

Modèles du Vrai Parallélisme

Eric Goubault

Commissariat à l'Énergie 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)

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?

Quelques définitions importantes

- Si (M, L, I) est un langage de traces, on définit $s \preceq t$ ssi il existe u tel que $su \cong t$
- On définit alors \sim comme étant la plus petite relation d'équivalence sur les chaînes non vides de M telles que $sa \sim sba$ si bIa et $sa \sim ta$ si $s \cong t$.
- Soit $s \in M$. Les *événements* de s sont donnés par:

$$ev(s) = \{\{u\}_\sim \mid u \text{ est un préfixe non-vide de } s\}$$

Propriété

- L'ordre partiel quotient de \preceq que l'on avait construit sur les traces est exactement défini par $ev(s) \subseteq ev(t)$
- Pour tous les événements e , il existe $sa \in e$ tel que

$$\forall ta \in e, sa \preceq ta$$

- Cela signifie que chaque événement a un représentant minimal vis-à-vis de \preceq .

Adjonction langage de trace/structure d'événements

Soit $T = (M, L, I)$ un langage de traces. On définit $tle(M, L, I) = (E, \leq, \#)$ où:

- E est l'ensemble d'événements de T
- \leq est définie par $e \leq e'$ ssi $e \in ev(s)$ où sa est un représentant minimal de e'
- $e \# e'$ ssi il existe e_0, e'_0 avec $e_0 \#_0 e'_0$ et $e_0 \leq e$ et $e'_0 \leq e'$ où, par définition, $e_0 \#_0 e'_0$ ssi il existe v, a, b tels que $va \in e_0$ et $vb \in e'_0$ et $\neg(a(I \cup Id_L)b)$.

On définit également $\lambda_T : E \rightarrow L$ par $\lambda_T(\{sa\}_\sim) = a$

Propriété

$tle(M, L, I) = (E, \leq, \#)$ tel que:

- $e \leq e'$ ssi $\forall s \in M, e' \in ev(s)$ implique $e \in ev(s)$
- $e \# e'$ ssi $\forall s \in M, e \in ev(s)$ implique $e \notin ev(s)$

Adjonction langage de trace/structure d'événements

Soit ES une structure d'événements dont l'ensemble d'événements est E . On définit $etl(ES) = (M, E, co)$ où $s = e_1 \dots e_n \in M$ ssi il existe une chaîne

$$\emptyset \xrightarrow{e_1} x_1 \xrightarrow{e_2} x_2 \dots \xrightarrow{e_n} x_n$$

de configurations de ES

Définit une coréflexion de la catégorie des structures d'événements vers la catégorie des langages de trace.

On verra qu'il y a une réflexion entre la catégorie des structures d'événements *étiquetés* (généralisant les arbres de synchronisation) et la catégorie des langages de trace.

Labelled Event Structures

- objets: $(ES, l : (E, co) \rightarrow (L, I))$ où ES est une structure d'événements avec ensemble d'événements E , relation de concurrence co et la fonction d'étiquetage l est une fonction totale qui envoie des événements concurrents vers des étiquettes indépendantes (I est la relation d'indépendance)
- morphismes:
 $(\eta, \lambda) : (ES, l : (E, co) \rightarrow (L, I)) \rightarrow (ES', l' : (E', co') \rightarrow (L', I'))$
où,
 - $\eta : ES \rightarrow ES'$ est un morphisme de structures d'événements
 - $\lambda : (L, I) \rightarrow (L', I')$ est un morphisme d'ensembles avec relation d'indépendance (i.e. respectant ces relations)
 - tels que $l' \circ \eta = \lambda \circ l$

(réflexion avec les traces de Mazurkiewicz)

Sémantique vraiment parallèle de CCS

(réf. Winskel/Boudol-Castellani)

- nil est représenté par la structure d'évènement vide
- Si $S = (E, \leq, \#, \lambda)$ alors $a.S = (\{\epsilon\} \cup E, \leq', \#, \lambda')$ avec,
 - $\epsilon \notin E$
 - $e \leq' e'$ ssi $(e \leq e')$ ou $(e = \epsilon \& \emptyset \vdash e')$
 - $\lambda'(\epsilon) = a$ et $\lambda'(e) = \lambda(e)$ pour tout $e \in E$

Sémantique vraiment parallèle de CCS

- Si $S_i = (E_i, \leq_i, \#_i, \lambda_i)$ pour $i = 0, 1$ alors $S_0 \mid S_1 = (E, \leq, \#, \lambda)$

avec

$$- E = \{(i, e_i) \mid e_i \in E_i\} \cup \{(e_0, e_1) \mid e_i \in E_i \& \lambda(e_0) = \overline{\lambda(e_1)}\}$$

$$- e \leq e' \text{ ssi } \begin{cases} e = (i, e_i) \text{ ou } e = (e_0, e_1) & \text{et} \\ e' = (i, e'_i) \text{ ou } e' = (e'_0, e'_1) & \text{et} \\ e_i \leq_i e'_i & \text{pour tout } i \end{cases}$$

$$- e \# e' \text{ ssi } \begin{cases} e = (i, e_i) \text{ ou } e = (e_0, e_1) & \text{et} \\ e' = (i, e'_i) \text{ ou } e' = (e'_0, e'_1) & \text{et} \\ e_i \#_i e'_i & \text{pour tout } i \end{cases}$$

$$- \lambda(i, e) = \lambda_i(e) \text{ et } \lambda(e_0, e_1) = \tau$$

(produit!)

Sémantique vraiment parallèle de CCS

- Si $S_i = (E_i, \leq_i, \#_i, \lambda_i)$ pour $i = 0, 1$ alors $S_0 + S_1 = (E, \leq, \#, \lambda)$ avec
 - $E = \{(i, e_i) \mid e_i \in E_i\}$
 - $e \leq e'$ ssi $e = (i, e_i) \& e' = (i, e'_i) \& e_i \leq_i e'_i$
 - $e \# e'$ ssi $\begin{cases} e = (i, e_i) \& e' = (i, e'_i) \& e_i \#_i e'_i \\ e = (i, e_i) \& e' = (j, e'_j) \& i \neq j \end{cases}$ ou
 - $\lambda(i, e) = \lambda_i(e)$

(coproduit!)

Sémantique vraiment parallèle de CCS

- Si $S_i = (E, \leq, \#, \lambda)$ alors $S \setminus \alpha = S \upharpoonright H$ avec
 $H = E - \bigcup \{E_n \mid n \geq 0\}$ et
 - $E_0 = \{e \mid nm(\lambda(e)) = \alpha\}$
 - $E_{n+1} = E_n \cup \{e \mid G \vdash e \Rightarrow G \cap E_n \neq \emptyset\}$

Exemple

Considérons $c.\alpha \mid \bar{\alpha}b$:

avec $\alpha\#\tau$ et $\tau\#\bar{\alpha}$.

Exemple

Considérons $\alpha.c.\alpha \mid \bar{\alpha}$:
avec $\alpha \# \tau$, $\tau \# \tau$ et $\tau \# \bar{\alpha}$.

Propriétés

En utilisant l'adjonction avec les systèmes de transition, on peut montrer que la sémantique est correcte vis-à-vis de la sémantique par entrelacements.

Adjonction langage de trace/structure d'événements étiquetés

Soit (M, L, I) un langage de traces. On définit

$\mathcal{E}(M, L, I) = (E, \leq, \#, \lambda_T)$ où

- $(E, \leq, \#) = tle(T)$
- $\lambda_T : (E, co) \rightarrow (L, I)$ est la co-unité en T de la coréflexion entre les structures d'événements (non-étiquetées) et les langages de trace.

Pomsets

Les pomsets sont des ordres partiels d'événements étiquetés. On définit Pom_L comme étant la sous-catégorie de E_L (catégorie des structures d'événements étiquetés par L):

- objets: pomsets finis (donc pas de conflit)
- morphismes: tous les morphismes de structures d'événements étiquetés possibles

Bisimulations préservant l'historique

Soient E_1 et E_2 deux structures d'événements. Une bisimulation préservant l'historique entre E_1 et E_2 est un ensemble H de triplets (x_1, f, x_2) où x_1 est une configuration de E_1 , x_2 est une configuration de E_2 , f est un isomorphisme de pomsets entre E_1 et E_2 , tel que $(\emptyset, \emptyset, \emptyset) \in H$ et dès que (x_1, f, x_2) :

- (i) Si $x_1 \rightarrow^a x'_1 \in E_1$ alors $x_2 \rightarrow^a x'_2 \in E_2$ et $(x'_1, f', x'_2) \in H$ avec $f \subseteq f'$, pour un f' et un x'_2
- (ii) Si $x_2 \rightarrow^a x'_2 \in E_2$ alors $x_1 \rightarrow^a x'_1 \in E_1$ et $(x'_1, f', x'_2) \in H$ avec $f \subseteq f'$, pour un f' et un x'_1

Bisimulations fortes préservant l'historique

H est forte si en plus, elle satisfait:

- (I) $(x, f, y) \in H \& x' \subseteq x$, pour une configuration $x' \in E_1$, implique $(x', f', y') \in H$ pour un $f' \subseteq f$ et $y' \subseteq y$
- (II) $(x, f, y) \in H \& y' \subseteq y$, pour une configuration $y' \in E_2$, implique $(x', f', y') \in H$ pour un $f' \subseteq f$ et $x' \subseteq x$

Caractérisation

Deux structures d'événements sur L sont Pom_L -bisimilaires ssi elles sont fortement bisimilaires par une bisimulation préservant l'historique.

Bisimulation pour les systèmes de transition asynchrones

- On identifie Pom_L avec son image dans la catégorie ATS
- Les morphismes Pom_L -ouverts dans ATS sont les morphismes (σ, η) zig-zags:
 - η est totale et préserve les étiquettes
 - $(\sigma(s), e', u') \in Tran'$ implique $\exists (s, e, u) \in Tran$ tel que $\eta(e) = e'$ et $\sigma(u) = u'$
 - (s, e, u) et (u, e', v) dans $Tran$ (s atteignable) et $\eta(e)I'\eta(e')$ impliquent eIe'

Preuve

- Première partie similaire au cas des systèmes de transitions simples
- Deuxième partie: considérer l'inclusion du pomset:

$$\cdot \xrightarrow{a_1} \cdot \xrightarrow{a_2} \dots \xrightarrow{a_n} \cdot \xrightarrow{a} \cdot \xrightarrow{a'}$$

dans

$$\cdot \xrightarrow{a_1} \cdot \xrightarrow{a_2} \dots \xrightarrow{a_n} \cdot \xrightarrow{a} \dots \xrightarrow{a'}$$

Rapports entre les modèles

Cube de Winskel

Rapports entre les modèles

Soit M une sous-catégorie coréfléctive de N avec R adjoint à droite de l'inclusion $M \hookrightarrow N$ et P sous-catégorie de M . Alors:

- M_1 et M_2 sont P -bisimilaires dans M ssi M_1 et M_2 sont P -bisimilaires dans N
- N_1 et N_2 sont P -bisimilaires dans N ssi $R(N_1)$ et $R(N_2)$ sont P -bisimilaires dans M

Systemes de transitions avec independance

Définition 1 *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,

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

Morphismes

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

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

Morphismes Pom_L -ouverts

Les morphismes Pom_L -ouverts de TI_L sont les morphismes zig-zags tels que:

- si $(s, a, s'), (s', b, s'') \in Tran_1$, avec s atteignable, et
- si $(\sigma(s), a, \sigma(s'))I_2(\sigma(s'), b, \sigma(s''))$ dans T_2
- alors $(s, a, s')I_1(s', b, s'')$ dans T_1

Bisimulation préservant l'historique

Deux systèmes de transitions avec indépendance, avec ensemble d'étiquetage L , sont Pom_L -bisimilaires si leur déroulement en structures d'événements sont fortement bisimilaires en préservant les historiques.

Réseaux de Pétri

Définition 2 *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

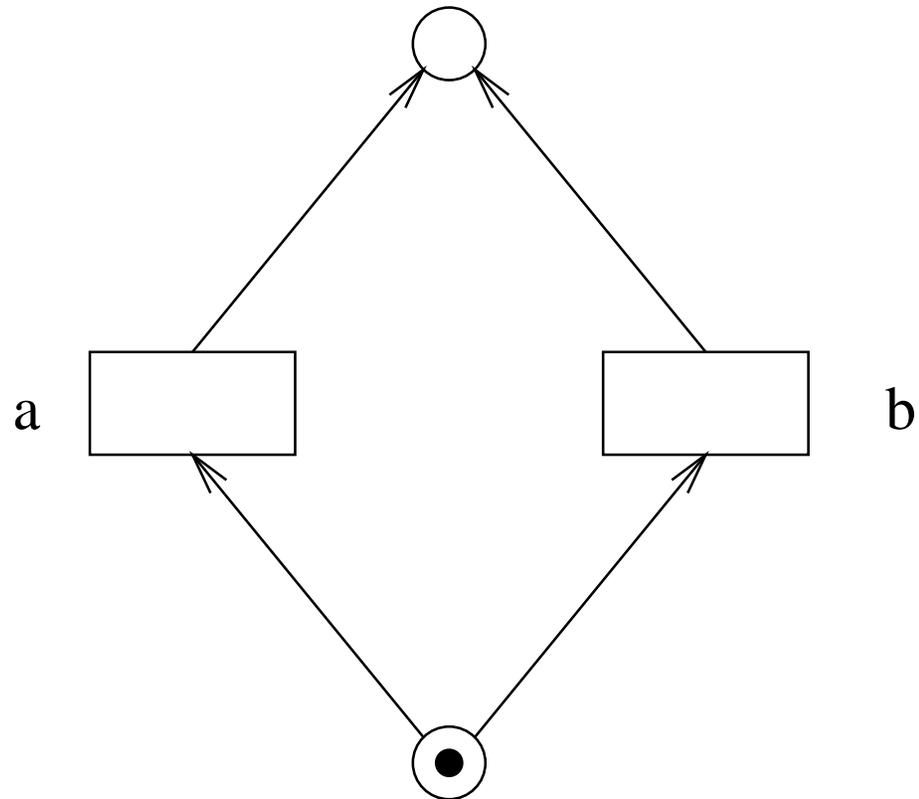
La fonction de pré-condition décrit comment les transitions “consomment” les ressources et la fonction de post-condition montre comment les transitions “créent” des nouvelles ressources.

Cela permet de définir 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

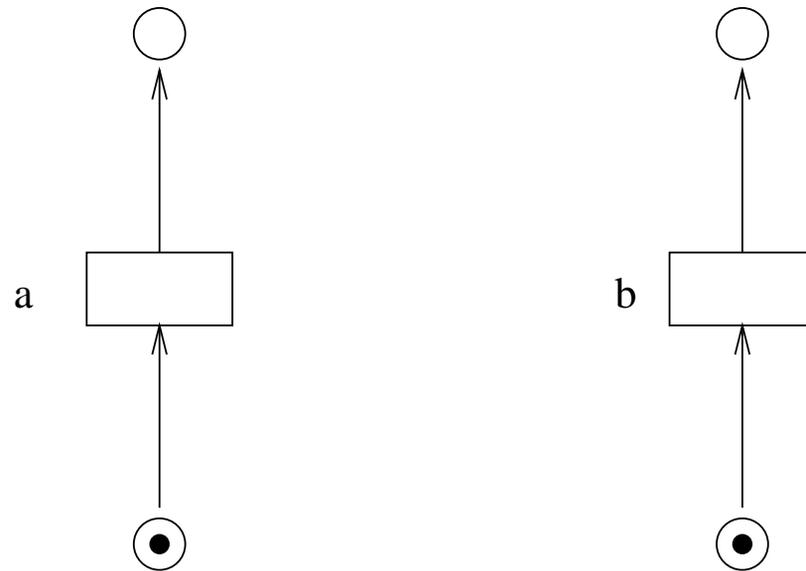
$$\exists M'' \in \tilde{M}, M = M'' + pre(t) \text{ et } post(t) + M'' = M'$$

Exemple: Exclusion Mutuelle entre a et b



Exercice: n -sémaphore?

Exemple: Exécution parallèle de a et de b



Coproduit

Soient $N_0 = (B_0, M_0, E_0, pre_0, post_0)$ et $N_1 = (B_1, M_1, E_1, pre_1, post_1)$ deux réseaux de Petri. On définit $(B, M, E, pre, post) = N_0 + N_1$ avec:

- $B = M_0 \times M_1 \cup (B_0 \setminus M_0) \times_* (B_1 \setminus M_1)$
- relations $j_0 \subseteq B_0 \times B, j_1 \subseteq B_1 \times B$ avec
 - $b_0 j_0 c \Leftrightarrow \exists b_1 \in B_1 \cup \{*\}, c = (b_0, b_1)$
 - $b_0 j_1 c \Leftrightarrow \exists b_0 \in B_0 \cup \{*\}, c = (b_0, b_1)$
- $M = M_0 \times M_1$

Coproduit

- $E = E_0 \cup E_1$ avec injections in_0 et in_1
- $pre(e) = j_0 \circ pre_0(e_0), post(e) = j_0 \circ post_0(e_0)$ si $e = in_0(e_0)$
- $pre(e) = j_1 \circ pre_1(e_1), post(e) = j_1 \circ post_1(e_1)$ si $e = in_1(e_1)$

Product

Soient $N_0 = (B_0, M_0, E_0, pre_0, post_0)$ et $N_1 = (B_1, M_1, E_1, pre_1, post_1)$ deux réseaux de Petri. On définit $(B, M, E, pre, post) = N_0 + N_1$ avec:

- $E = E_0 \times_* E_1$ (avec projections π_0 et π_1)
- $B = B_0 \cup B_1$ (avec injections ρ_0, ρ_1)
- $M = \rho_0^{op} M_0 + \rho_1^{op} M_1$ marquage initial du produit
- $pre(e) = \rho_0^{op}[pre_0(\pi_0(e))] + \rho_1^{op}[pre_1(\pi_1(e))]$
- $post(e) = \rho_0^{op}[post_0(\pi_0(e))] + \rho_1^{op}[post_1(\pi_1(e))]$

Systemes de transitions asynchrones et reseaux de Petri

- Soit $N = (B, M_0, E, pre, post)$ un reseau de Petri. On definit

$$na(N) = (S, i, E, I, Tran)$$

- $S = Pow(B)$ avec $i = M_0$
 - $e_1 I e_2$ ssi $prepost(e_1) \cap prepost(e_2) = \emptyset$
 - $(M, e, M') \in Tran$ ssi $M \xrightarrow{e} M' \in N$ pour $M, M' \in Pow(B)$
- Soit $(\beta, \eta) : N \rightarrow N'$ un morphisme de reseaux de Petri. On definit $na(\beta, \eta) = (\sigma, \eta)$ ou $\sigma(M) = \beta M$ pour tout $M \in Pow(B)$.

Dans l'autre sens...

Quelques définitions...

- Soit b une condition du réseau de Petri N . Soit $Tran$ la relation de transition de $na(N)$. On définit l'extension de b par:

$$|b| = \{(M, e, M') \in Tran_* \mid b \in M \& b \in M' \& b \notin prepost(e)\}$$

- Soit $T = (S, i, E, I, Tran)$ un système de transitions asynchrones. Ses conditions sont les sous-ensembles non-vides $b \subseteq Tran_*$ tels que:
 - $(s, e, s') \in b$ implique $(s, *, s) \in b$ et $(s', *, s') \in b$
 - * $(s, e, s') \in pre(b)$ et $(u, e, u') \in Tran$ impliquent $(u, e, u') \in pre(b)$
 - * $(s, e, s') \in post(b)$ et $(u, e, u') \in Tran$ impliquent $(u, e, u') \in post(b)$

Dans l'autre sens...

- où, pour $(s, e, s') \in Tran$, on définit:
 - $(s, e, s') \in pre(b)$ ssi $(s, e, s') \notin b$ et $(s', *, s') \in b$
 - $(s, e, s') \in post(b)$ ssi $(s, *, s) \in b$ et $(s, e, s') \notin b$
 - $prepost(b) = pre(b) \cup post(b)$
- $(s, e_1, s') \in prepost(b)$ et $(u, e_2, u') \in prepost(b)$ impliquent $\neg e_1 I e_2$

Vérification

On vérifie aisément que l'extension d'une condition d'un réseau de Pétri est une condition de son système de transition asynchrone associé.

Dans l'autre sens...

- Soit B l'ensemble des conditions de T . Pour $e \in E_*$, on définit
 - $post(e) = \{b \in B \mid \exists s, s', (s, e, s') \in pre(b)\}$
 - $pre(e) = \{b \in B \mid \exists s, s', (s, e, s') \in post(b)\}$
 - $prepost(e) = pre(e) \cup post(e)$
- Pour $s \in S$, on définit $M(s) = \{b \in B \mid (s, *, s) \in b\}$

Dans l'autre sens...

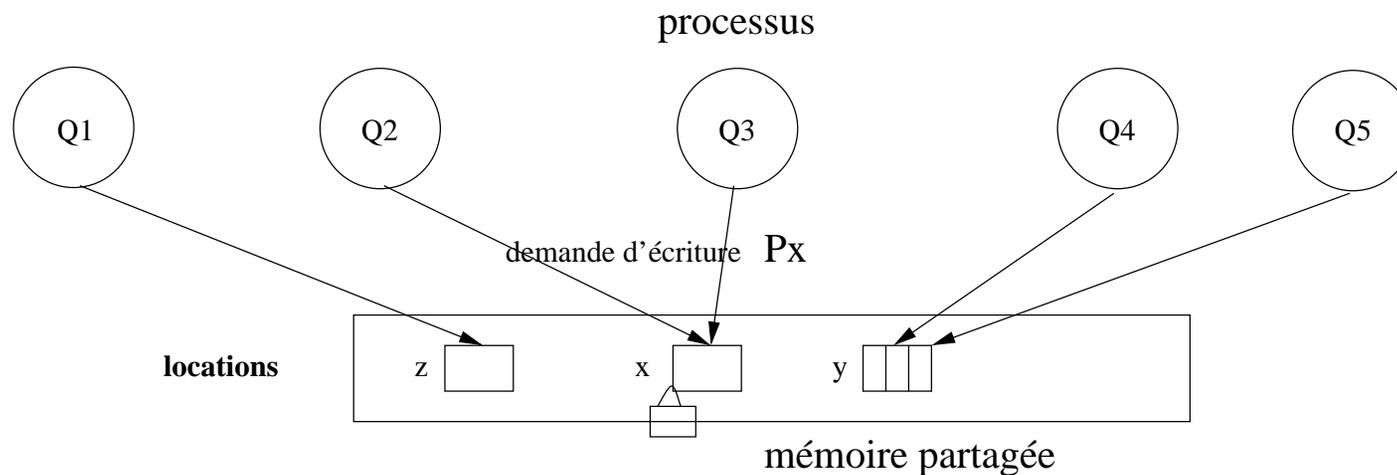
- Soit $T = (S, i, E, Tran)$ un système de transitions asynchrones.
On définit

$$an(T) = (B, M_0, E, pre, post)$$

- B est l'ensemble des conditions de T
- $M_0 = M(i)$
- pre et $post$ sont donnée par les opérations correspondantes dans T

Exercice: écrire un système de transitions asynchrones donnant la sémantique vraiment parallèle de $a \mid b$. Quel est le réseau de Petri correspondant? Les marquages possibles? Quel est le système de transitions asynchrones correspondant?

Machine parallèle à mémoire partagée



≠ séquentiel; problème de coordination (verrous P, V)

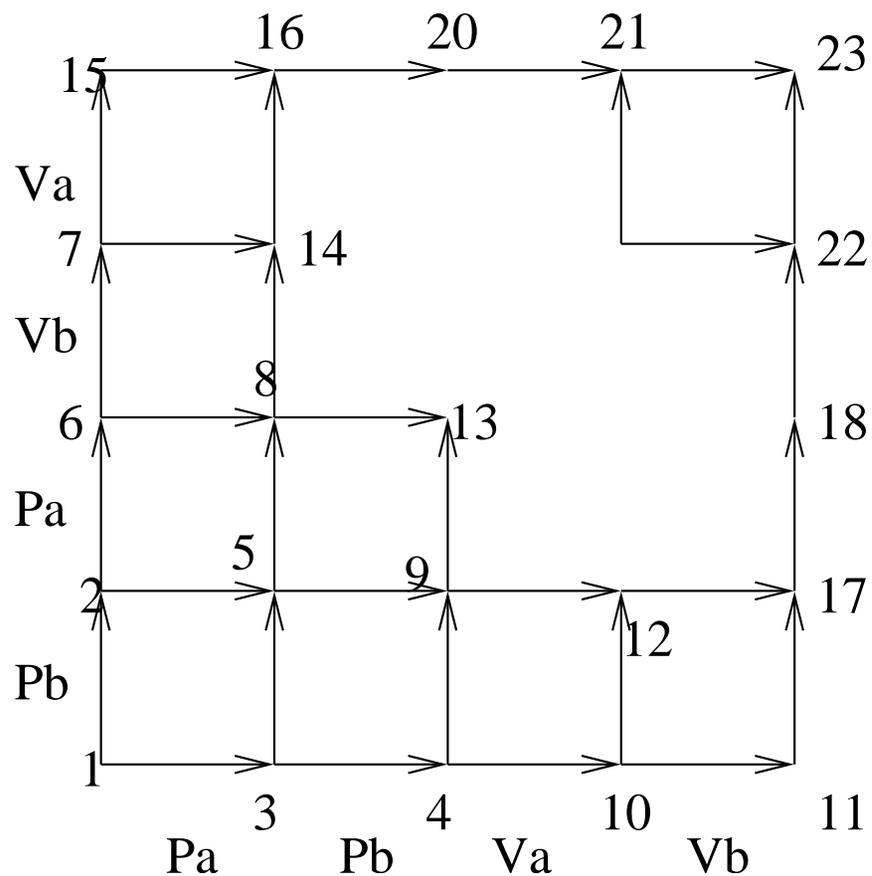
Coordination trop contrainte? → points morts

“Mauvais” états? → atteignabilité

Suites d'accès “corrects”? → chemins...

