$.

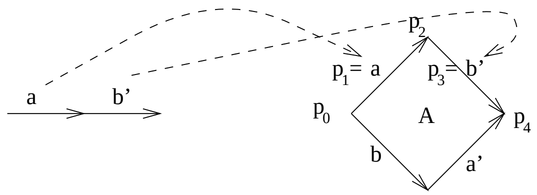
Son effet est de restreindre l'exécution à n processeurs.

Chemins

Définition 16 *Un chemin dans un HDA semi-régulier M est $p = (p_0, \dots, p_n)$ tel que p_0 et p_n sont des états et*

$$\forall k, 0 < k < n, \exists j, p_k = \begin{cases} d_j^1(p_{k-1}) & (i) \\ ou, & \\ p_k = d_j^0(p_{k+1}) & (ii) \end{cases}$$

Exemple



Propriétés catégoriques

Proposition 1 *Υ_{sr} est un* topos élémentaire. *Il est complet co-complet.*

Preuve. Soit $\square$ la catégorie libre dont les objets sont $[n]$, où $n \in \mathbb{N}$, et dont les morphismes sont engendrés par

$$[n] \underset{\delta_j^1}{\overset{\delta_i^0}{\rightrightarrows}} [n-1]$$

pour tout $n \in \mathbb{N}^*$ et $0 \leq i, j \leq n-1$, tels que

Preuve

$$\delta_i^k \delta_j^l = \delta_{j-1}^l \delta_i^k \quad (i < j)$$

Maintenant, la catégorie $\square$**Set** des foncteurs de $\square$ vers **Set** (les morphismes sont les transformations naturelles) est isomorphe à Υ_{sr}. C'est donc un topos élémentaire. Elle est complète et co-complète parce-que **Set** est complète et co-complète. $\square$

Description combinatoire

Soit $D_{[n]}$ le HDA semi-régulier $Hom_{\square}([n],.)$.

Définition 17 *Un n-cube singulier d'un HDA M est un morphisme $\sigma : D_{[n]} \to M$.*

Un lemme préparatoire

Lemma 1 *L'ensemble des n-cubes singuliers d'un HDA semi-régulier M est en correspondance bi-univoque avec M_n. L'unique n-cube singulier correspondant à un n-cube $x \in M_n$ est appelé $\sigma_x : D_{[n]} \to M$. C'est l'unique n-cube singulier σ tel que $\sigma(Id_{[n]}) = x$.*

Preuve. Par le lemme de Yoneda. Υ_{sr} isomorphe à la categorie de foncteurs de $\square$ dans **Set**. Donc $D_{[n]}$ sont les foncteurs représentables et $Nat(D_{[n]}, M) \cong M([n])$, où M est un foncteur de $\square$ vers **Set**. Cela se traduit en $\Upsilon_{sr}(D_{[n]}, M) \cong M_n$. $\square$

Une proposition préparatoire

Proposition 2 *Soit M un HDA semi-régulier. Le diagramme suivant est co-cartesien (pour $n \in \mathbb{N}$),*

$$\begin{tikzcd}
\coprod_{x \in M_{n+1}} \dot{D}_{[n+1]} \arrow[r, "\coprod_{x \in M_{n+1}} \dot{\sigma}_x"] \arrow[d, "\subseteq"'] & T_n(M) \arrow[d, "\subseteq"'] \\
\coprod_{x \in M_{n+1}} D_{[n+1]} \arrow[r, "\coprod_{x \in M_{n+1}} \sigma_x"] & T_{n+1}(M)
\end{tikzcd}$$

où $\dot{D}_{[n+1]} = T_n(D_{[n+1]})$ et $\dot{\sigma}_x = \sigma_{x|\dot{D}_{[n+1]}}$.

Preuve

Preuve. Il suffit de prover que le diagramme suivant (dans la categorie des ensembles) est co-cartesien pour tout $p \leq n+1$,

$$\begin{tikzcd}
\coprod_{x \in M_{n+1}} (\dot{D}_{[n+1]})_p \arrow[r, "\coprod_{x \in M_{n+1}} (\dot{\sigma}_x)_p"] \arrow[d, "\subseteq"'] & (T_n(M))_p \arrow[d, "\subseteq"'] \\
\coprod_{x \in M_{n+1}} (D_{[n+1]})_p \arrow[r, "\coprod_{x \in M_{n+1}} (\sigma_x)_p"] & (T_{n+1}(M))_p
\end{tikzcd}$$

comme les co-limites (donc les pushouts) sont calculées point par point dans une catégorie de foncteurs dans **Set**.

Preuve

Pour tout $p < n+1$, les inclusions sont en fait des bijections, et le diagramme est alors co-cartesien de manière évidente.

Pour $p = n+1$, le complément de $\coprod_{x \in M_{n+1}} (\dot{D}_{[n+1]})_p$ dans $\coprod_{x \in M_{n+1}} (D_{[n+1]})_p$ est l'ensemble des copies de cubes $Id_{[n+1]}$, une pour chaque cube de M_{n+1}. Cela signifie que la fonction $\coprod_{x \in M_{n+1}} (\sigma_x)_p$ induit une bijection du complément de $\coprod_{x \in M_{n+1}} (\dot{D}_{[n+1]})_p$ dans le complément de $(T_n(M))_p$. Cela implique que le diagramme est co-cartesien pour $p = n+1$ également. $\Box$

Signification

Aller du squelette de dimension n au squelette de dimension $n+1$ c'est:

- rajouter des relations d'indépendance,
- ou de façon duale, supprimer des exclusions mutuelles.

Cf. les systèmes de transition asynchrones etc.