Un premier modèle

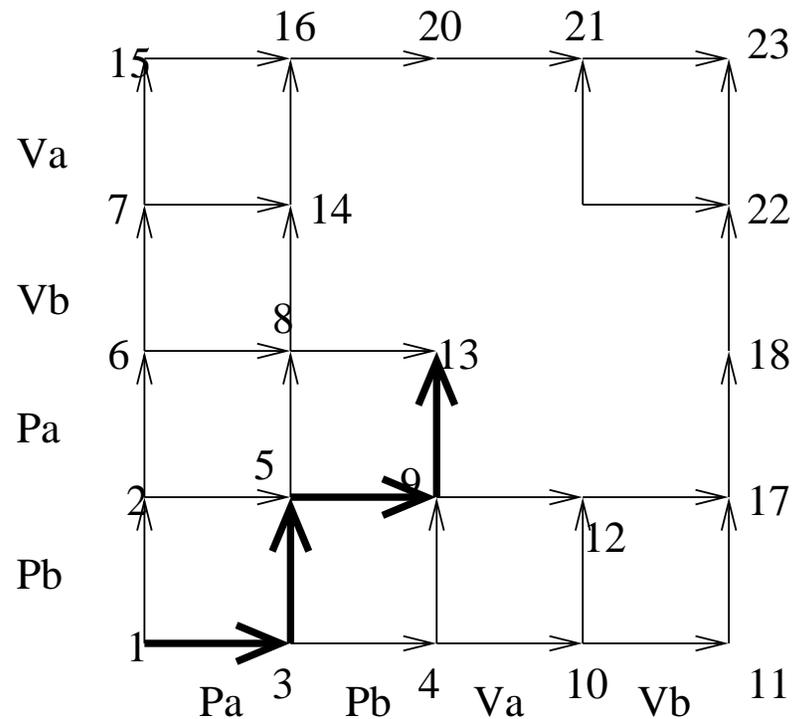
Processus: systèmes dynamiques (discrets en général) = graphes dirigés d'actions: (ou traces de Mazurkiewicz etc.)



Exemple d'exécution possible

Programme $(T_1 = PaPbVaVb) \mid (T_2 = PbPaVbVa)$:

T_1	T_2
Pa	—
—	Pb
$Pb(?)$	—
—	$Pa(?)$

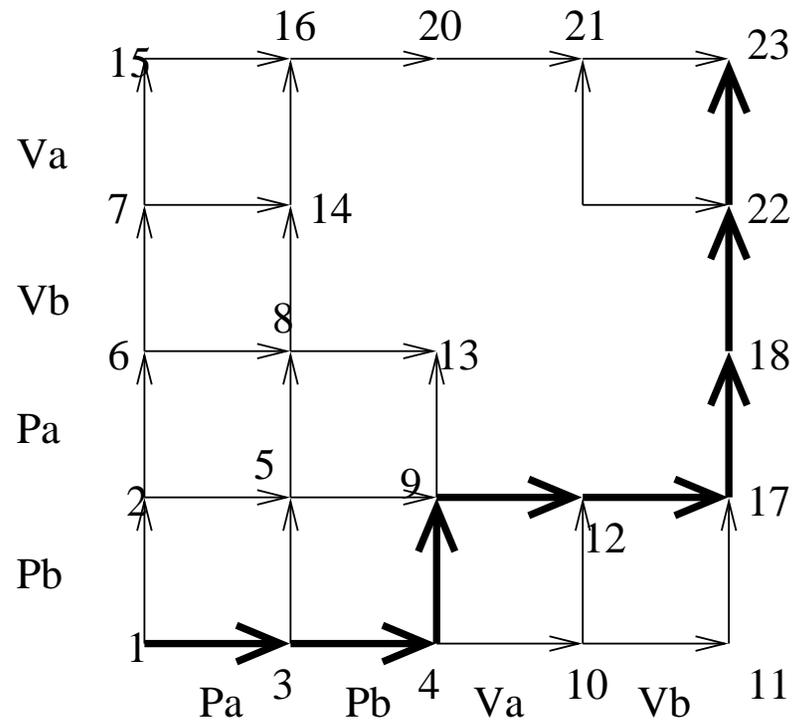


Point mort!

Exemple

1 autre exécution possible, complète (au total 16!):

T_1	T_2
Pa	—
Pb	—
—	$Pb(?)$
Va	—
Vb	—
—	Pa
—	Va
—	Vb



Remarquez qu'au fond...

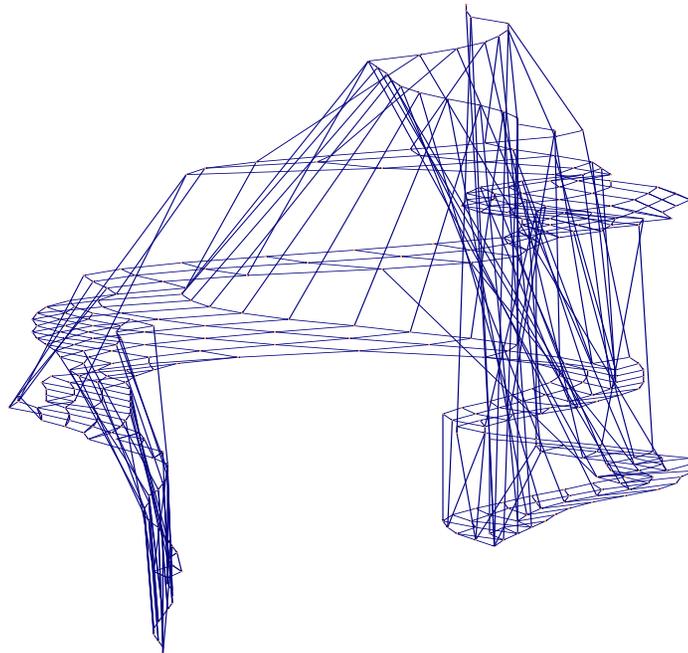
... il y a très peu de chemins "intéressants"

En fait $T_1 = PaPb(b = b + 1)VaVb$ et $T_2 = PbPa(b = b * 2)VbVa$ et au début, $b = 2$:

- Les 5 chemins les plus en haut à gauche sont "équivalents" à T_2 puis T_1 : ils calculent $b = 5$ ($2*2+1$)
- Les 5 chemins les plus en bas à droite sont "équivalents" à T_1 puis T_2 : ils calculent $b = 6$ ($(2+1)*2$)
- Les 6 chemins près de la diagonale sont équivalents: ils ne terminent pas

En général, pénible...

Grand nombre d'états: ici algorithme de Dekker, quelques lignes de C pour 2 processus, centaines de chemins / 2^t "vrais" chemins!



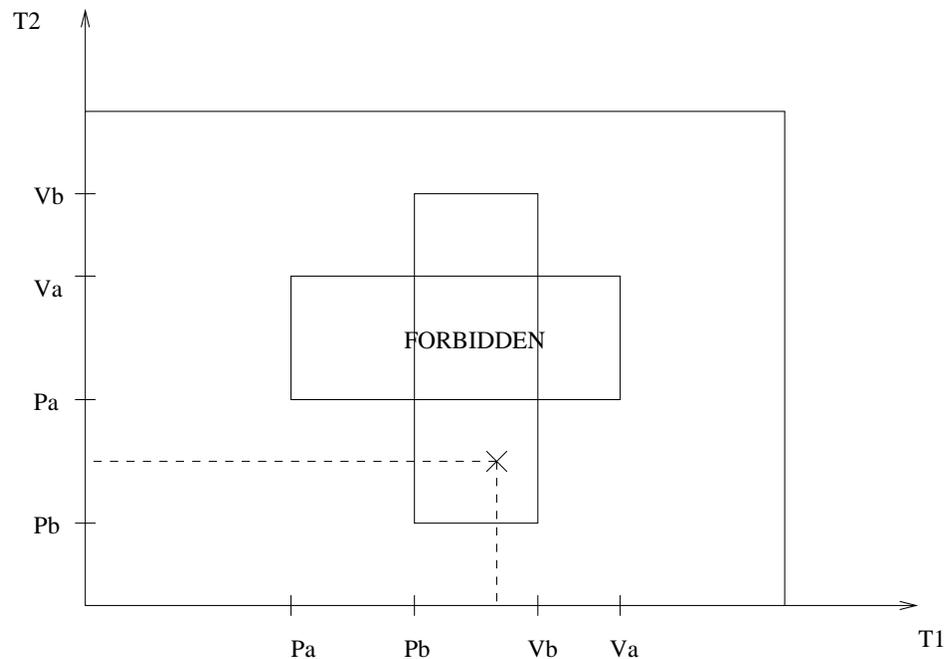
Principe d'un analyseur géométrique

- Sémantique continue (**déformable**)
- Catégorie de chemins modulo déformation (**composantes**)
- Chemins continus **essentiels**
- Traces séquentielles correspondant à ces chemins
- Analyse séquentielle classique

Géométrie?

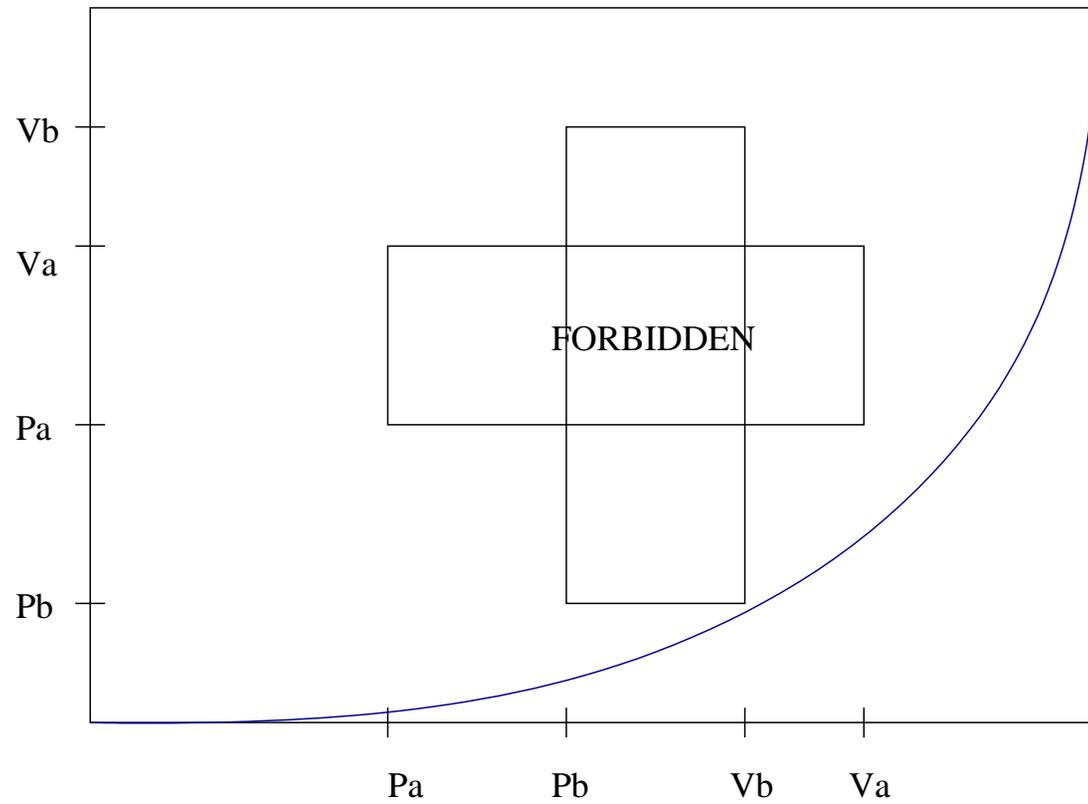
Première idée: "Progress graphs" (E.W.Dijkstra (1968))

$T1=Pa.Pb.Vb.Va$ en parallèle avec $T2=Pb.Pa.Va.Vb$



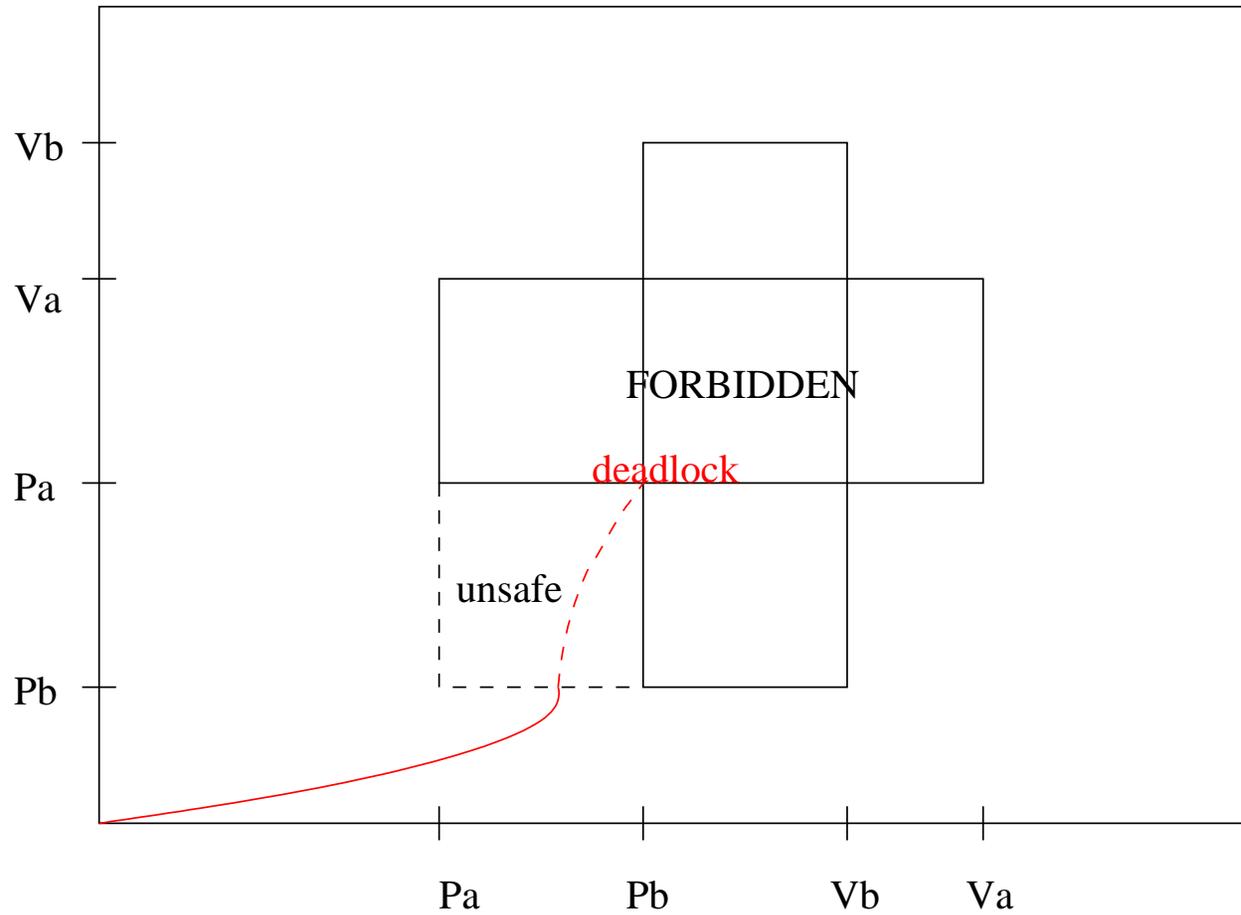
"Image continue": $x_i =$ temps local; rectangles hachurés=interdits!

Chemins d'exécutions

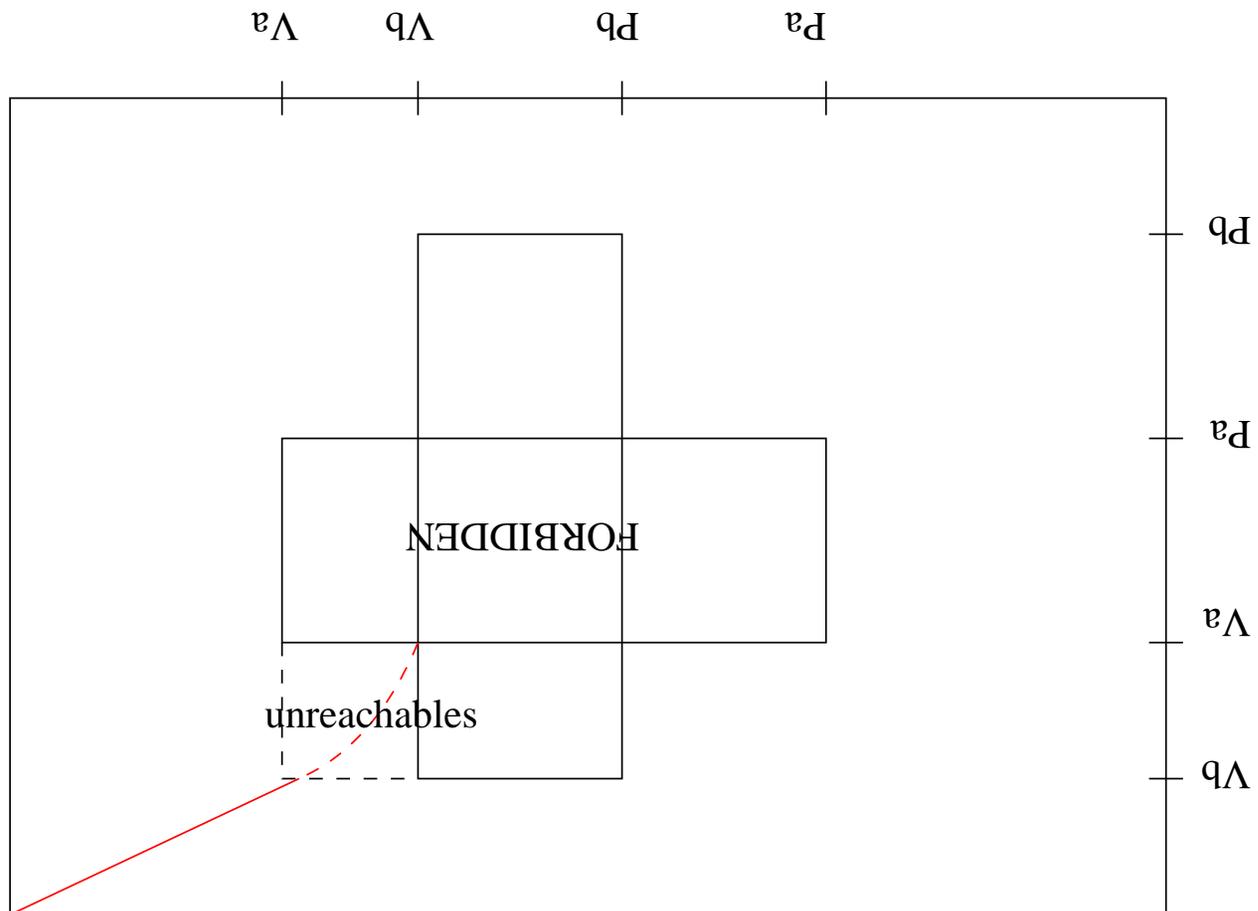


Traces = chemins croissants en chaque coordonnée = "di-chemins"

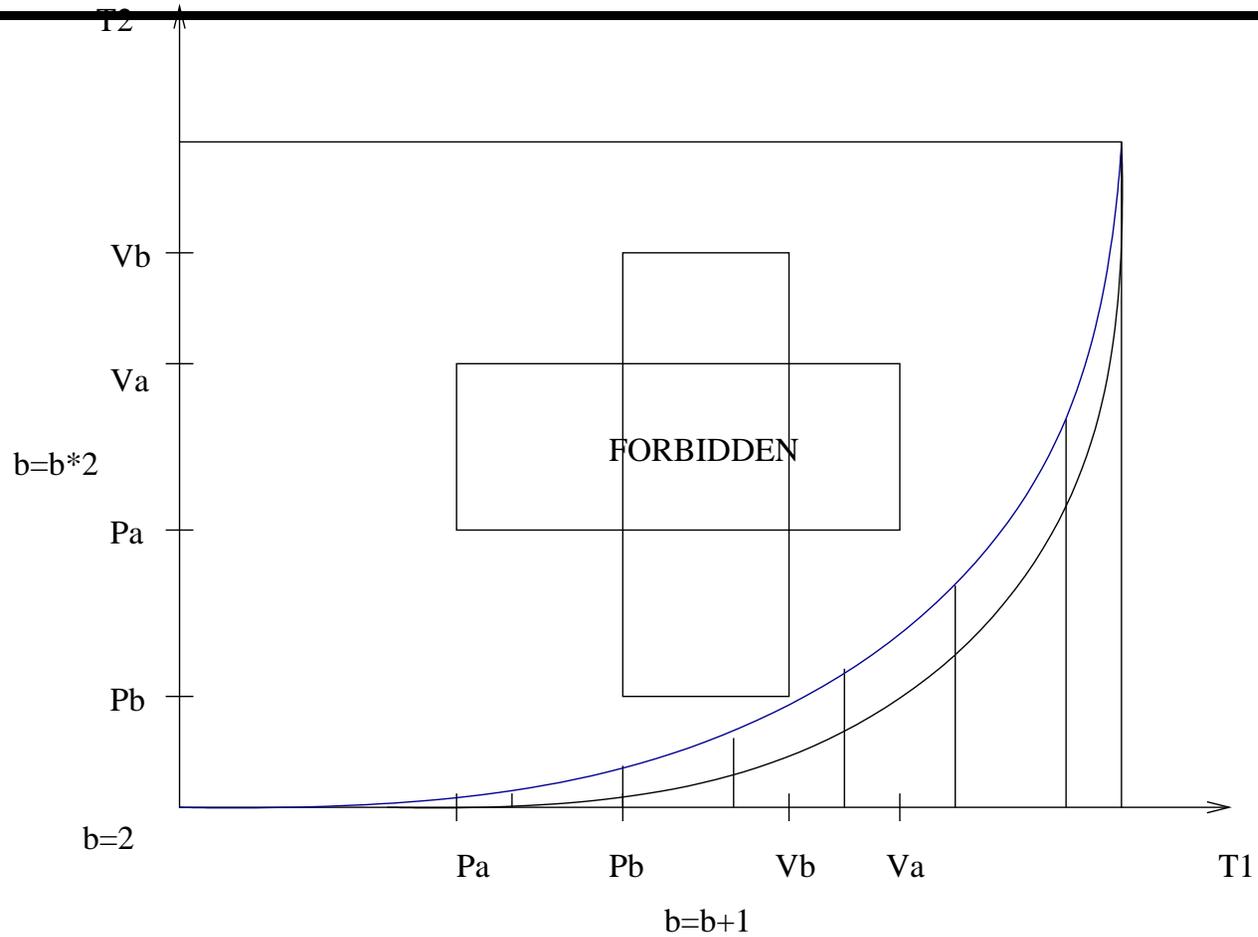
Première remarque



Et de faon duale...

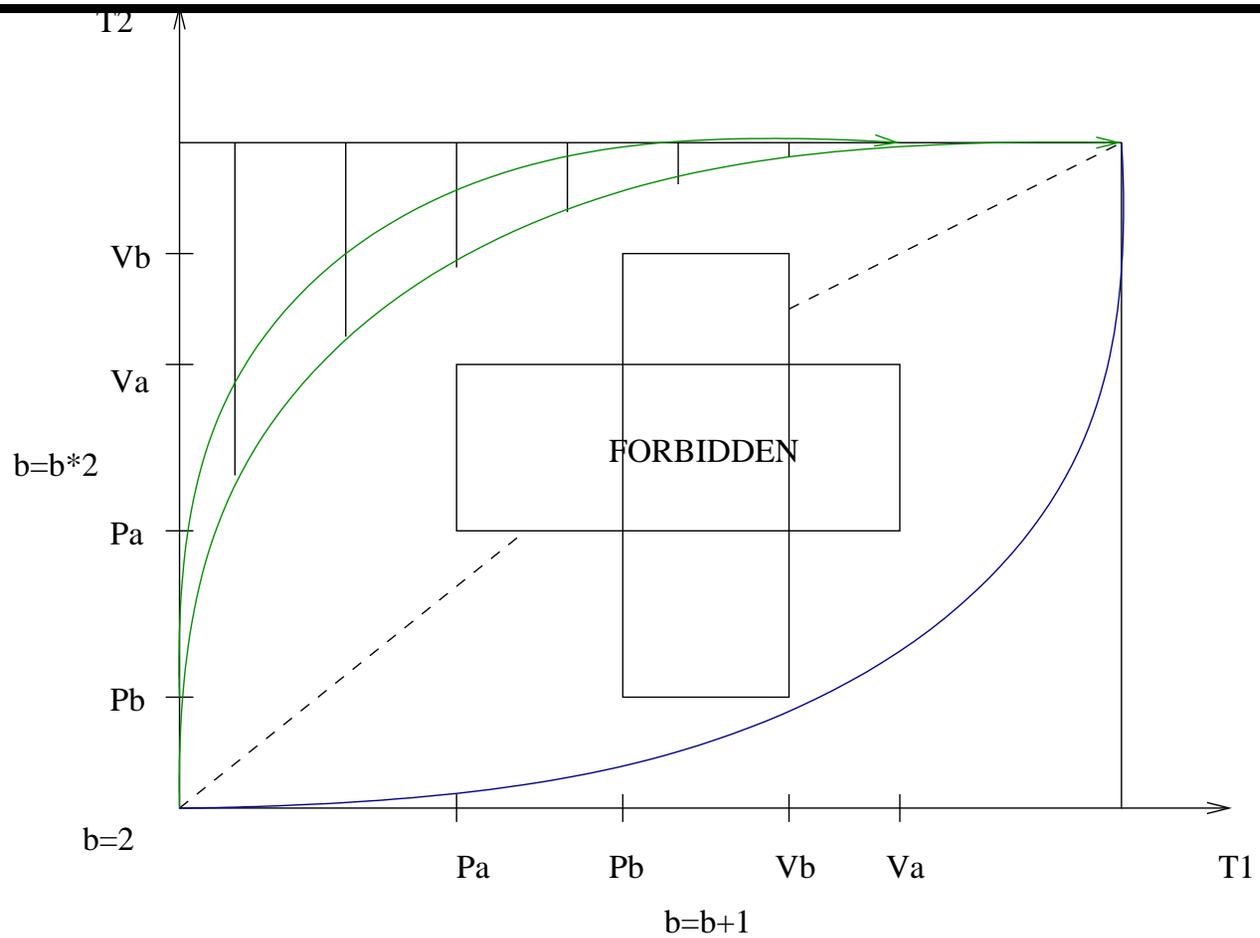


Déformation des chemins d'exécution



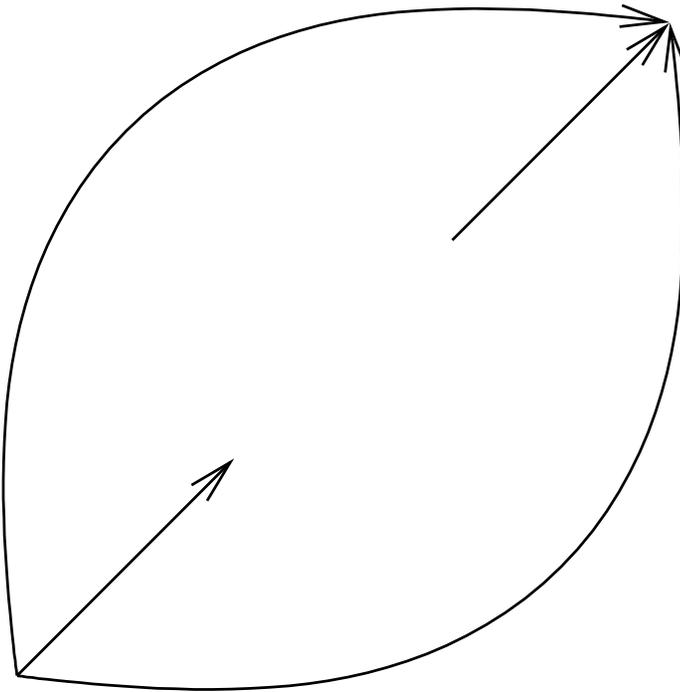
T1 gets a and b before T2 does: $b=6$

Chemins non-déformables

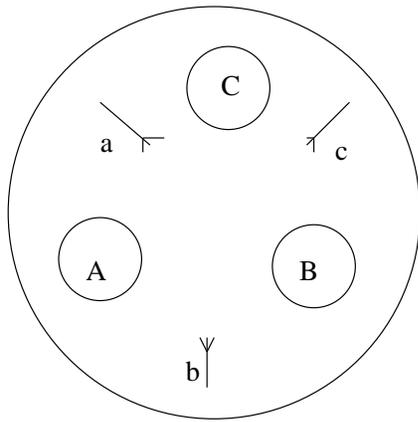


T2 gets a and b before T1 does: $b=5!$

Idéalement (?)...



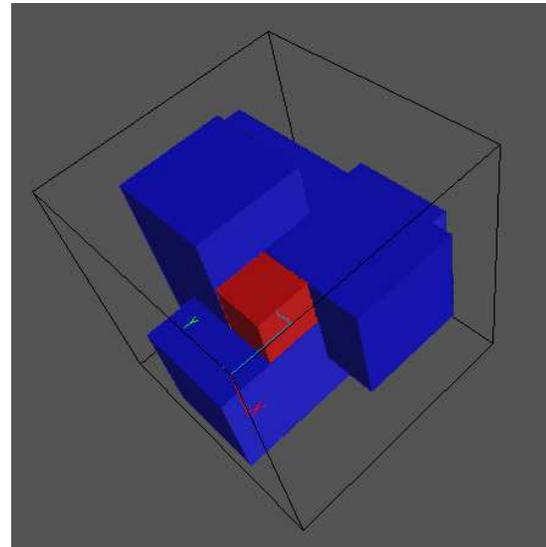
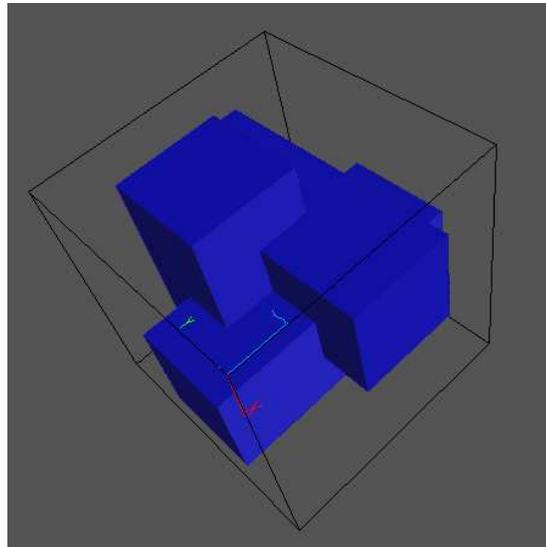
En dimension supérieure



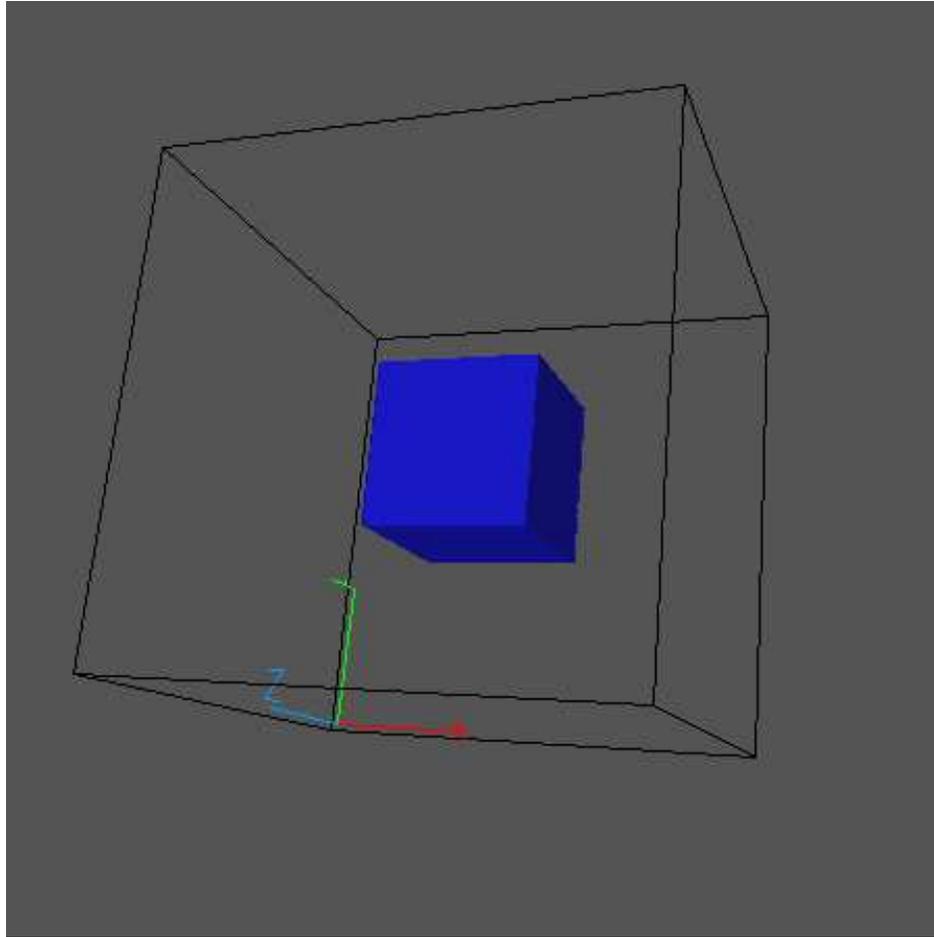
$$A = Pa \cdot Pb \cdot Va \cdot Vb$$

$$B = Pb \cdot Pc \cdot Vb \cdot Vc$$

$$C = Pc \cdot Pa \cdot Vc \cdot Va$$



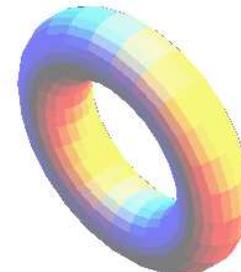
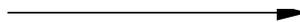
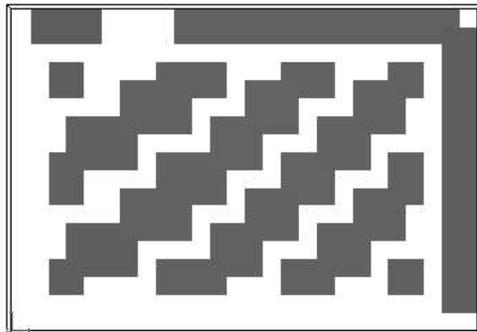
Un sémaphore à compteur



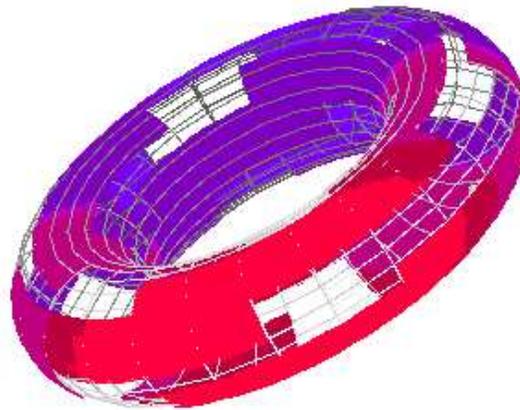
Boucles?

$$A = P_d \cdot P_a \cdot (P_b \cdot V_a \cdot V_d \cdot P_c \cdot V_b \cdot P_a \cdot P_d \cdot V_c \cdot P_b \cdot V_a \cdot P_c \cdot V_b \cdot P_a \cdot V_c \cdot P_b \cdot V_a \cdot P_c \cdot V_b \cdot P_a \cdot V_c)^* \cdot V_a \cdot P_e \cdot V_d \cdot V_e$$

$$B = P_e \cdot P_a \cdot (P_b \cdot V_a \cdot P_c \cdot V_b \cdot P_a \cdot V_c \cdot P_b \cdot V_a \cdot P_c \cdot V_b \cdot P_a \cdot V_c)^* \cdot V_a \cdot P_d \cdot V_e \cdot V_d$$



Donne...

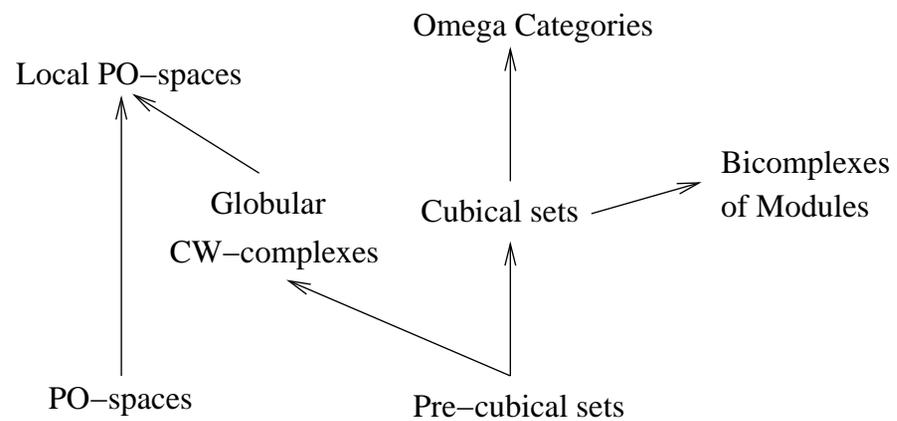


Pas de deadlock, mais 2 tours de boucle T_1 / 3 tours pour T_2

Correspondances (ou presque)

Modèle [discret]	complexes combinatoires
Modèle [continu]	espaces topologiques
Relation discret/continu	réalisation géométrique
Composition parallèle	produit
Raffinement d'action	subdivision
Compositionnalité	Seifert/van Kampen
Points morts/atteignabilité	composantes connexes
Ordonnements	groupe fondamental
Equivalence observationnelle	équivalence d'homotopie (faible/forte)
Propriétés calculables	invariants topologiques (homologie etc.)

Plus formellement



Plus formellement...

Pour les “graphes de processus”, on a seulement besoin de (MFPS’98):

Un espace topologique avec un ordre partiel (global) fermé

- Les morphismes sont les fonctions **continues et croissantes**
- Les traces [finies] sur $(X, <)$ sont les morphismes de $\vec{I} = ([0, 1], \leq)$ vers $(X, <)$.
- Les dihomotopies entre les dichemins f et g **avec extrémités fixes** sont les morphismes $H : \vec{I} \times I \rightarrow X$ tels que pour tout $x \in \vec{I}$, $t \in I$,

$$H(x, 0) = f(x), \quad H(x, 1) = g(x) \quad H(0, t) = \alpha, \quad H(1, t) = \beta$$

“Vrai” parallélisme géométrique

- Premières images et algorithmes: Dijkstra (1968), Carson et Reynolds (71) etc.
- Pratt, V. (1991). Modeling concurrency with geometry. In Proc. of the 18th POPL. [HDA, n -cat, monoidal homotopy etc.]
- van Glabbeek, R. (1991). Bisimulation semantics for higher dimensional automata. Technical report, Stanford University, <http://theory.stanford.edu/~rvg/hda>. [“truly-concurrent” bisimulation]
- Goubault, E. and Jensen, T. P. (1992). Homology of higher-dimensional automata. In Proc. of CONCUR’92. [CCS, bisimulation, homology]

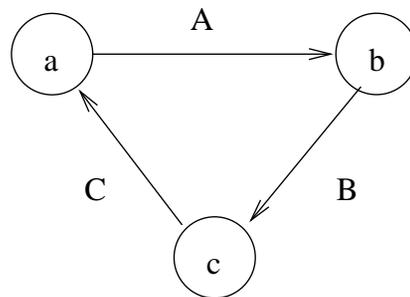
Historique

- Séquentialisabilité (J. Gunawardena 1994, puis L. Fajstrup, M. Raussen, E. Goubault 1999),
- Analyse d'ordonnement par IA (E. Goubault 1995),
- Model-checking abstrait (R. Cridlig, 1995),
- Analyse de points morts, atteignabilité (L. Fajstrup, E. Goubault, M. Raussen 1998),
- Ordonnements essentiels (M. Raussen 2000),
- Relations avec les systèmes distribués tolérants aux pannes (M. Herlihy, S. Rajsbaum, E. Goubault etc.)

Est-ce une vision "utile"?

Une réponse: détection de points morts, voir L. Fajstrup, E. Goubault, M. Raussen "Detecting Deadlocks in Concurrent Systems" Concur'98+...

Une condition classique **suffisante** pour qu'un système parallèle soit sans point mort est que le graphe de requête soit **acyclique** (condition géométrique...)

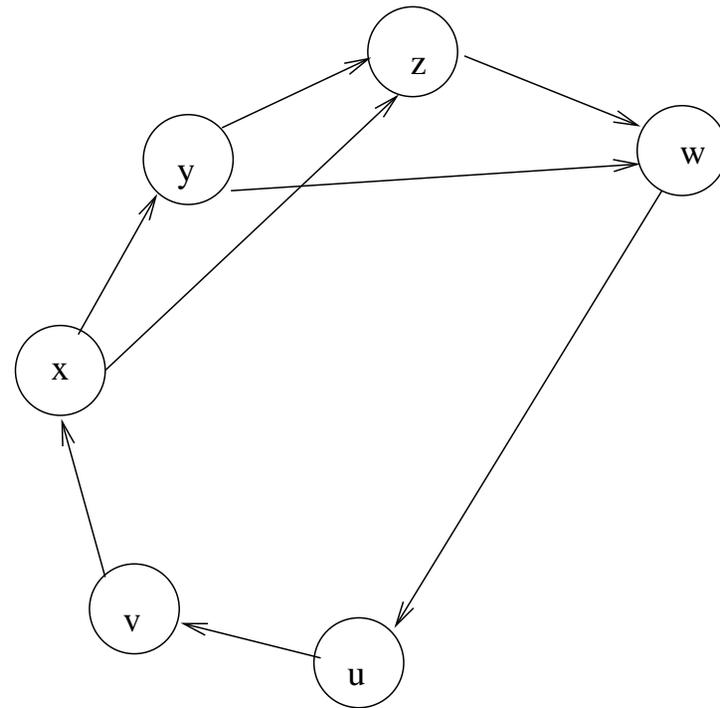


Mais...[Lipsky/Papadimitriou]

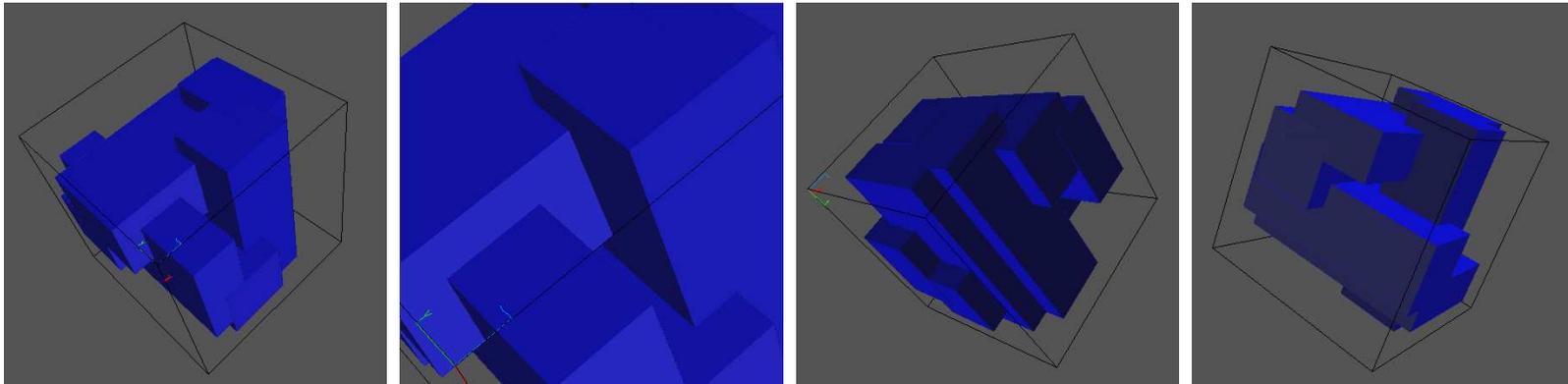
$A = Px . Py . Pz . Vx . Pw . Vz . Vy . Vw$

$B = Pu . Pv . Px . Vu . Pz . Vv . Vx . Vz$

$C = Py . Pw . Vy . Pu . Vw . Pv . Vu . Vv$



Graphe de requête...