Constructions catégoriques

Le produit cartesien de deux HDA semi-réguliers F, G est le HDA semi-régulier $F \times G$ avec

$$(F \times G)_n = F_n \times G_n$$
$$d_k^\epsilon[F \times G] = d_k^\epsilon[F] \times d_k^\epsilon[G]$$

C'est le produit complètement synchronisé des deux automates.

Coproduit

Le coproduit de deux HDA F, G est

$$(F \coprod G)_n = F_n \cup G_n$$
$$d_k^\epsilon[F \coprod G](x) = \begin{cases} d_i^\epsilon[F](x) & \text{si } x \in F_n \\ d_k^\epsilon[G](x) & \text{si } x \in G_n \end{cases}$$

C'est l'opérateur choix non-déterministe.

Foncteur Hom

L'adjoint à droite $\Rightarrow$ du produit cartésien est,

$$G \Rightarrow H([n]) = Nat(D_{[n]} \times G, H)$$

et pour $f \in G \Rightarrow H([n])$,

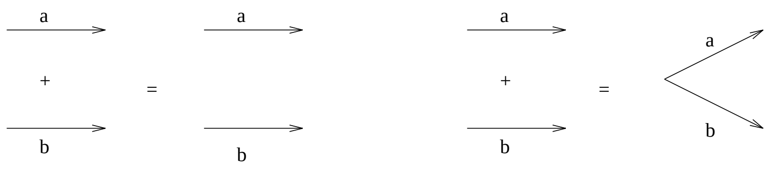
$$\begin{aligned} G \Rightarrow H(\delta_k^\epsilon)(f) : D_{[n-1]} \times G &\longrightarrow H \\ (u, v) &\longrightarrow f(u \circ \delta_k^\epsilon, v) \end{aligned}$$

(Nat est l'ensemble des transformations naturelles). Donc,

$$(G \Rightarrow H)_n = \{f : D_{[n]} \times G \to H / f \text{ morphisme}\}$$

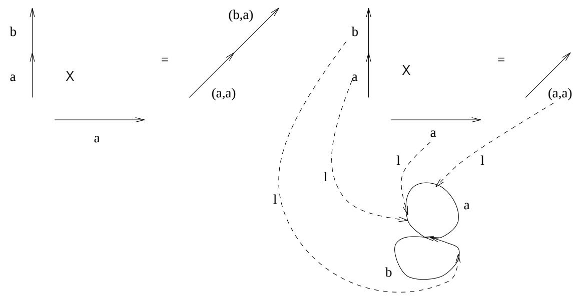
$$\begin{aligned} d_k^\epsilon[G \Rightarrow H](f) : D_{[n-1]} \times G &\longrightarrow H \\ (u, v) &\longrightarrow f(u \circ \delta_k^\epsilon, v) \end{aligned}$$

Exemple



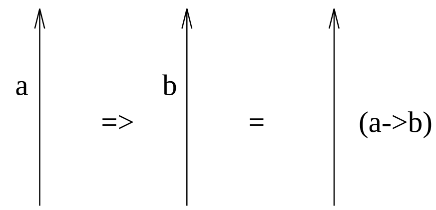
Coproduit et somme amalgamée

Exemple



Produit cartésien et produit fibré.

Exemple



Espace de fonction "synchrone"

Autres propriétés

Proposition 3 *Υ_{sr} est une catégorie monoidale fermée.*

Schéma de preuve. On construit $F \otimes G$ pour représenter la composition parallèle sans interférence,

$$(F \otimes G)_n = \cup_{i+k=n} F_i \times G_k$$

et pour $x \in F_i, y \in G_k$,

$$d_k^\epsilon[F \otimes G](x,y) = (d_k^\epsilon[F](x), y) \ \text{si} k \leq i-1$$
$$d_k^\epsilon[F \otimes G](x,y) = (x, d_{k-i}^\epsilon[G](y)) \ \text{si} k > i-1$$

Exemple

$$\begin{tikzcd} \beta \\ \alpha \arrow[u, "a"] \end{tikzcd} \otimes \begin{tikzcd} \delta \\ \gamma \arrow[u, "b"] \end{tikzcd} = \begin{tikzcd} & \beta\otimes\delta & \\ \beta\otimes\gamma \arrow[ur, "\beta\otimes b"] \arrow[rr, phantom, "a\otimes b"] & & \alpha\otimes\gamma \arrow[ul, "a\otimes\delta"'] \\ & \alpha\otimes\gamma \arrow[ul, "a\otimes\gamma"] \arrow[ur, "\alpha\otimes b"'] & \end{tikzcd}$$

Flèche linéaire

Le produit tensoriel a un adjoint à droite car il commute avec les colimites qui est encore donné par le lemme de Yoneda. C'est

$$(G \multimap H)_n = \{f : D_{[n]} \otimes G \to H / f \text{ est un morphisme}\}$$

et pour $f \in (G \multimap H)_n, d_k^\epsilon[G \multimap H](f) : D_{[n-1]} \otimes G \longrightarrow H$

$$(u,v) \longrightarrow f(u \circ \delta_k^\epsilon, v)$$

Dans $G \multimap H$ on a les fonctions qui créent d'autres processus (comme `pvm_spawn`).

Exemple

Dans

$$\begin{tikzcd} \beta \\ \alpha \arrow[u, "a"] \end{tikzcd} \multimap \begin{tikzcd} \beta \\ \alpha \arrow[u, "a"] \end{tikzcd} \otimes \begin{tikzcd} \delta \\ \gamma \arrow[u, "b"] \end{tikzcd}$$

Il y a par exemple une 1-transition "spawn l'action b"

$$\lambda x.x \otimes \gamma \xrightarrow{\lambda x.x \otimes b} \lambda x.x \otimes \delta$$