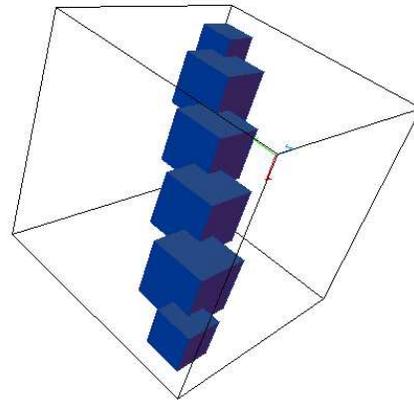
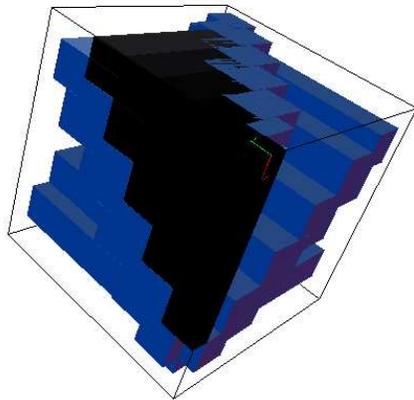


Egalement: effet du partage

A=Pa . Pb . Va . Pc . Vb . Pd . Vc . Pe . Vd . Pf . Ve . Vf

B=Pf . Pe . Vf . Pd . Ve . Pc . Vd . Pb . Vc . Pa . Vb . Va

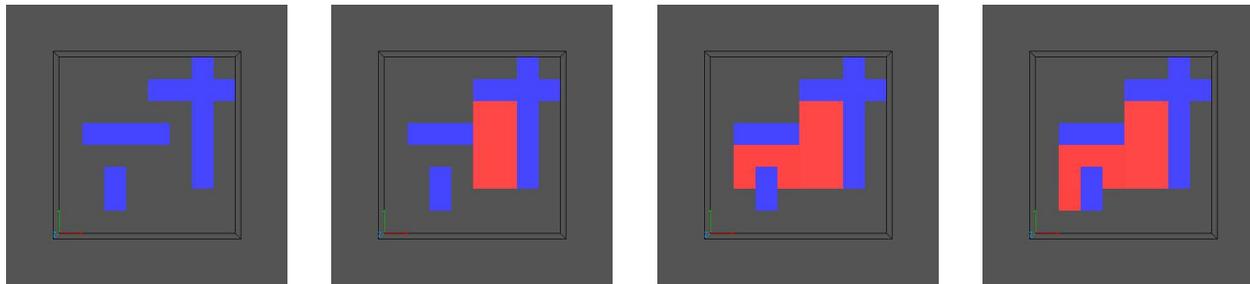
C=Pf . Pe . Vf . Pd . Ve . Pc . Vd . Pb . Vc . Pa . Vb . Va



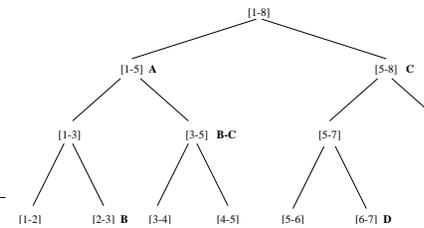
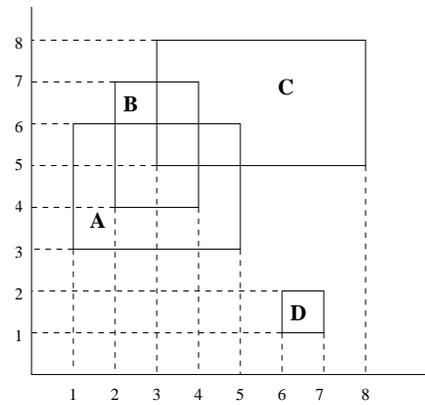
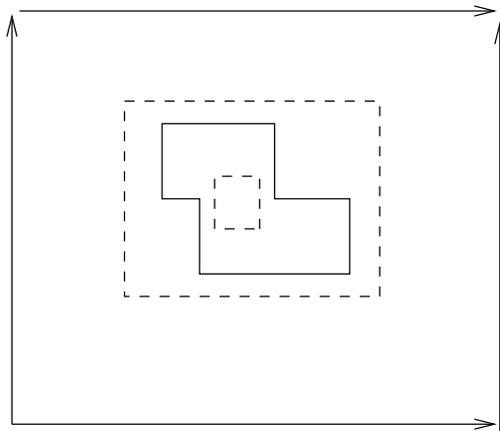
Idée (simplifiée) de l'algorithme

CONCUR'98-rapport CEA-implémentation (ENS)

- Ne calculer que la région interdite (une abstraction),
- Les intersections J des n (=dimension) hyperrectangles interdits $R^i = [a_1^i, b_1^i] \times \cdots \times [a_n^i, b_n^i]$ créent les points morts,
- Interdire les hyperrectangles $[\tilde{x}, x]$, où $x = \min J = (\max_i a_1^i, \cdots, \max_i a_n^i)$ et $\tilde{x} = (2^{\text{ème}} \max_i a_1^i, \cdots, 2^{\text{ème}} \max_i a_n^i)$,

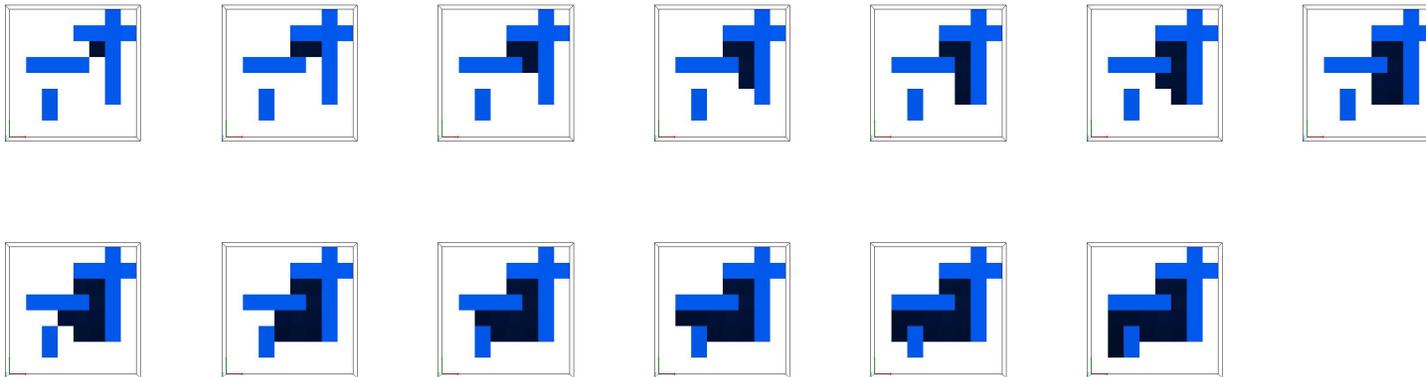


Abstraction et algorithmique



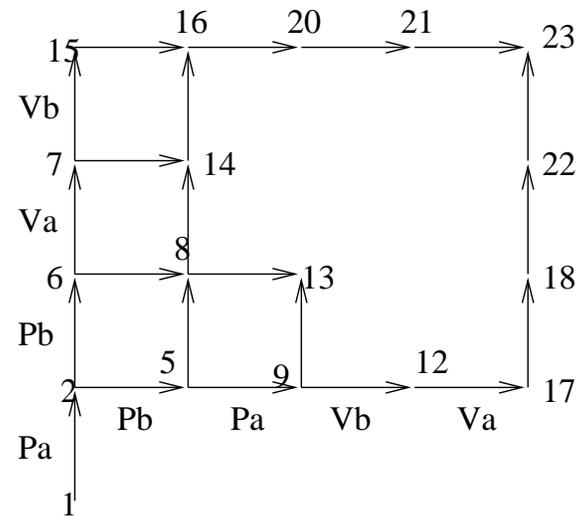
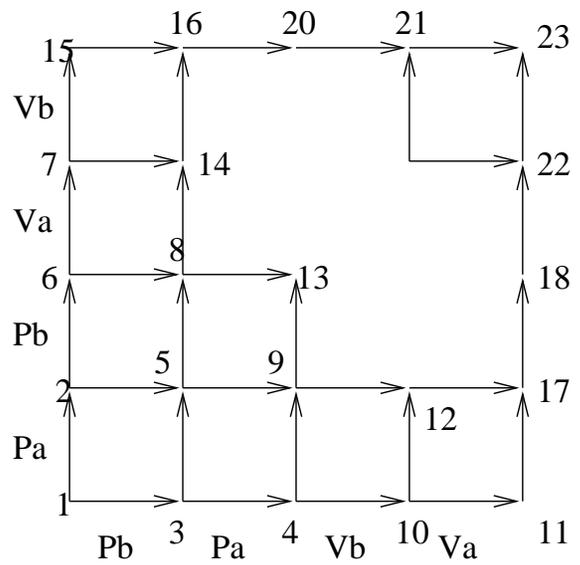
[*rapport CEA'99*]

Au lieu de l'exploration naive de l'espace d'états



[en fait, plus rapide même que les ensembles [persistants](#)
Godefroid/Wolper, cf. Eric Goubault et Martin Raussen LATIN'02
etc.]

Persistent sets



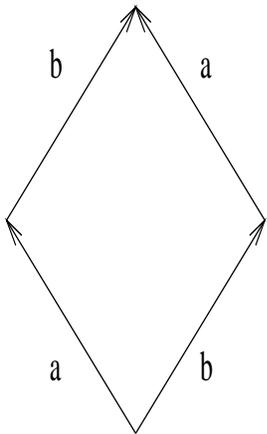
Généralisation

- Au moins... ensembles précubiques (ou cubiques) - utilisé dans ma thèse et mes premiers articles, *CONCUR'92* avec T. P. Jensen, *CONCUR'93* et l'article de V. Pratt de 1991, donnant une sémantique CCS (*CONCUR'92*), la mémoire partagée (*CONCUR'98*) etc.
- On a besoin de **branchements**, **boucles**...
→ on n'a pas un ordre partiel **global!**

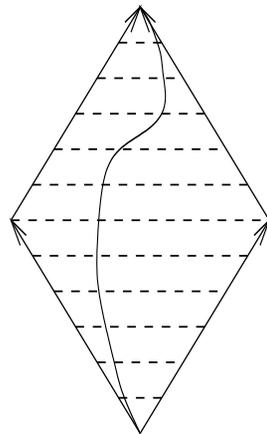
Vision discrète

Premières intuitions: Vaughan Pratt (POPL'90) "Modeling Concurrency with Geometry" (ensembles cubiques)

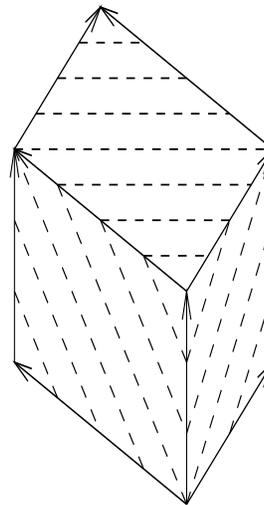
2 processus, 1 processeur



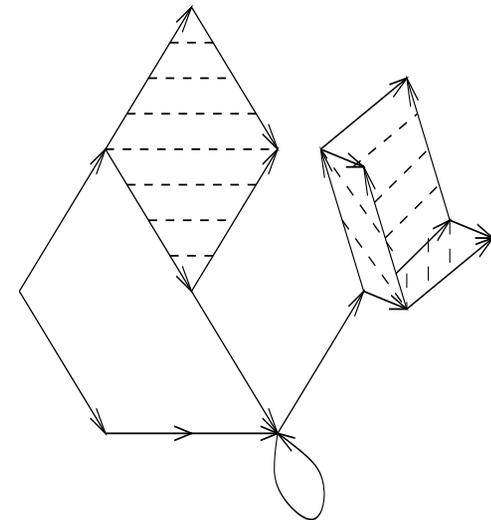
2 processus, 2 processeurs



3 processus, 3 processeurs



ensemble cubique



Ensembles précubiques

Ensembles de “ n -transitions” avec des opérateurs sources et buts:

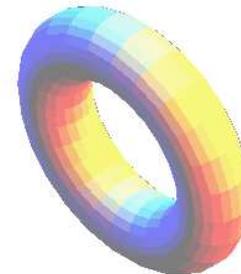
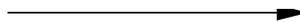
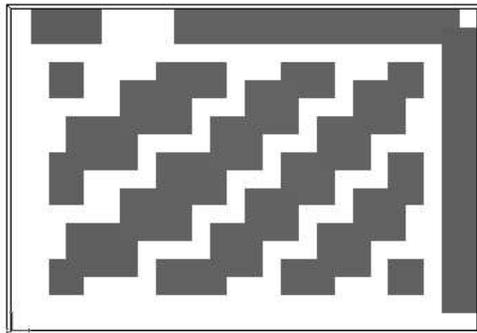
- Extension simple de la notion de source et de but dans les graphes dirigés,
- Par exemple, en dimension 2: le carré [relation d'indépendance],
2 sources d_0^0 et d_1^0 , avec $\partial_0^0(A) = a$, $\partial_1^0(A) = b$ 2 buts d_0^1 et d_1^1 , avec

$$\partial_0^1(A) = a', \quad \partial_1^1(A) = b'$$

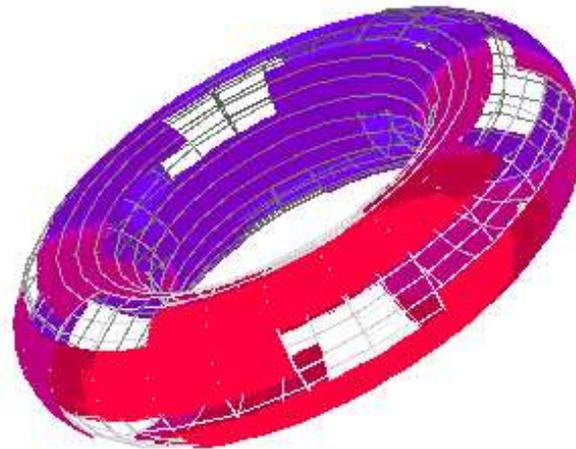
Boucles?

$$A = P_d \cdot P_a \cdot (P_b \cdot V_a \cdot V_d \cdot P_c \cdot V_b \cdot P_a \cdot P_d \cdot V_c \cdot P_b \cdot V_a \cdot P_c \cdot V_b \cdot P_a \cdot V_c \cdot P_b \cdot V_a \cdot P_c \cdot V_b \cdot P_a \cdot V_c)^* \cdot V_a \cdot P_e \cdot V_d \cdot V_e$$

$$B = P_e \cdot P_a \cdot (P_b \cdot V_a \cdot P_c \cdot V_b \cdot P_a \cdot V_c \cdot P_b \cdot V_a \cdot P_c \cdot V_b \cdot P_a \cdot V_c)^* \cdot V_a \cdot P_d \cdot V_e \cdot V_d$$



Donne...

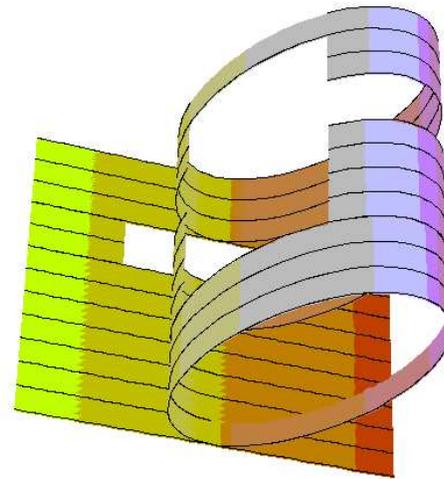
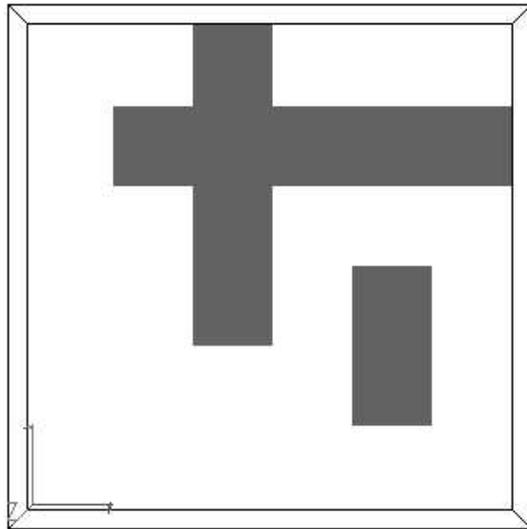


→ po-espace local (*MFPS'98*); définition similaire à une variété différentielle, localement un po-espace.

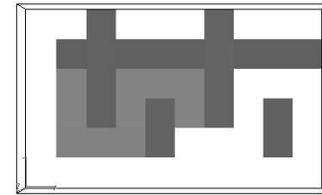
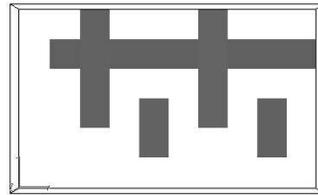
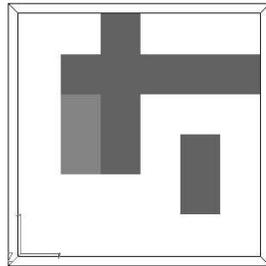
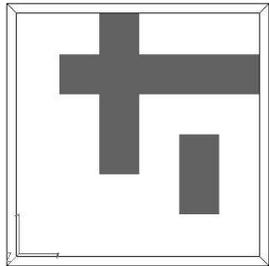
Les boucles

$$A = P_a \cdot (P_b \cdot V_b \cdot P_c \cdot V_c) \cdot V_a$$

$$B = P_c \cdot P_b \cdot V_c \cdot P_a \cdot V_a \cdot V_b$$



Région dangereuse?



Résultats (L. Fajstrup)

- Points morts calculables en un déroulement,
- Algorithme région dangereuse et région inatteignable avec boucles, en examinant certains déroulements seulement,

Séquentialisation

Valeurs initiales: $A=0, B=0$:

R: $A=1$ R: $A=1$ R: $A=1$ S: $B=2$ S: $B=2$ S: $B=2$
R: $B=3$ S: $B=2$ S: $B=2$ R: $A=3$ R: $A=3$ S: $A=6$
S: $B=5-$ R: $B=3-$ S: $A=4-$ S: $A=4-$ R: $B=3-$ R: $A=3-$
S: $A=10-$ S: $A=6-$ R: $B=3-$ R: $B=3-$ S: $A=6-$ R: $B=3-$

Seules la première ($A=10, B=5$) et dernière trace ($A=3, B=3$) sont correctes.

Protocole "2-phases"

R:

P A;

P B;

A:=B+1;

B:=3;

V B;

V A;

S:

P A;

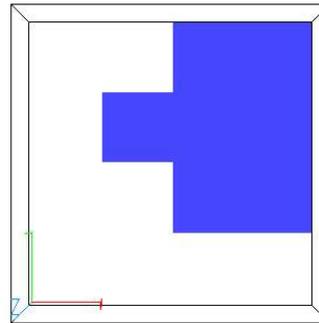
P B;

B:=2;

A:=2*B;

V B;

V A;



Est-ce utile?

Une deuxième réponse: Correction des bases de données distribuées (séquentialisation): “Toutes les traces doivent être *équivalentes* à une exécution des transactions, l’une après l’autre”

Condition suffisante: **graphe de tâches**.

Condition nécessaire et suffisante dans un cas particulier:

polygraphe de tâches (assez complexe) Papadimitriou/Yannakakis.

Origine de la difficulté: **connexité** de la région interdite en

dimension 2 est équivalente à la séquentialisation. Ce n’est ni **suffisant ni nécessaire** en dimension supérieure...**voir encore**

Lipsky/Papadimitriou!

Plus généralement...

Toute **caractérisation** (et/ou abstraction) de l'ordonnancement des actions (accès aux ressources partagées etc.) **modulo** "commutation" des actions donne:

- preuve de la correction ou de l'équivalence de systèmes; voir par exemple E. Goubault "Abstract Interpretation of Schedulers", *PEPM'95*, ou L. Fajstrup et M. Raussen, *MFPS'98*
- preuve ou analyse statique de programmes parallèles (sur des "ordonnements essentiels"); **réduction de l'espace d'états**

Plus généralement...

- Definissabilité et complexité de l'implémentation de protocoles pour les systèmes distribués tolérants aux pannes (voir M. Herlihy, S. Rajsbaum etc. Premiers pas dans E. Goubault "Optimal Construction of Wait-Free Binary Relations", *CAAP'97* ou *ICW'96*)
- etc.

Survey dans mon article "Geometry and Concurrency: A User's Guide" *Mathematical Structures in Computer Science'00*.

Homéomorphismes d'espaces topologiques

X et Y deux espaces topologiques:

- $f : X \rightarrow Y$ est **continue** si $f^{-1}(U) := \{x \in X \mid f(x) \in U\} \subset X$ est *ouvert* pour tout ouvert $U \subseteq Y$.
- Une fonction continue $f : X \rightarrow Y$ est un **homéomorphisme** si
 - c'est une bijection
 - son inverse $f^{-1} : Y \rightarrow X$ est continue également
- Deux espaces topologiques X, Y sont **homéomorphes** si il existe un homéomorphisme $f : X \rightarrow Y$.

Mais...

- Classification des espaces topologiques généraux (modulo homéomorphismes) trop compliquée
- Par contre, existence d'"invariants" plus ou moins simples:
 - nombre de composantes connexes (ou connexes par arcs)
 - caractéristique d'Euler-Poincaré, nombres de Betti etc.
 - plus généralement groupes d'homotopie, groupes d'homologie, anneaux de cohomologie, K-théorie etc.

Chemins dans un espace topologique

- Soit $I = [0, 1]$ l'intervalle unité et soit X un espace topologique. Un **chemin** dans X est une fonction **continue** $\alpha : I \rightarrow X$. Ses extrémités sont $x_0 = \alpha(0)$ et $x_1 = \alpha(1)$.
- Un chemin dans X avec $x_0 = \alpha(0) = \alpha(1) = x_1$ est une **boucle** dans X .
- La **concaténation** (pas **associative** en général) de α_1 et α_2 est non définie sauf si $\alpha_1(1) = \alpha_2(0)$. Dans ce cas:

$$(\alpha_1 * \alpha_2)(s) = \begin{cases} \alpha_1(2s), & t \leq \frac{1}{2} \\ \alpha_2(2s - 1), & t \geq \frac{1}{2}. \end{cases}$$

Composantes connexes par arcs

- Un espace topologique X est **connexe par arc** si quelque soient $x_0, x_1 \in X$, il existe un chemin $\alpha : I \rightarrow X$ avec $\alpha(0) = x_0$ et $\alpha(1) = x_1$.
- Les **composantes connexes par arcs** de $x \in X$ sont les *ouverts et fermés* $\{y \in X ; | ; \exists \alpha : I \rightarrow X \text{ avec } \alpha(0) = x \text{ et } \alpha(1) = y\}$.

Homotopie

- Une **homotopie** est une famille à un paramètre de fonctions continues $H_t : X \rightarrow Y$, ; $t \in I$ of maps, décrite par une fonction continue $H : X \times I \rightarrow Y$.
- Deux fonctions continues $f, g : X \rightarrow Y$ sont **homotopes** ($f \simeq g$) si il existe une homotopie $H : X \times I \rightarrow Y$ telle que $H_0(x) = H(x, 0) = f(x)$ et $H_1(x) = H(x, 1) = g(x)$ pour tout $x \in X$.

Equivalence d'homotopie

- Une fonction continue $f : X \rightarrow Y$ est appelée **équivalence d'homotopie** si il existe une fonction continue $g : Y \rightarrow X$ et deux homotopies entre $g \circ f : X \rightarrow X$ and id_X , resp. $f \circ g : Y \rightarrow Y$ et id_Y .
- Deux espaces X et Y sont **homotopiquement équivalents** si il existe une équivalence d'homotopie $f : X \rightarrow Y$.

Homotopie des boucles

Soit X espace topologique, et $x_0 \in X$ un *point base*.

- Un chemin $\alpha : I \rightarrow X$ est une **boucle** de point base x_0 si $\alpha(0) = \alpha(1) = x_0$. Cet ensemble de boucles est noté $P_1(X; x_0)$.
- La concaténation définit une opération binaire $C : P_1(X; x_0) \times P_1(X; x_0) \rightarrow P_1(X; x_0)$.
- Une *homotopie* de boucles en x_0 est une famille de boucles $H_t : I \rightarrow X$ en x_0 telle que $H : I \times I \rightarrow X$, ; $H(x, t) = H_t(x)$ soit continue.
- Deux boucles α et β en x_0 sont **homotopes** ($\alpha \simeq \beta$) si il existe une homotopie H_t de boucles avec $H_0 = \alpha$ et $H_1 = \beta$.

Groupe fondamental

- L'ensemble des classes d'équivalences des boucles en x_0 modulo homotopie de boucle est notée $\pi_1(X; x_0)$.
- La concaténation passe au quotient et définit

$$C : \pi_1(X; x_0) \times \pi_1(X; x_0) \rightarrow \pi_1(X; x_0).$$

On note $[\alpha] * [\beta]$ pour $C([\alpha], [\beta])$.

- $\pi_1(X; x_0)$ avec l'opération $*$ est un groupe, le **groupe fondamental** de X avec point base x_0 .

Le groupe fondamental: propriétés et exemples

- Si $x_0, x_1 \in X$ sont dans la même composante connexe par arcs, les groupes fondamentaux correspondants sont **isomorphes**:
 $\pi_1(X; x_0) \cong \pi_1(X; x_1)$.
- $\pi_1(\mathbf{R}^n)$ est le groupe trivial (un seul élément: l'unité).
- Le groupe fondamental du **cercle** S^1 est isomorphe aux entiers:
 $\pi_1(S^1) \cong \mathbf{Z}$.
- Le groupe fondamental d'une **sphère** S^n , ; $n > 1$ est trivial:
 $\pi_1(S^n) \cong 0$, ; $n > 1$.
- Le groupe fondamental de la **figure 8** (collage en un point de deux cercles) est le groupe libre sur deux générateurs et est donc *non* commutatif.

Homomorphismes induits: Définition et Propriétés

- Une fonction continue $f : X \rightarrow Y$ **induit** une fonction $f_{\#} : \pi_1(X, x_0) \rightarrow \pi_1(Y, f(x_0))$, i.e., $f_{\#}[\alpha] = [f \circ \alpha]$.
- $f_{\#}$ est un **homomorphisme de groupe**.
- Si $f_0, f_1 : X \rightarrow Y$ sont homotopes via une homotopie préservant les *points-bases* $H : X \times I \rightarrow Y$, alors les homomorphismes induits $f_{j\#} : \pi_1(X, x_0) \rightarrow \pi_1(Y, f(x_0))$ sont les *mêmes*.

Groupes d'homotopie d'ordre supérieur: Définition

- Les **boucles** peuvent être vues comme des fonctions $\alpha : S^1 \rightarrow X$ définies sur le **cercle**.
- Plus généralement, on peut considérer des fonctions continues $\alpha : S^n \rightarrow X$ préservant les points-bases – et les classier modulo homotopie.
- On obtient le **n ième groupe d'homotopie** $\pi_n(X)$ de X – groupe **abélien** pour $n > 1$.
- Une fonction continue $f : X \rightarrow Y$ induit un **homomorphisme** $f_{\#} : \pi_n(X) \rightarrow \pi_n(Y)$.
- **Invariant de l'homotopie**: deux fonctions **homotopes** induisent le **même** homomorphisme sur les groupes d'homotopie.

Groupes d'homotopie d'ordre supérieur: Propriétés et Exemples

- Les espaces **contractiles** (par exemple \mathbf{R}^n) ont tous leurs groupes d'homotopies triviaux (et **inversement** pour les **CW-complexes**).

- Groupes d'homotopie des sphères:

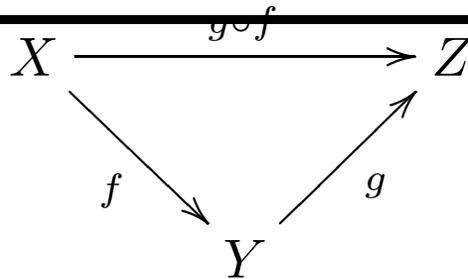
$$0 \quad k < n$$

$$\pi_k(S^n) = \mathbf{Z} \quad k = n$$

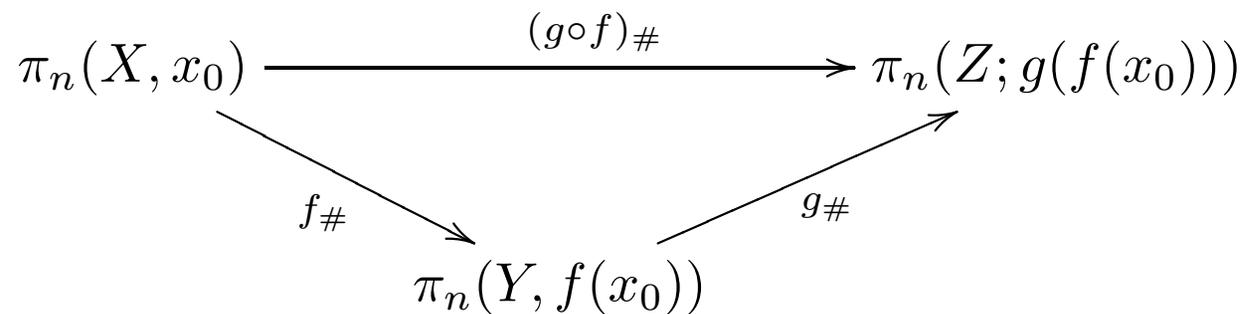
$$\text{pas connu} \quad k > n \text{ en général}$$

- Un espace X est **k -connexe** si $\pi_i(X) = 0$ pour $i \leq k$.

Homomorphismes induits: Naturalité



induit



- $id : X \rightarrow X$ induit $id : \pi_n(X : x_0) \rightarrow \pi_n(X : x_0)$.
- Des espaces homotopiquement équivalents ont des groupes d'homotopie **isomorphes**.

Application: théorème de Brouwer

Soit $B^n := \{x \in \mathbb{R}^n ; \|x\| \leq 1\}$.

- (BFT) Tout endomorphisme continu $f : B^n \rightarrow B^n$ a un point fixe $x_0 \in B^n : f(x_0) = x_0$.
- Lemme: il n'y a pas de fonction continue $r : B^n \rightarrow S^{n-1}$ étendant l'identité.

$$\begin{array}{ccc} S^{n-1} & \xrightarrow{id} & S^{n-1} \\ & \searrow i & \nearrow r \\ & & B^n \end{array}$$

$$\begin{array}{ccc} \mathbf{Z} & \xrightarrow{id} & \mathbf{Z} \\ & \searrow i_{\#} & \nearrow r_{\#} \\ & & 0 \end{array}$$

Application “classique”: Protocoles tolérants aux pannes

Existe-t-il un protocole qui gère N processus asynchrones tels que,

- ils démarrent avec certaines valeurs d'entrée spécifiées, calculent et communiquent, et terminent avec certaines valeurs de sortie spécifiées,

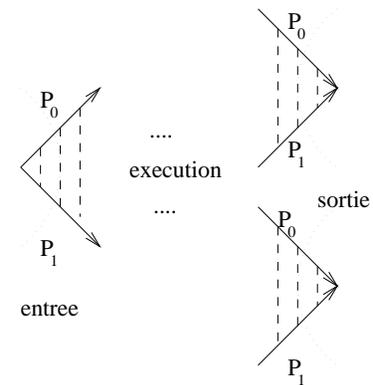
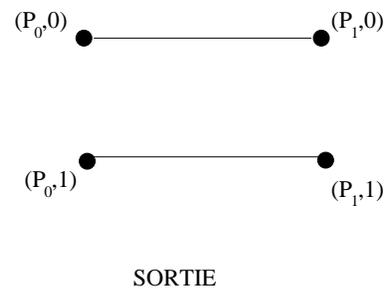
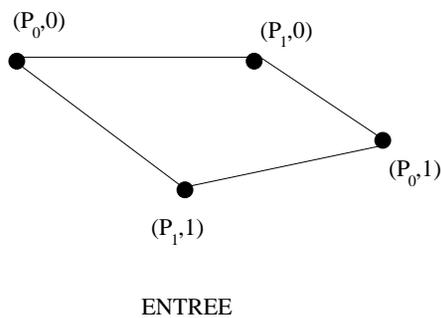
- ils fonctionnent selon un modèle de calcul spécifié (mémoire partagée, passage de messages etc.)

et qui soit, par exemple sans-attente (résistant aux pannes), c'est à dire qui termine même si $N - 1$ processeurs tombent en panne?

Exemple: Le consensus: les valeurs d'entrée sont une valeur quelconque dans un ensemble S , les valeurs de sortie doivent être toutes les mêmes et égales à une des valeurs d'entrée.

Résultat

Il n'existe pas de protocole pour le consensus sur une machine à mémoire partagée sans attente [Herlihy/Shavit+FLP etc.].



Application: Invariance de la Dimension

Theorem \mathbf{R}^n homeomorphe à $\mathbf{R}^m \Leftrightarrow m = n$.

(de même pour l'équivalence d'homotopie).

Preuve:

- $\mathbf{R}^n \cong \mathbf{R}^m \Rightarrow S^{n-1} \simeq \mathbf{R}^n \setminus \{P\} \cong \mathbf{R}^m \setminus \{Q\} \simeq S^{m-1}$.
- Donc: S^{n-1} et S^{m-1} ont des groupes d'homotopie isomorphes en toutes dimensions.
- Conclusion: $m = n$.

Algèbre: somme amalgamée de groupes

- Le groupe $G_1 * G_2$ engendré par l'union de deux groupes G_1 et G_2 est l'ensemble de mots $g_1^1 * g_2^1 * \cdots * g_1^n * g_2^n$, ; $g_1^j \in G_1, g_2^j \in G_2, n \in \mathbf{N}$.
- Soit H un autre groupe, et $i_j : H \rightarrow G_j$, ; $j = 1, 2$, deux homomorphismes. La somme amalgamée $G_1 *_H G_2$ de G_1 et G_2 au dessus de H est le groupe quotient de $G_1 * G_2$ par le sous-groupe normal engendré par tous les mots du type $i_1(h) * i_2(h^{-1})$.

Algèbre: somme amalgamée de groupes

- $G_1 *_H G_2$ est le **pushout** de:

$$\begin{array}{ccc} H & \xrightarrow{i_1} & G_1 \\ i_2 \downarrow & & \downarrow \\ G_2 & \longrightarrow & G_1 *_H G_2 \end{array}$$

Le théorème de Seifert-van Kampen

Soit $X = A_1 \cup A_2$ un espace topologique et deux sous-espaces connexes par arcs A_1 et A_2 . Soit $x_0 \in A_1 \cap A_2$, supposé connexe par arcs.

- $\pi_1(X; x_0) \cong \pi_1(A_1; x_0) *_{\pi_1(A_1 \cap A_2; x_0)} \pi_1(A_2; x_0)$.
- Interpretation: Pushout d'espaces \mapsto pushout de groupes fondamentaux:

$$\begin{array}{ccc} A_1 \cap A_2 & \longrightarrow & A_1 \\ \downarrow & & \downarrow \\ A_2 & \longrightarrow & X \end{array}$$

$$\begin{array}{ccc} \pi_1(A_1 \cap A_2; x_0) & \longrightarrow & \pi_1(A_1; x_0) \\ \downarrow & & \downarrow \\ \pi_1(A_2; x_0) & \longrightarrow & \pi_1(X; x_0) \end{array}$$

Application et avertissement...

- Le calcul de $\pi_1(S)$; S d'une surface (sphère, tore, bouteille de Klein etc.)
- Peut être utilisé pour prouver que deux surfaces *ne sont pas* homotopiquement équivalentes, et donc pas homéomorphes.
- Il n'y a pas de généralisation immédiate aux dimensions supérieures!
- En général, les groupes d'homotopie sont faciles à définir – mais difficiles ou impossibles à calculer!

Ensembles simpliciaux: les n -simplexes

Soit $\mathbf{v}_0, \dots, \mathbf{v}_n$ une liste de $n + 1$ points dans \mathbf{R}^k , ; $k > n$, qui sont *indépendants*, i.e., pas contenus dans un hyperplan de dimension n ;

- $\Delta_n = [\mathbf{v}_0, \dots, \mathbf{v}_n] = \{ \sum t_i \mathbf{v}_i \mid t_i \geq 0, \sum t_i = 1 \}$ – le n -simplexe engendré par \mathbf{v}_i .
- une *face*: un sous-simplexe $[\mathbf{v}_{i_0}, \dots, \mathbf{v}_{i_k}]$, ; $0 \leq i_0 \leq \dots \leq i_k \leq n$.

Ensembles simpliciaux: vue géométrique

- Collection de simplexes *disjoints*
- Où l'on *identifie* certaines faces par des fonctions *affines* *préservants l'ordre*
- Cela donne un *espace quotient* avec la topologie quotient

Ensembles simpliciaux: vue combinatoire

Un **ensemble simplicial** X_* est une famille d'ensembles $X_n, n \geq 0$ et de fonctions *faces* $d_i : X_n \rightarrow X_{n-1}, 0 \leq i \leq n$, et *dégénérescences* $s_j : X_n \rightarrow X_{n+1}, 0 \leq j \leq n$, satisfaisants les relations:

- $d_i d_j = d_{j-1} d_i$ for $i < j$;

- $s_i s_j = s_{j+1} s_i$ for $i \leq j$,

$$s_{j-1} d_i \quad \text{for } i < j,$$

- $d_i s_j = \text{id}$ for $i = j, i = j + 1$,

$$s_j d_{i-1} \quad \text{for } i > j + 1.$$

Homologie simpliciale 1

Soit X un ensemble simplicial.

- $\Delta_n(X) = \{ \sum n_i \Delta_i^n \mid ; n_i \in \mathbf{Z}, \Delta_i^n \in X_n \}$
le n ième **groupe de chaine**, groupe abélien libre.
- $\partial_n : \Delta_n(X) \rightarrow \Delta_{n-1}(X)$,
 $\partial_n([v_0, \dots, v_n]) = \sum_{i=0}^n (-1)^i [v_0, \dots, \hat{v}_i, \dots, v_n]$
ou $\partial_n(x) = \sum_{i=0}^n (-1)^i x_i(x)$, ; $x \in X^n$; -
l'**opérateur bord**.
- Lemme: $\partial_n \circ \partial_{n+1} = 0 : \Delta_{n+1}(X) \rightarrow \Delta_{n-1}(X)$.
- Les **cycles** = $\ker \partial_n \subseteq \partial_{n+1}(\Delta_{n+1}(X))$ (les **bords**) $\subseteq \Delta_n(X)$.

Homologie simpliciale 2

•

$$H_n(X) = \ker \partial_n / \text{im} \partial_{n+1}$$

est le n ième groupe d'homologie de X , défini comme groupe quotient.

- Une fonction simpliciale $f : X \rightarrow Y$ induit
 - un **morphisme de chaîne** $f_{\#} : \Delta_*(X) \rightarrow \Delta_*(Y)$ commutant avec les opérateurs bords ∂_* ,
 - et est donc un **homomorphisme de groupe** $f_* : H_*(X) \rightarrow H_*(Y)$, ; $* \geq 0$.
 - L'homologie simpliciale peut être étendue à un foncteur de la catégorie des paires d'ensembles simpliciaux vers les groupes abéliens satisfaisant les axiomes d'une théorie homologique.

Calculs homologiques simples

- Une **équivalence d'homotopie** $f : X \rightarrow Y$ induit un **isomorphisme** $f_* : H_*(X) \rightarrow H_*(Y)$, ; $* \geq 0$.
- X un espace **contractile** (équivalent homotopiquement à un point par exemple, une boule D^n , l'espace euclidien \mathbf{R}^n)
 $\Rightarrow H^*(X) = 0$, ; $* > 0$.

sphère $H_0(P) = G \Rightarrow H_i(S^n) = \begin{cases} G & i = 0, n \\ 0 & \text{else} \end{cases}$

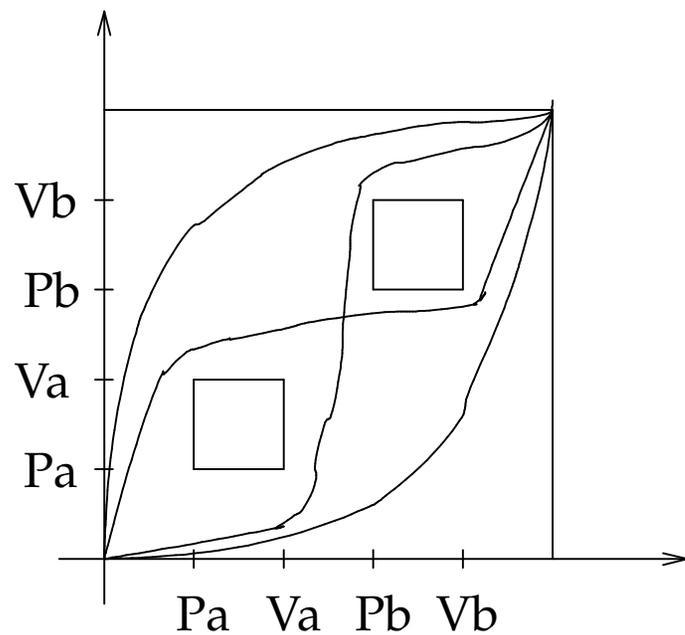
- Il n'y a aucune fonction continue $f : D^n \rightarrow S^{n-1}$ étendant l'identité sur la sphère \Rightarrow **théorème du point fixe de Brouwer**.

Liens entre homotopie et homologie

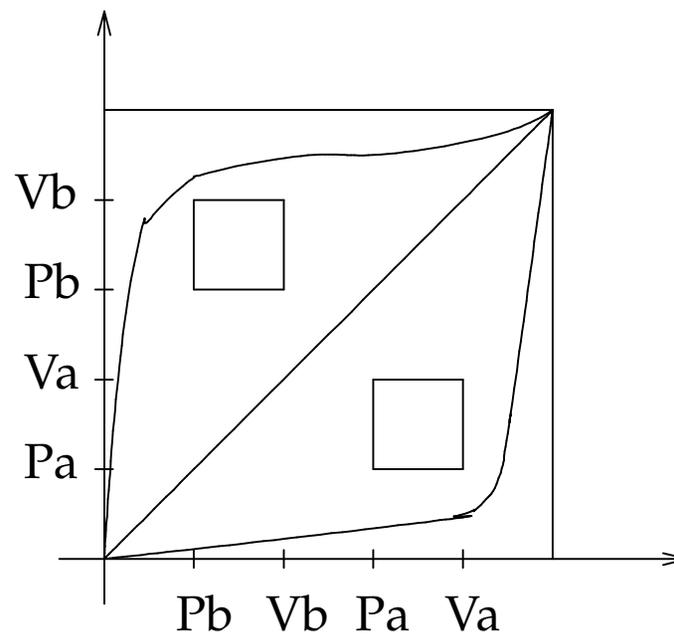
- L'homologie d'un espace (pas trop compliqué) peut être calculé (soit inductivement, soit directement) – bien plus accessible que les groupes d'homotopie!
- La connaissance de l'homologie/cohomologie permet souvent de résoudre des problèmes homotopiques
- **Théorème d'Hurewicz** Un espace **simplement connexe** X ($\pi_1(X) = 0$) est **k -connexe** ($\pi_i(X) = 0, ; i \leq k$) ssi $H_i(X) = 0, ; i \leq k$.

Première subtilité

$PaVaPbVb \mid PaVaPbVb:$



$PbVbPaVa \mid PaVaPbVb:$



La dihomotopie n'est pas l'homotopie "standard"

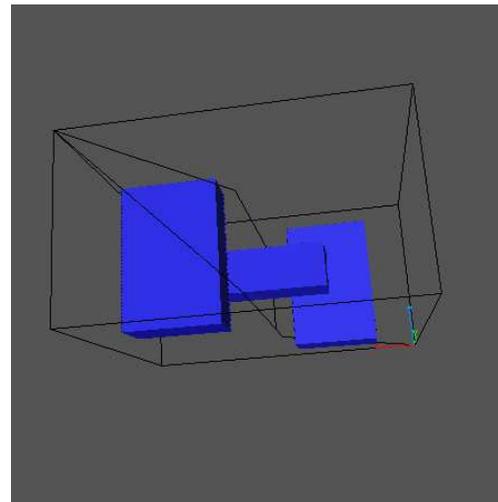
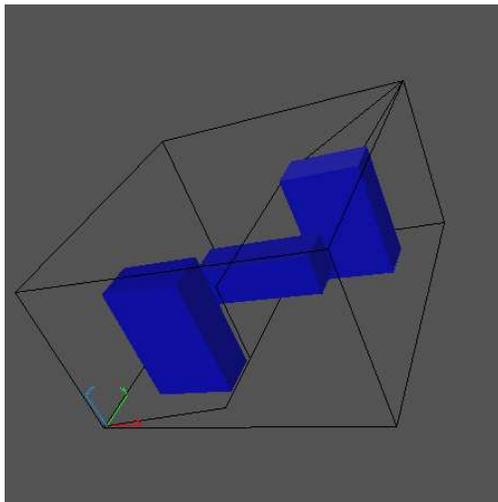
Deuxième subtilité...

#sem c 2

$A = Pa . Pc . Va . Pb . Vc . Vb$

$B = Pa . Va . Pc . Vc . Pb . Vb$

$C = Pc . Vc$



Dihomotopie \neq l'homotopie classique relativement aux extrémités.

Catégorie fondamentale

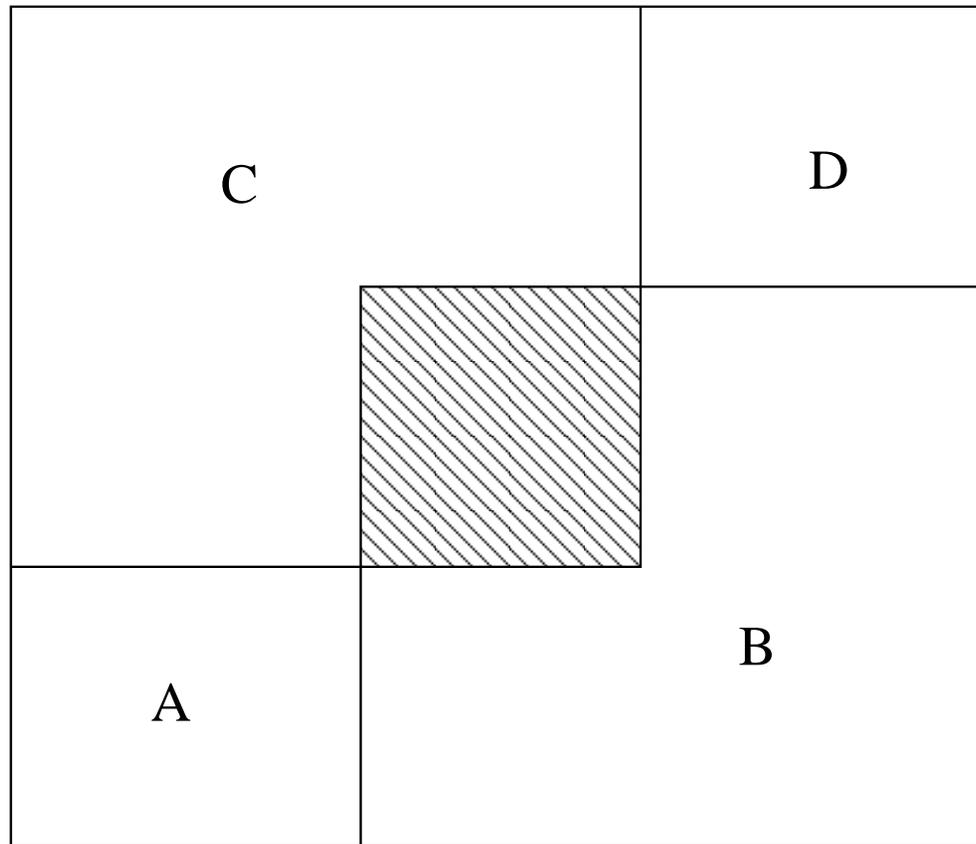
Catégorie donnée par:

- objets: tous les points de l'espace topologique partiellement ordonné
- morphismes: $f : x \rightarrow y$ est une classe d'équivalence de chemins de x vers y modulo (di)homotopie

Remarques:

- En quelque sorte: "algèbre des chemins essentiels".
- Bien lourde à manipuler et à calculer...
- Générateurs et relations? \implies *catégorie de composantes*

Exemple



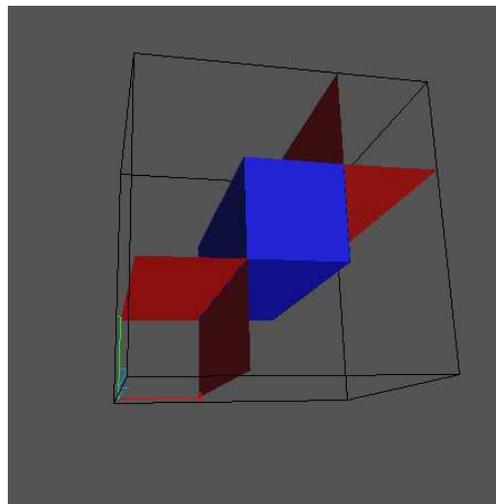
Classes d'homotopie

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>A</i>	1	1	1	2
<i>B</i>	0	1	0	1
<i>C</i>	0	0	1	1
<i>D</i>	0	0	0	1

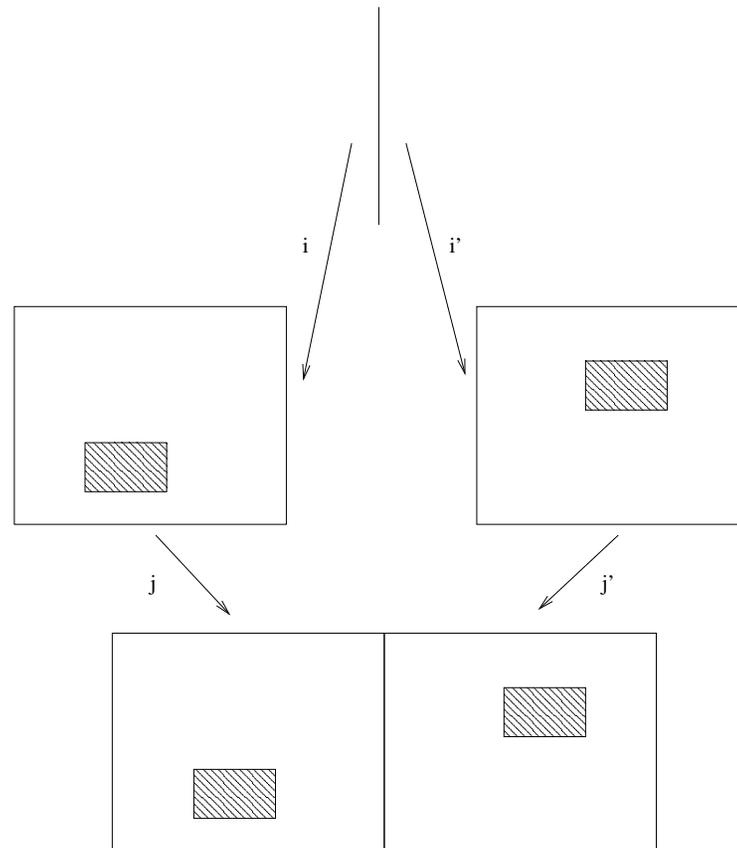
Calcul? produit [composition parallèle “indépendante”]

Comme dans le cas classique:

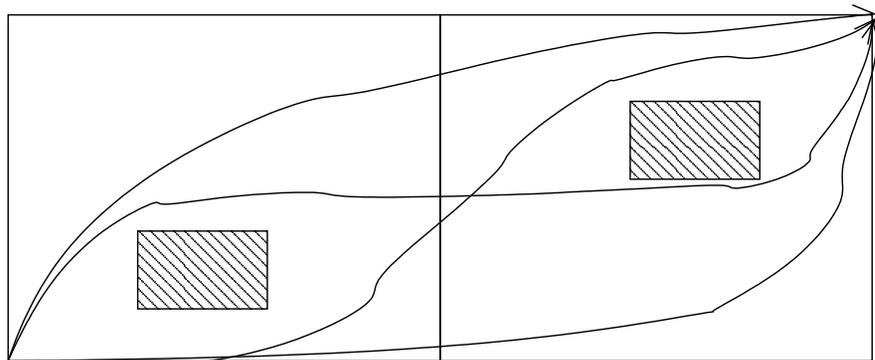
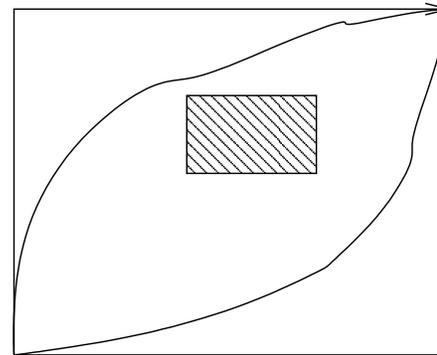
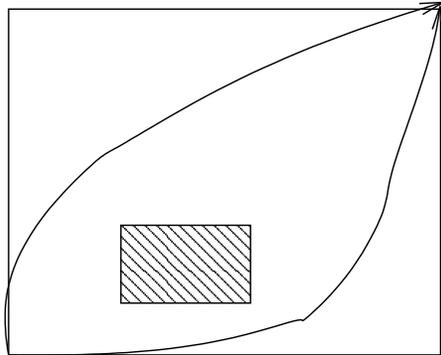
Il y a une équivalence de catégorie entre $\vec{\pi}_1 (X \times Y)$
et $\vec{\pi}_1 (X) \times \vec{\pi}_1 (Y)$



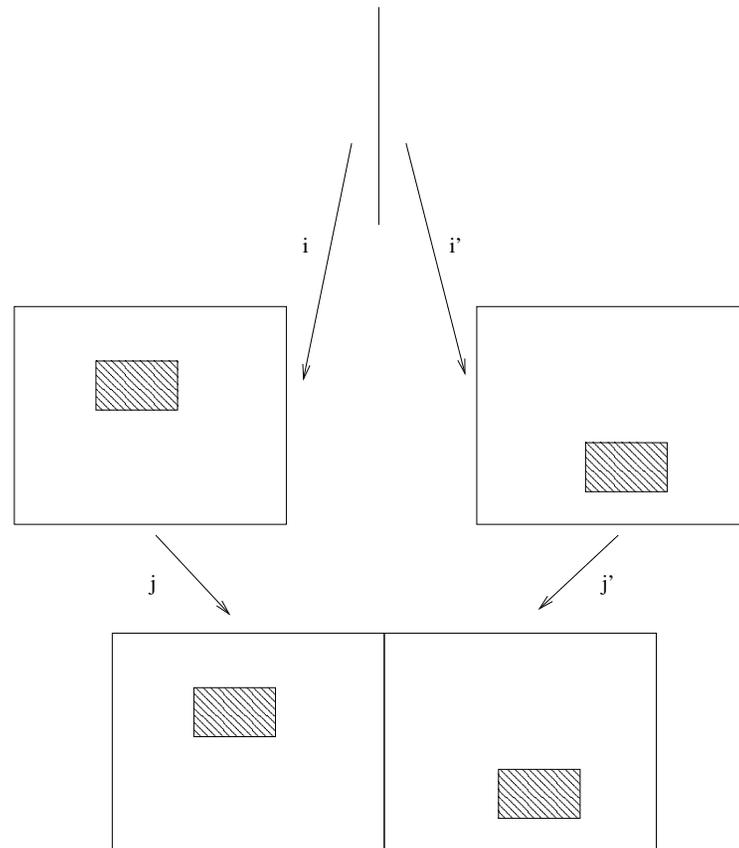
Compositionnalité: exemple 1



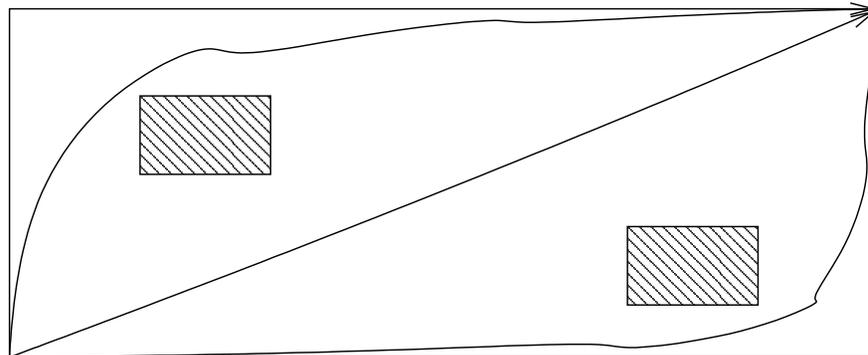
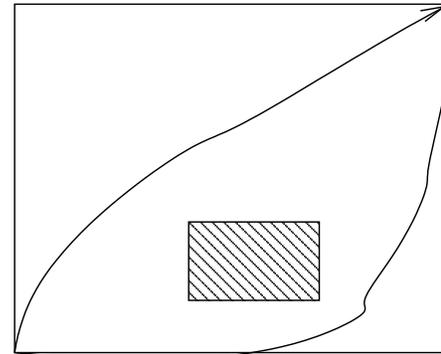
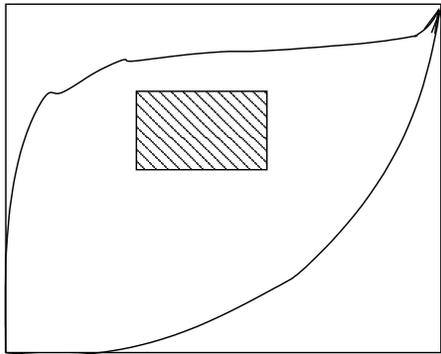
On devrait trouver:



Compositionnalité: exemple 2



On devrait trouver:



Théorème de Seifert/van Kampen [compositionnalité]

[cf. également M. Grandis]

On suppose $X = \overset{\circ}{X}_1 \cup \overset{\circ}{X}_2$,

Tous les chemins **continus** $\overset{\circ}{X}_1 \cap \overset{\circ}{X}_2$ sont des concaténations d'un nombre fini de dichemins et de dichemins inverses. Soient $j_1 : X_1 \cap X_2 \rightarrow X_1$ (respectivement $j_2 : X_1 \cap X_2 \rightarrow X_2$) et $i_1 : X_1 \rightarrow X$ (respectivement $i_2 : X_2 \rightarrow X$) les inclusions canoniques.

$$\begin{array}{ccc} \vec{\pi}_1 (X_1 \cap X_2) & \xrightarrow{\vec{\pi}_1(j_1)} & \vec{\pi}_1 (X_1) \\ \downarrow \vec{\pi}_1(j_2) & & \vec{\pi}_1(i_1) \downarrow \\ \vec{\pi}_1 (X_2) & \xrightarrow{\vec{\pi}_1(i_2)} & \vec{\pi}_1 (X) \end{array}$$

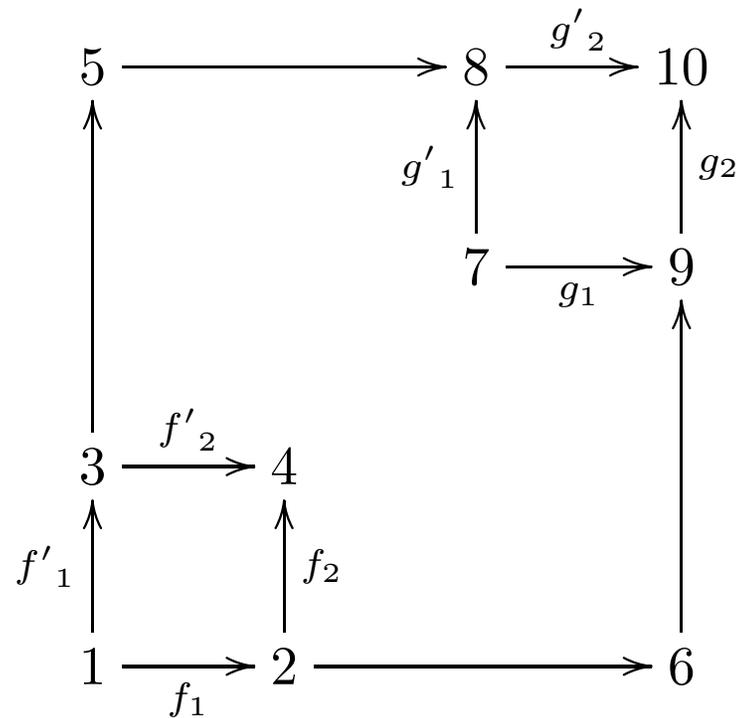
est co-cartésien dans la catégorie des catégories.

Catégorie de composantes

- Représentant le **plus efficace** des régions d'états, permettant un parcours des chemins d'exécution réduit au mieux
- Est une catégorie, c'est à dire une sorte d'**algèbre de chemins** (que l'on représente par "générateurs" et "relations")
- Est un des moyens "géométriques" imaginés pour **réduire l'espace d'états**

Cas simple

			8	10
5			7	9
3	4			
1	2		6	



et les relations $g'_2 \circ g'_1 = g_2 \circ g_1$
and $f'_2 \circ f'_1 = f_2 \circ f_1$

Définition algébrique

Les chemins que l'on souhaiterait pouvoir inverser sont ceux de Σ tels que:

- Σ contient tous les isomorphismes
- Tous les σ dans Σ sont epis et monos dans $\vec{\pi}_1(X)$, c.a.d. quelque soient u et v dans $\vec{\pi}_1(X)$, si on a un diagramme commutatif (ou son dual):

$$\begin{array}{c} \xrightarrow{\sigma} \quad \begin{array}{c} \xrightarrow{u} \\ \xleftarrow{v} \end{array} \end{array}$$

alors $u = v$.

Définition algébrique (suite)

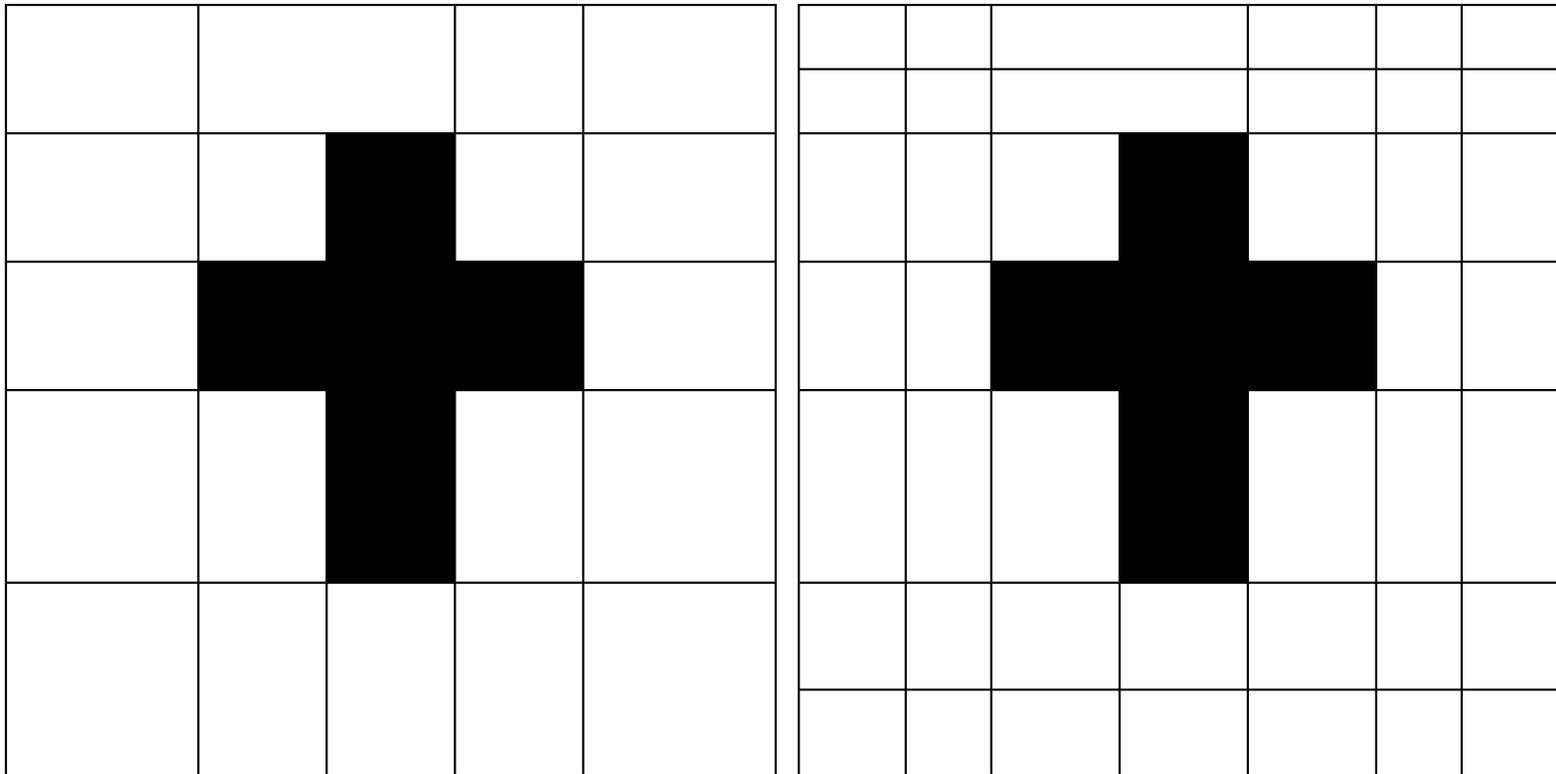
- S'il existe $u : \beta \rightarrow \gamma$ dans $\vec{\pi}_1(X)$, alors pour tout $\sigma : \alpha \rightarrow \beta$ de Σ , et tout $f : \alpha \rightarrow \gamma$ dans $\vec{\pi}_1(X)$, f se factorise à travers σ , c.a.d. qu'il existe $h : \beta \rightarrow \gamma$ tel que le diagramme suivant commute:

$$\begin{array}{ccc} & & \nearrow f \\ & & \uparrow h \\ \xrightarrow{\sigma} & & \end{array}$$

- Σ est stable par somme amalgamée et produit fibré (avec n'importe quel morphisme de $\vec{\pi}_1(X)$)

Propriétés importantes (APCS)

- Il existe de tels Σ ...
- Il en existe un, maximal (la rétraction de l'espace d'états optimale)

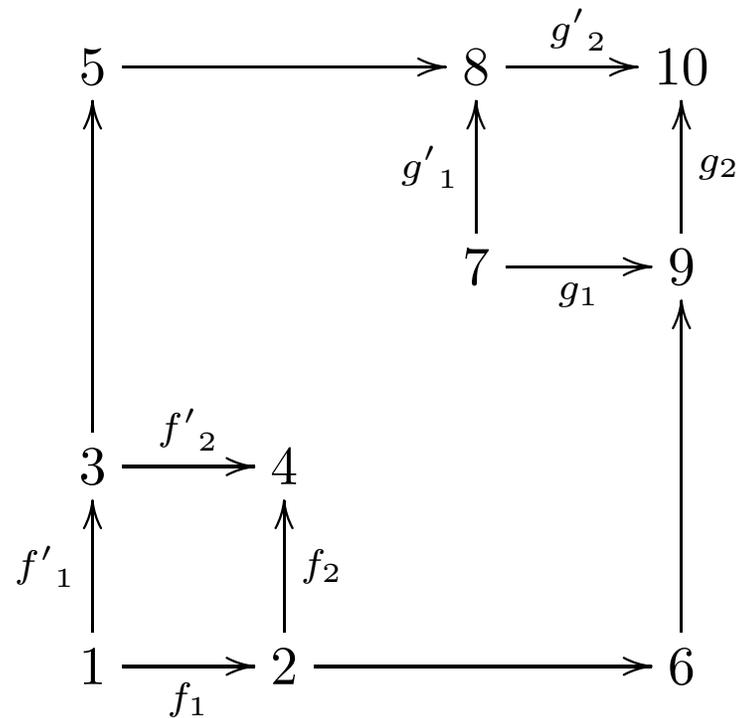
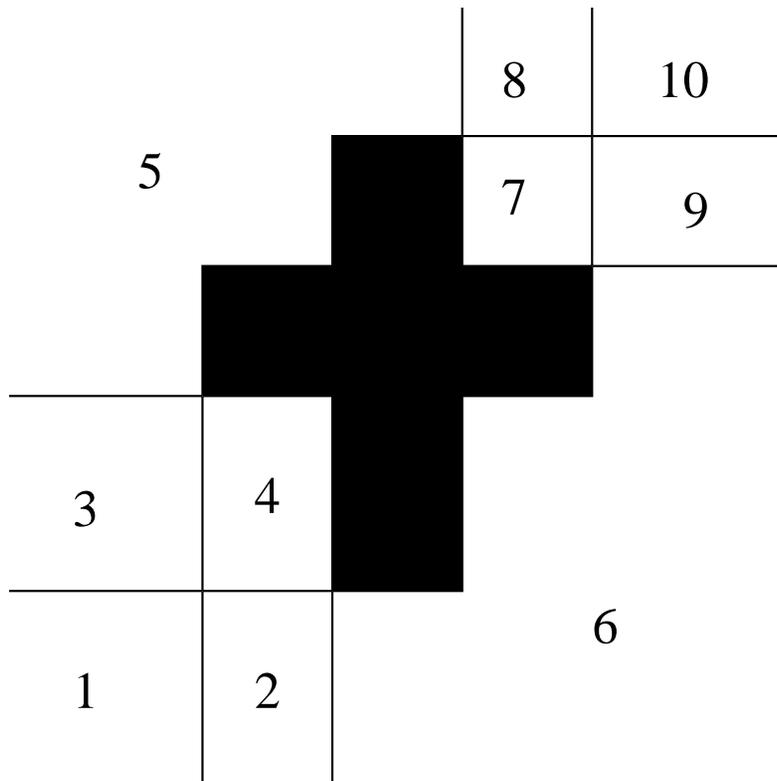


Propriétés importantes (APCS)

Propriété de relèvement, ou dit de façon plus ou moins simple:

- Tout morphisme f de la catégorie fondamentale a un représentant canonique $Q_{\Sigma}(f)$ dans la catégorie de composantes
- Inversement, pour tout morphisme g de la catégorie de composantes, on peut choisir un point de départ x (resp. d'arrivée y) dans la catégorie fondamentale, et il existe alors un morphisme f de x vers un y tel que $Q_{\Sigma}(f) = g$.

Catégorie de composantes



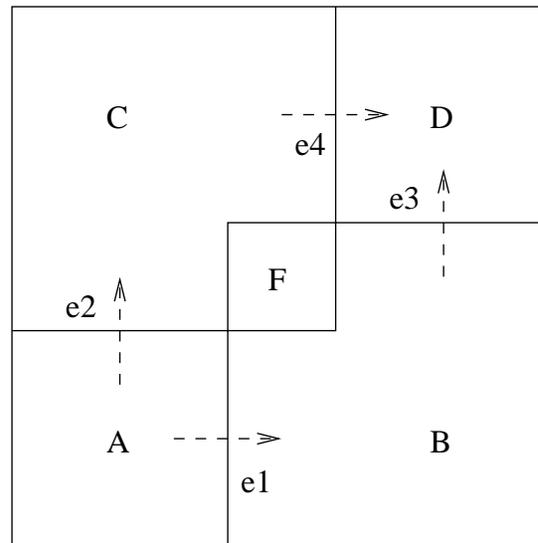
et les relations $g'_2 \circ g'_1 = g_2 \circ g_1$
and $f'_2 \circ f'_1 = f_2 \circ f_1$

Comment résoudre le problème...

... de la caractérisation des ordonnancements (ou traces essentielles)?

- Par l'application de théorèmes généraux (van Kampen, voir E. Haucourt, suites exactes etc.); moyen/long terme.
- Par algorithme particulier (dans le cas de \mathbb{R}^n): implémenté à l'heure actuelle, avec des améliorations algorithmiques à court et moyen terme.

La catégorie de composantes, géométriquement



Les composantes A , B , C and D correspondent aux carrés séparés par les lignes horizontales et verticales depuis les points min et max de la région interdite F .

La catégorie de composantes, géométriquement

- Le “morphisme unique” de A vers B (e_1) est en fait l’“orthogonal” de la ligne verticale venant du point min de F .
- On identifie e_1 avec la variété linéaire de codimension 1 (qui est ici le segment vertical)
- De façon similaire, e_2 est identifié avec la ligne horizontale à gauche du point min de F etc.
- Il n’y a pas de variété intéressante de codimension 2 ici, donc pas de “relations” entre morphismes.

On creuse un trou

Cela correspond ici au programme:

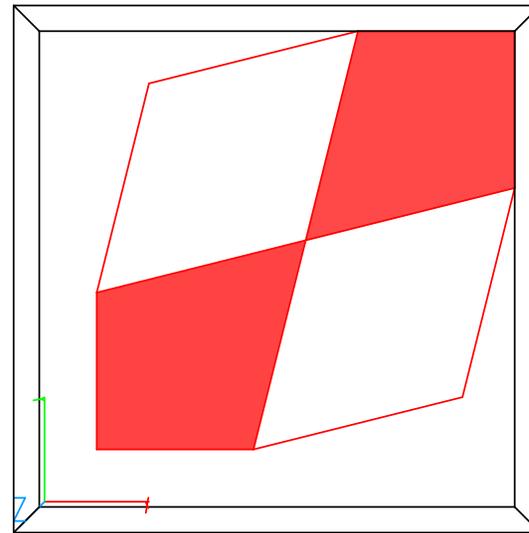
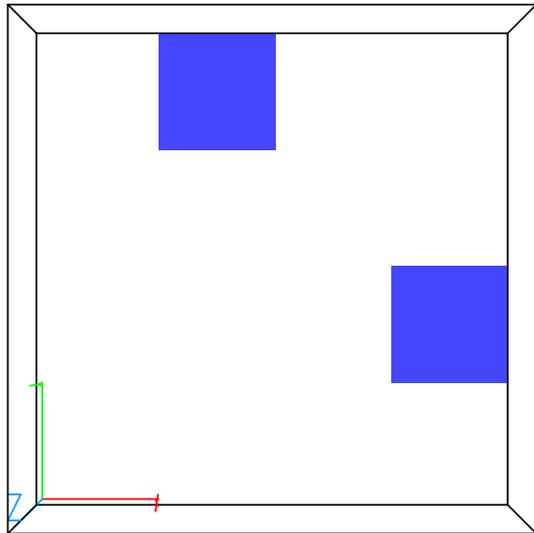
$$A = Pa \cdot Va \cdot Pb \cdot Vb$$

$$B = Pb \cdot Vb \cdot Pa \cdot Va$$

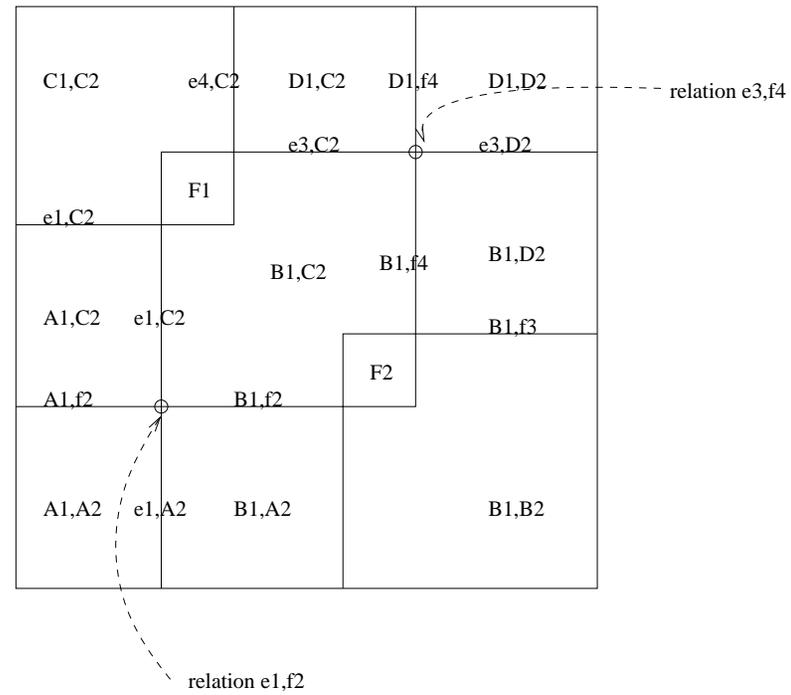
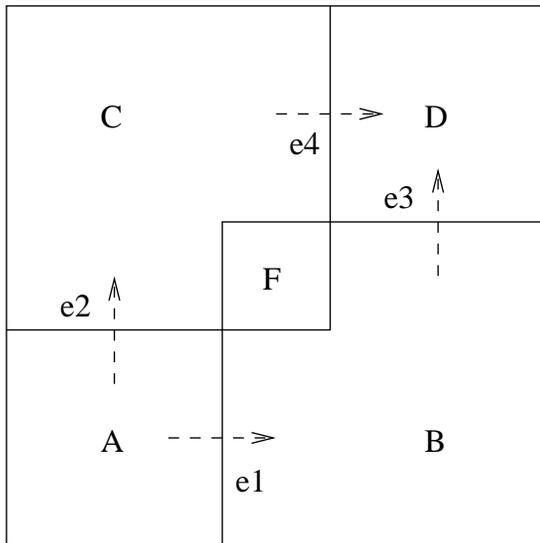
$$\text{PROG} = A \mid B$$

et la catégorie de composantes est:

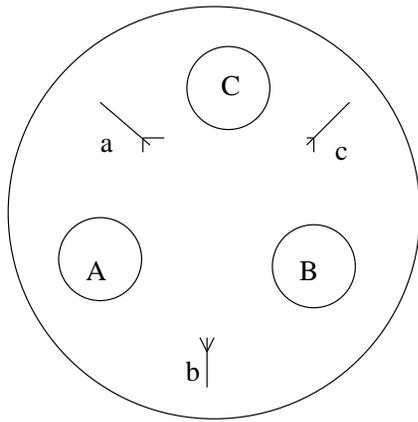
Catégorie de composantes



Formule inductive



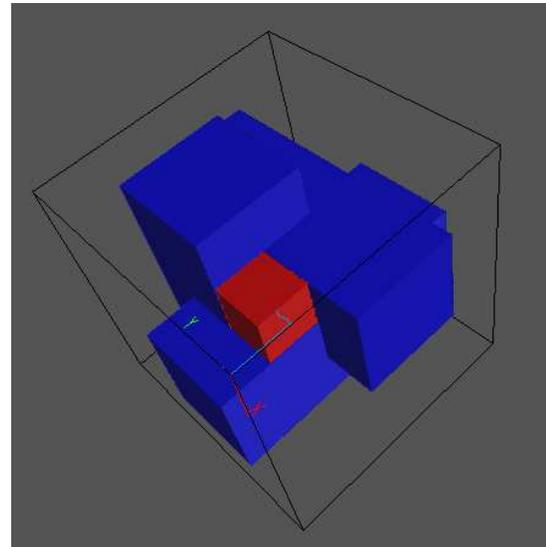
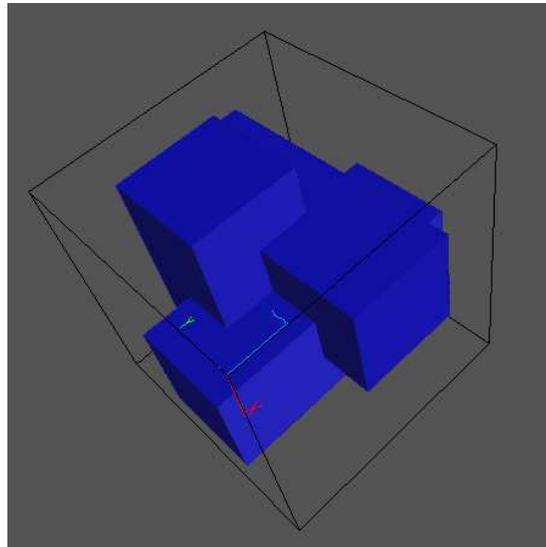
Les 3 philosophes



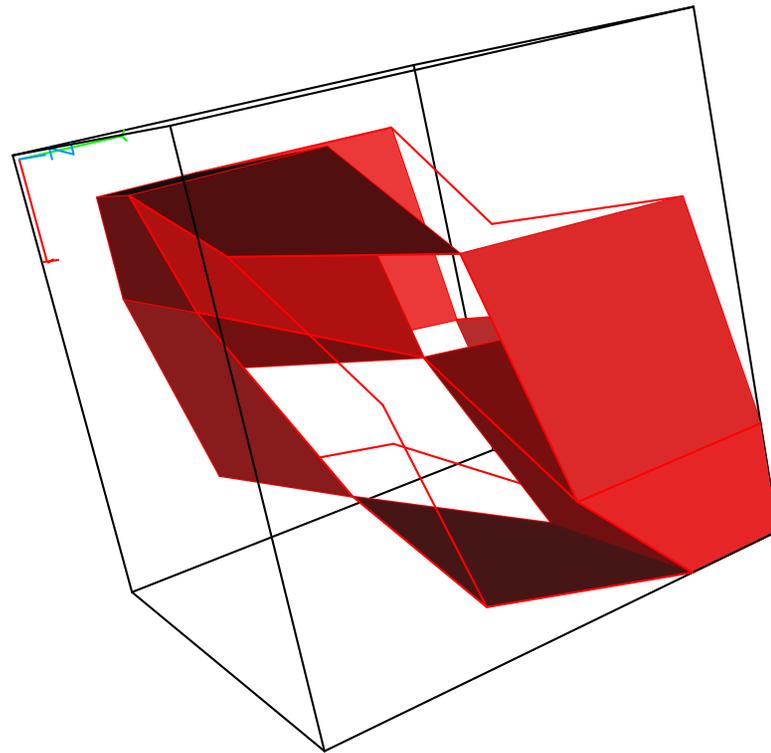
A=Pa . Pb . Va . Vb

B=Pb . Pc . Vb . Vc

C=Pc . Pa . Vc . Va



Catégorie de composantes

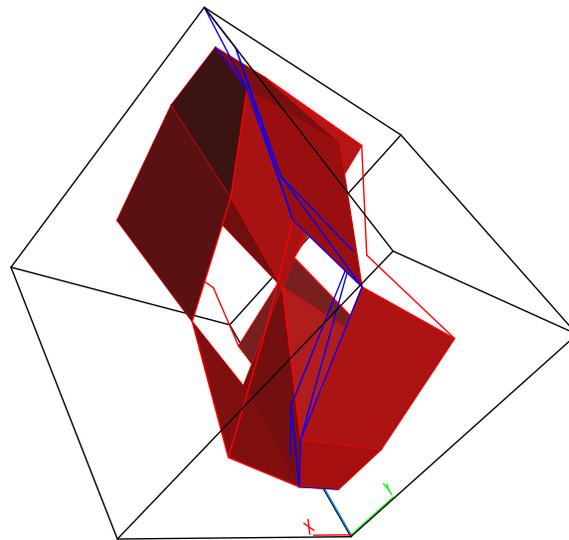


Quelques chiffres...

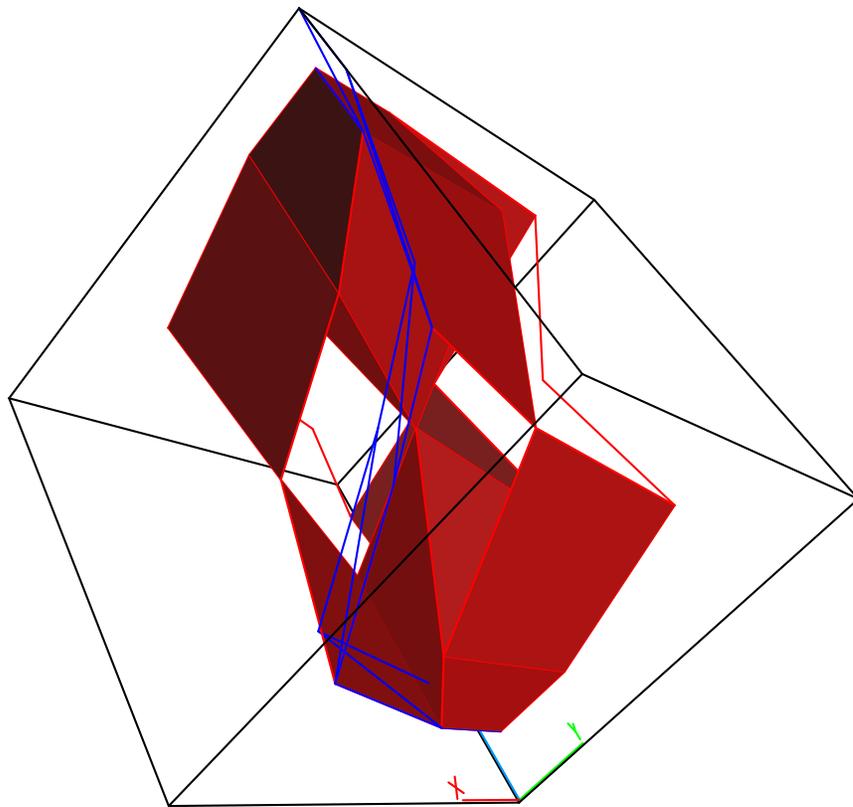
- new3phil.pv: (0.05s) Objets: 27, Morphismes: 48, Relations: 18
- new4phil.pv: (0.07s) Objets: 85, Morphismes: 200, Relations: 132
- new7phil.pv: 4s; 42 Mo; (environ un million de transitions dans un modèle standard) Objets: 2467, Morphismes: 10094, Relations: 15484
- new8phil.pv: 21s; 100Mo; (environ 10 millions de transitions dans un modèle standard par entrelacement) Objets: 7533, Morphismes: 35216, Relations: 64312
- new9phil.pv: 106s, 319Mo Objets: 22995, Morphismes: 120924, Relations: 256158 pour $9!=362880$ chemins maximaux...

Exemple

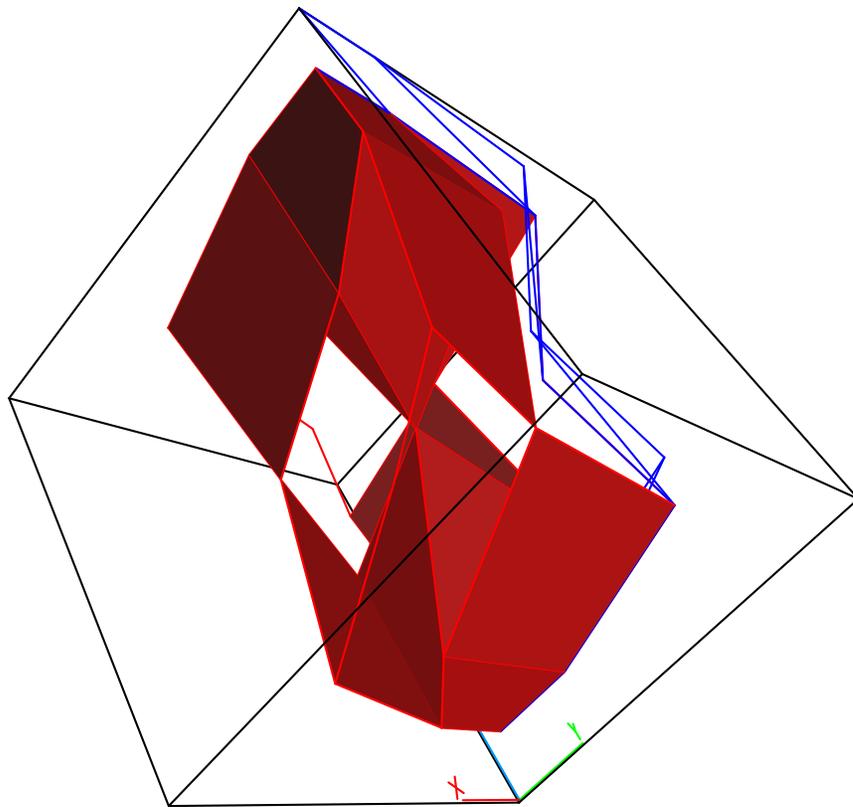
Pour le cas des trois philosophes, on trouve $6=3!$ chemins maximaux (il s'agit de toutes les séquentialisations possibles en fait):



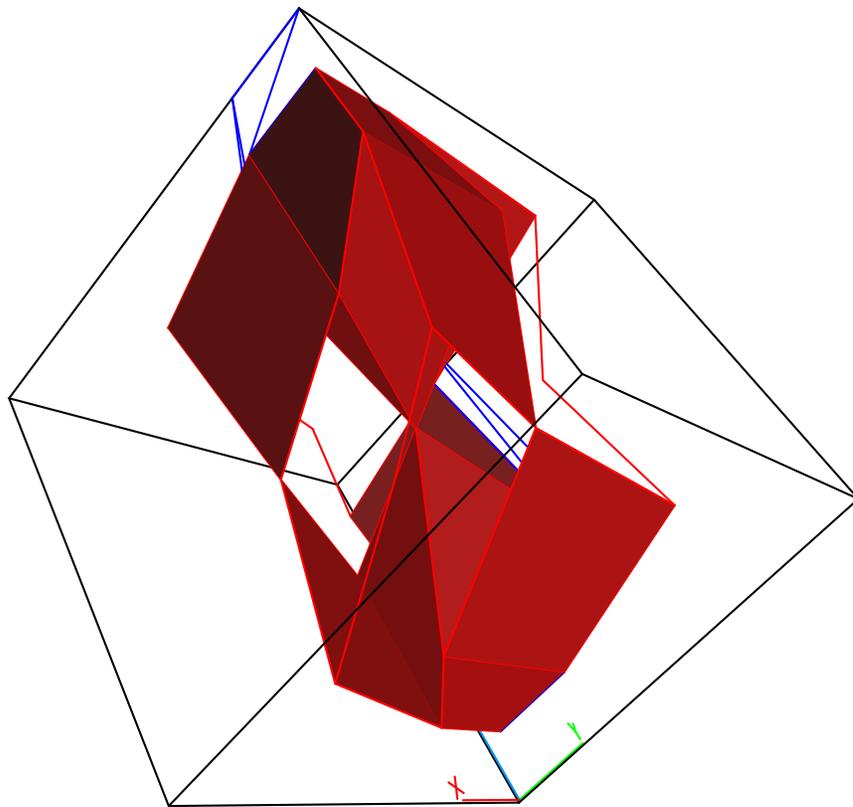
Chemin 2



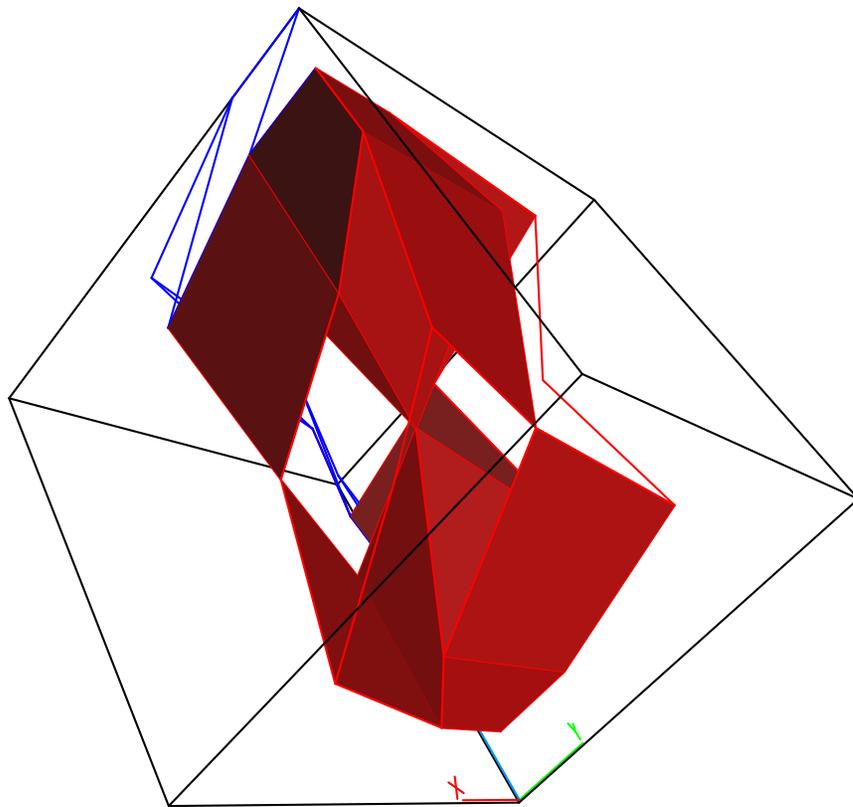
Chemin 3



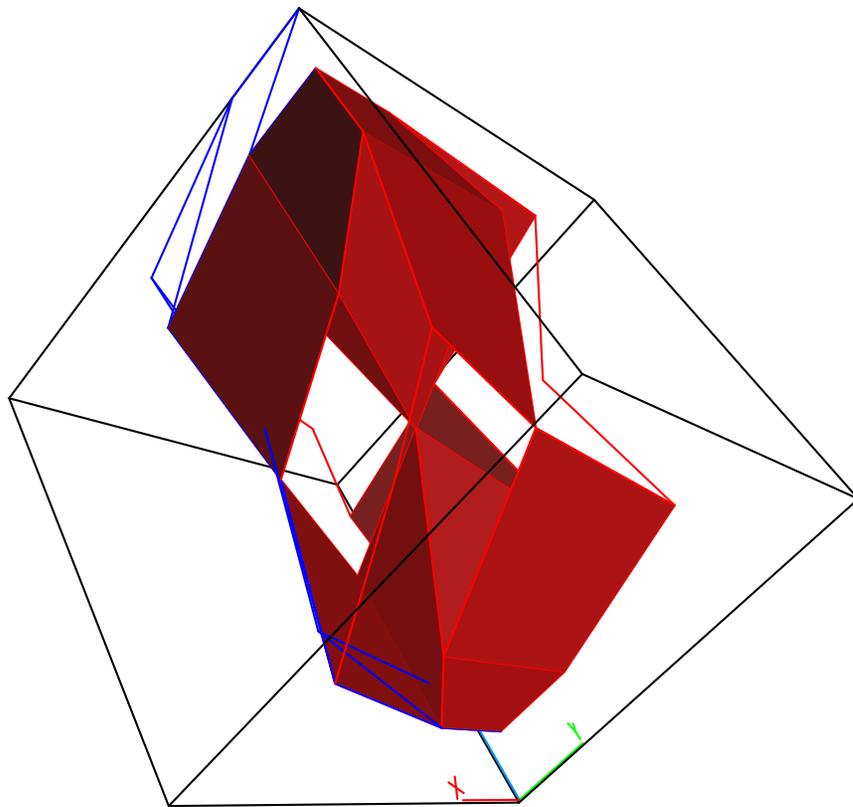
Chemin 4



Chemin 5



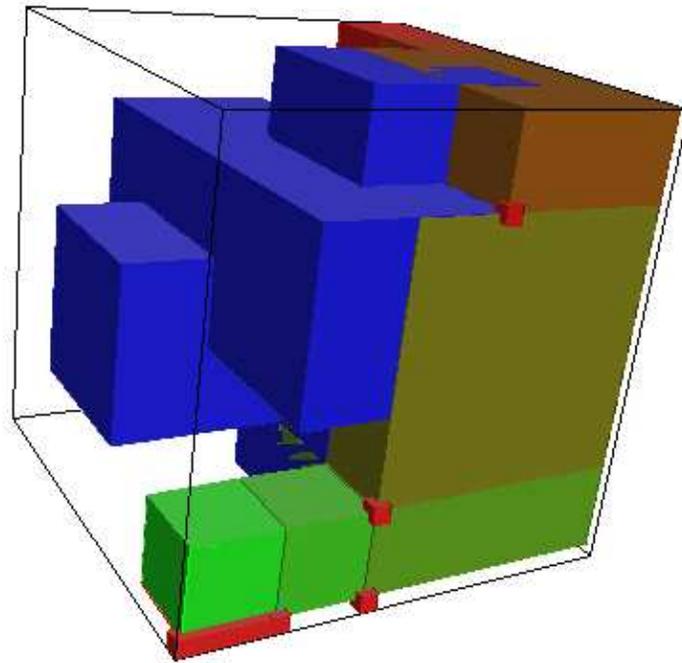
Chemin 6



Représentation syntaxique d'un chemin

- D'un chemin continu, on veut repasser à un chemin "discret"
- Ce chemin "discret" est en fait un entrelacement "correspondant" à cette exécution idéalisée
- Que l'on peut ensuite analyser avec un *analyseur séquentiel*

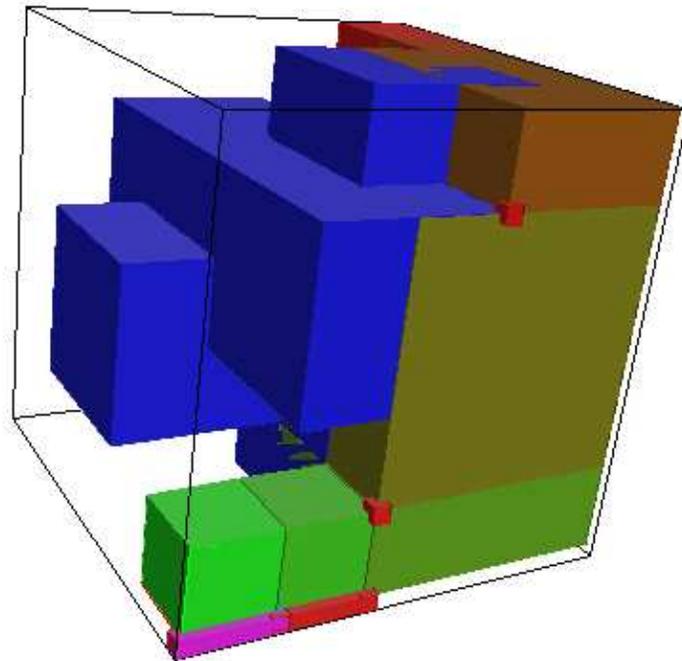
Discrétisation de chemins



0 | 0 | $P(c)$

Dans le contexte #sem c 1, #sem b 1, et #sem a 1.

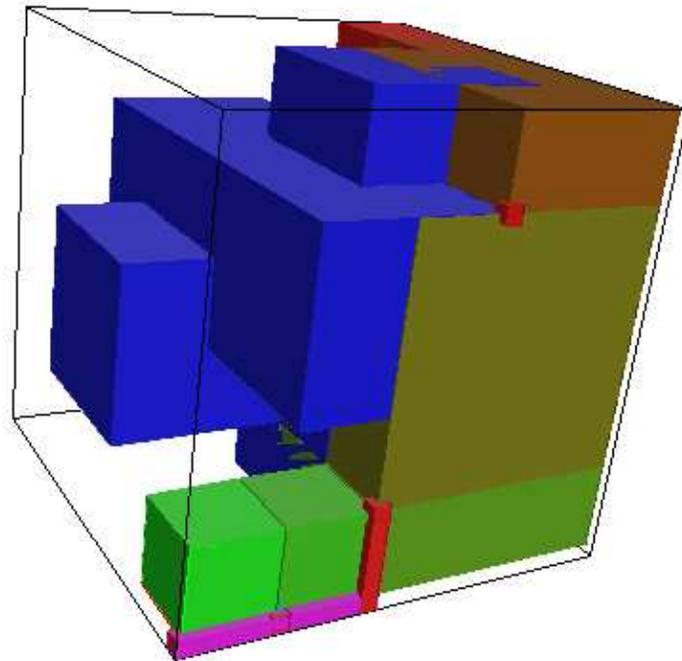
Discrétisation de chemins



0 | 0 | $P(a)$

Dans le contexte #sem c 0, #sem b 1, #sem a 1.

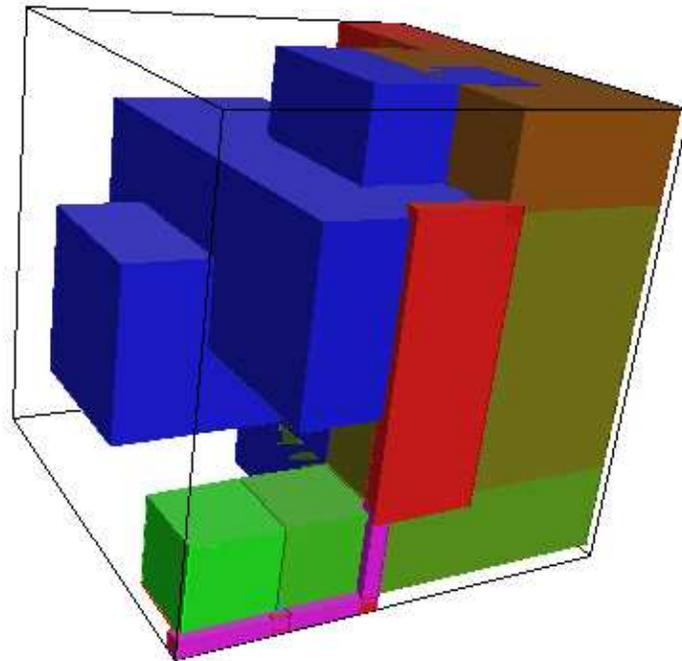
Discrétisation de chemins



0 | P(b) | 0

Dans le contexte #sem c 0, #sem b 1, #sem a 0.

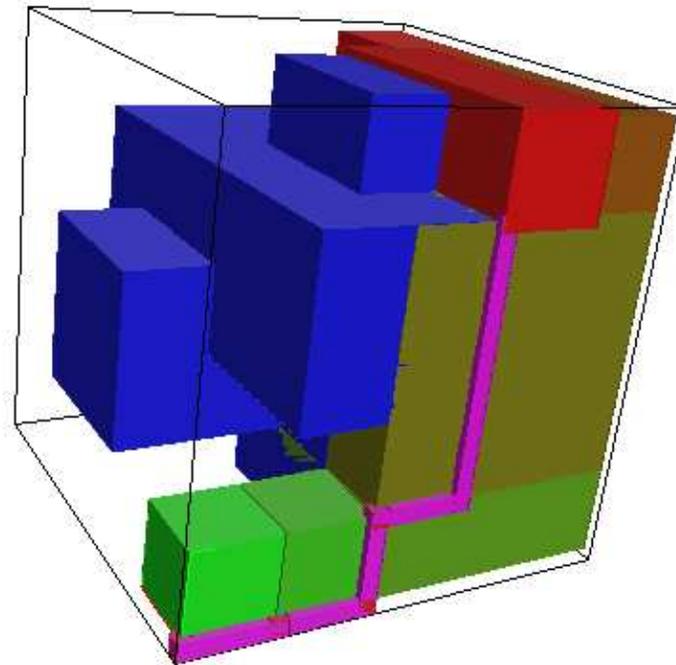
Discrétisation de chemins



0 | P(c).V(b).V(c) | V(c)

Dans le contexte #sem c 0, #sem b 0, #sem a 0.

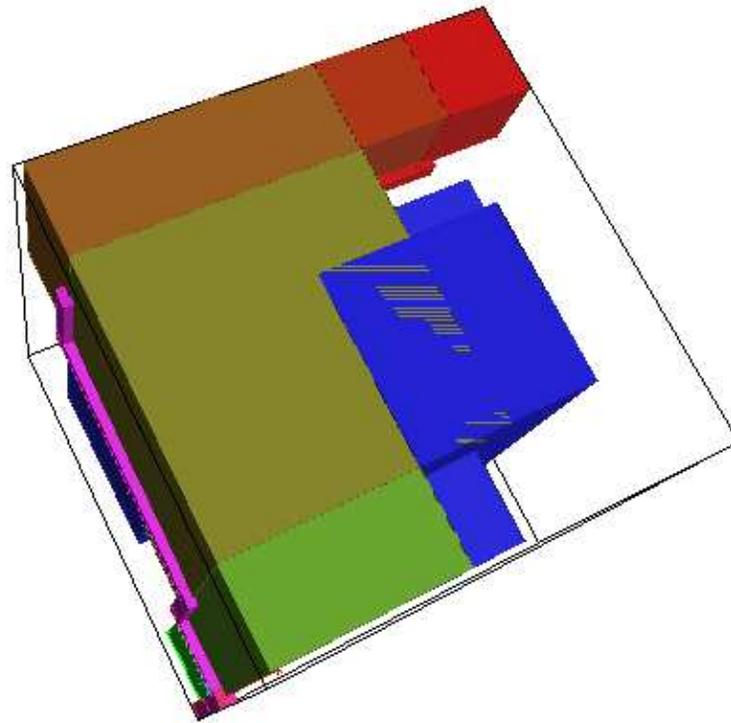
Discrétisation de chemins



$P(a) . P(b) . V(a) \mid 0 \mid V(a)$

Dans le contexte #sem c 1, #sem b 1, #sem a 0.

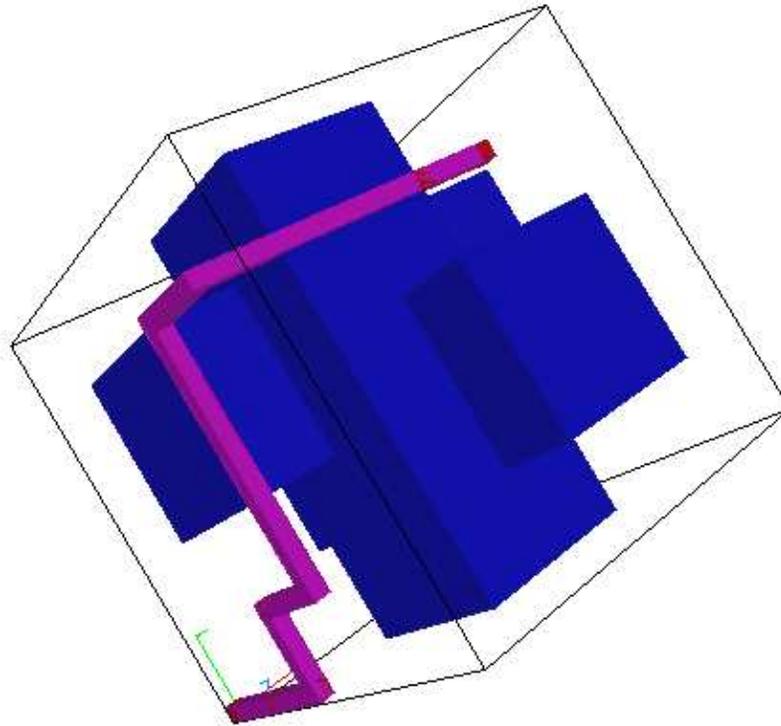
Discrétisation de chemins



$v(b) \mid 0 \mid 0$

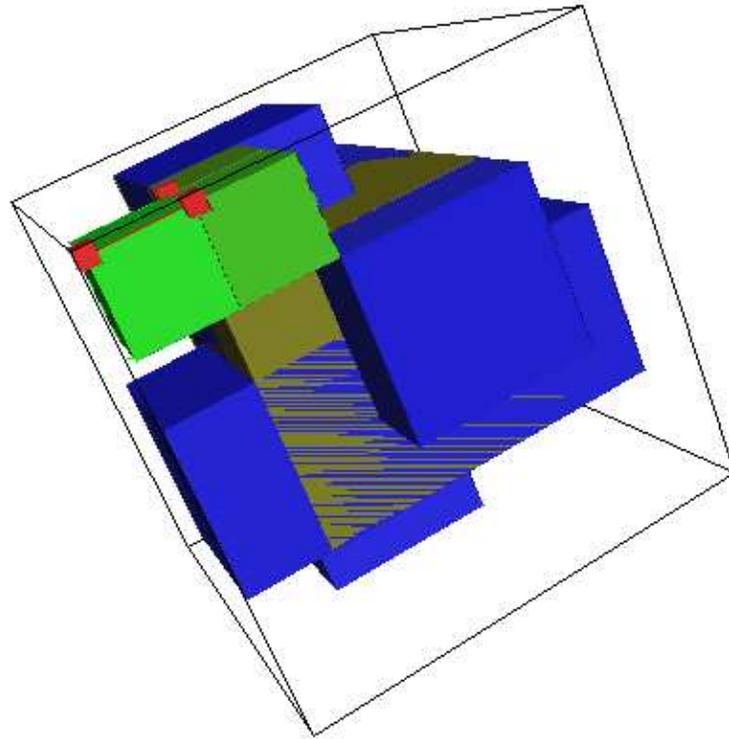
Dans le contexte #sem c 1, #sem b 0, #sem a 1.

Discrétisation de chemins



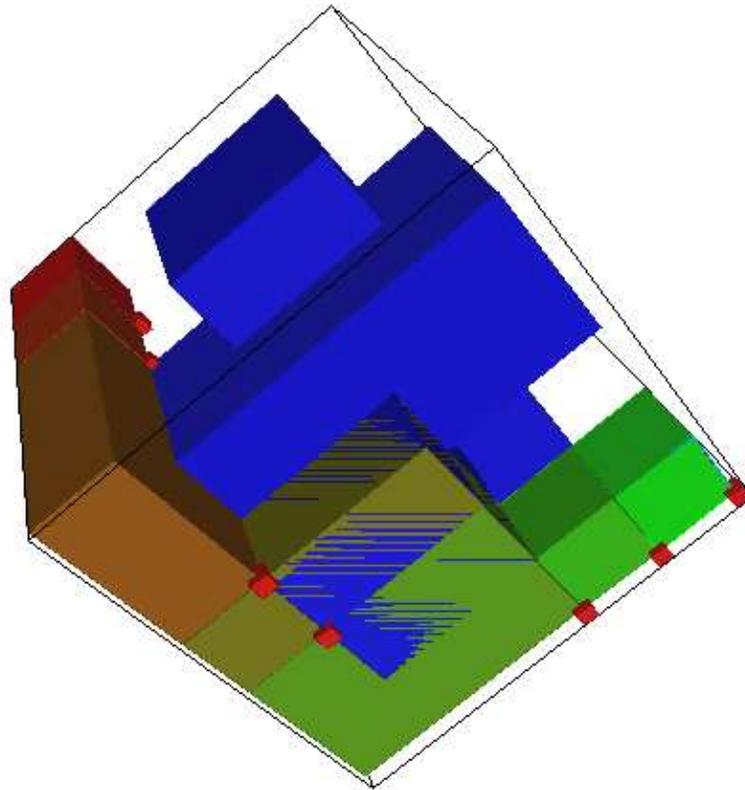
$\{3\}P(c) ; \{3\}P(a) ; \{2\}P(b) ; \{3\}V(c) ; \{2\}P(c) ; \{2\}V(b) ;$
 $\{2\}V(c) ; \{3\}V(a) ; \{1\}P(a) ; \{1\}P(b) ; \{1\}V(a) ; \{1\}V(b) ;$

Discrétisation de chemins (2 - deadlock)



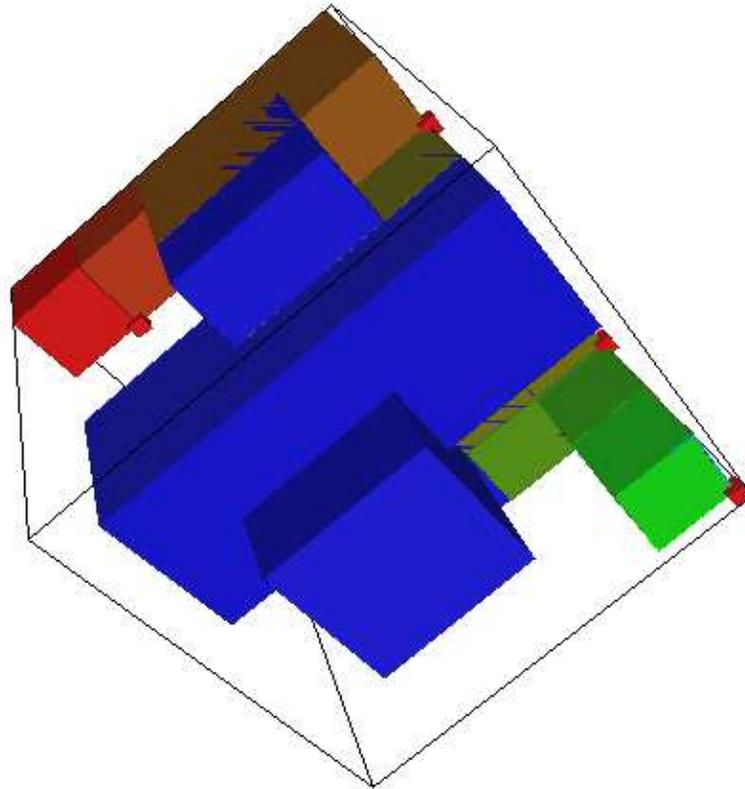
$\{1\}P(a); \{3\}P(c); \{2\}P(b);$

Discrétisation de chemins (3)



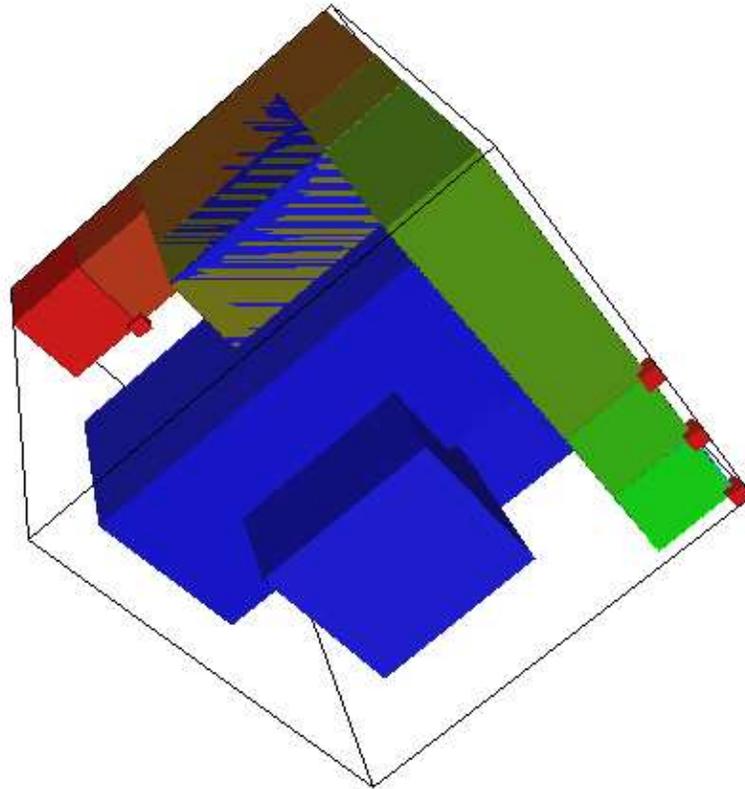
$\{2\}P(b) ; \{2\}P(c) ; \{2\}V(b) ; \{2\}V(c) ; \{3\}P(c) ; \{3\}P(a) ;$
 $\{3\}V(c) ; \{3\}V(a) ; \{1\}P(a) ; \{1\}P(b) ; \{1\}V(a) ; \{1\}V(b) ;$

Discrétisation de chemins (4)



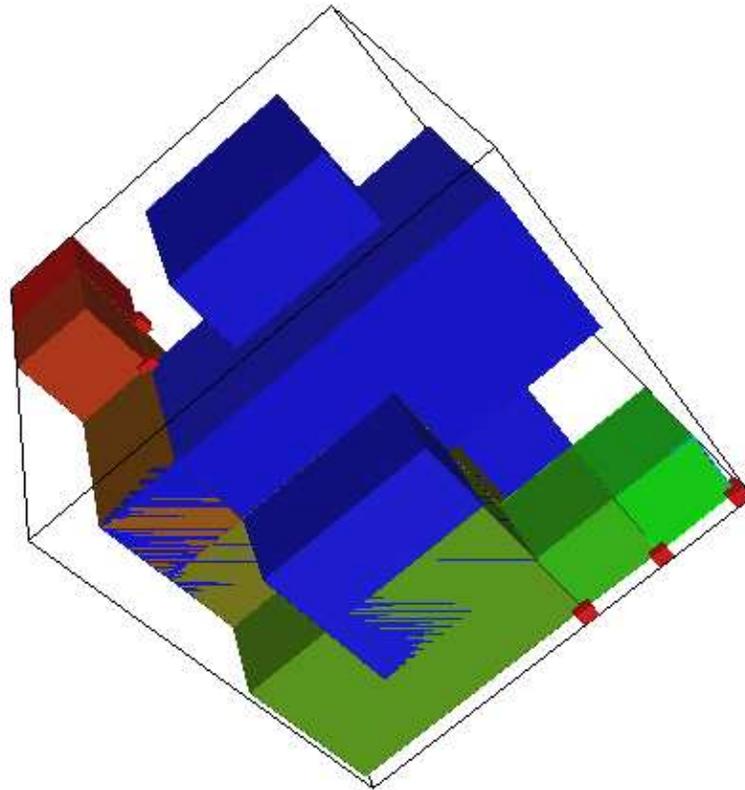
$\{1\}P(a); \{1\}P(b); \{3\}P(c); \{1\}V(a); \{3\}P(a); \{3\}V(c);$
 $\{3\}V(a); \{1\}V(b); \{2\}P(b); \{2\}P(c); \{2\}V(b); \{2\}V(c);$

Discrétisation de chemins (5)



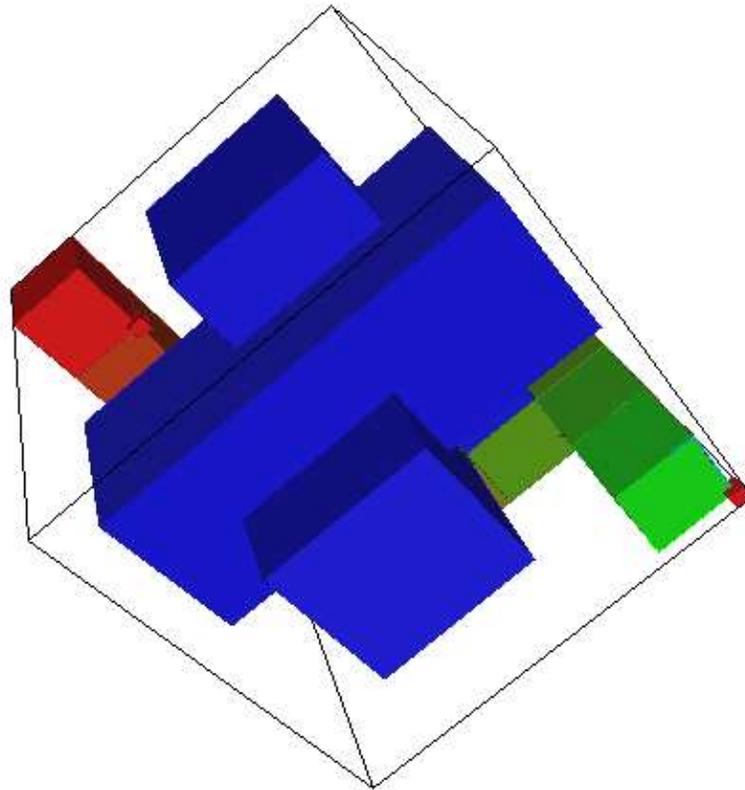
$\{3\}P(c); \{3\}P(a); \{3\}V(c); \{3\}V(a); \{1\}P(a); \{1\}P(b);$
 $\{1\}V(a); \{1\}V(b); \{2\}P(b); \{2\}P(c); \{2\}V(b); \{2\}V(c);$

Discrétisation de chemins (6)



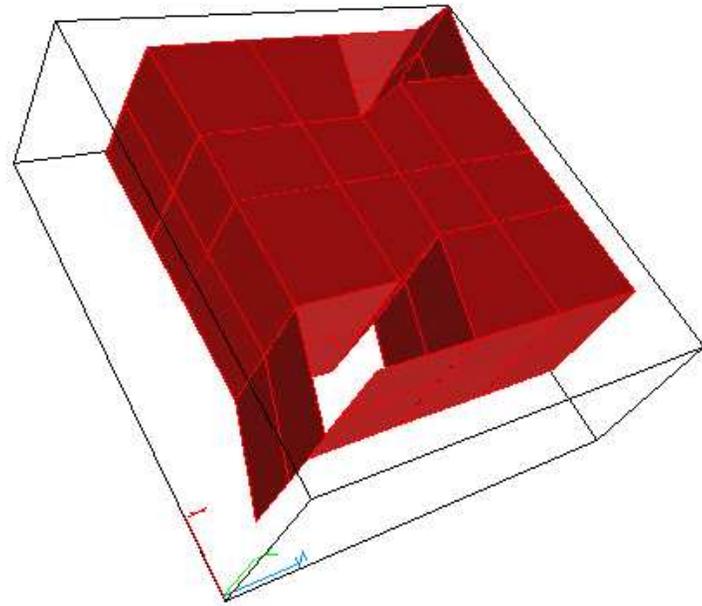
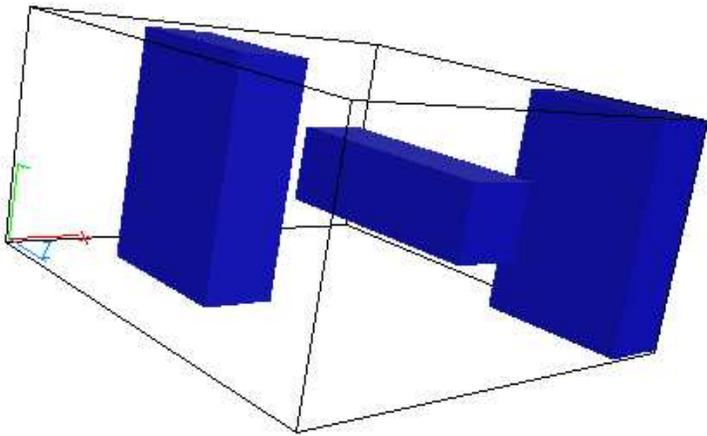
$\{2\}P(b); \{2\}P(c); \{1\}P(a); \{2\}V(b); \{1\}P(b); \{1\}V(a);$
 $\{2\}V(c); \{3\}P(c); \{3\}P(a); \{3\}V(c); \{3\}V(a); \{1\}V(b);$

Discrétisation de chemins (7)

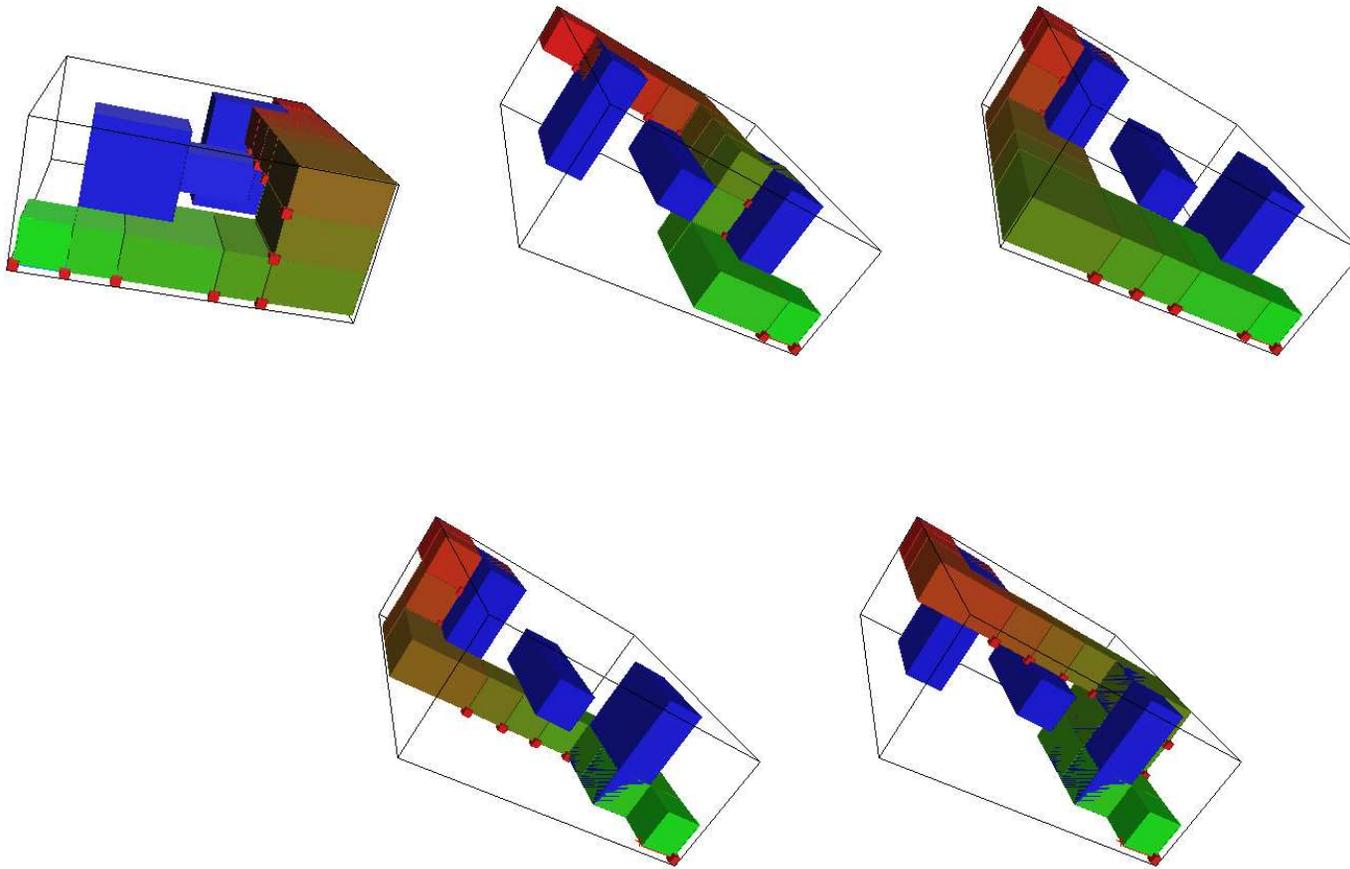


$\{1\}P(a); \{1\}P(b); \{1\}V(a); \{1\}V(b); \{2\}P(b); \{2\}P(c);$
 $\{2\}V(b); \{2\}V(c); \{3\}P(c); \{3\}P(a); \{3\}V(c); \{3\}V(a);$

Catégorie de composantes



Discrétisation de chemins



Améliorations algorithmiques

- Calcul des morphismes dans la catégorie des composantes:
 - Par calcul du premier groupe d'homologie
 - (ou plus simple) approximation des relations (type ensembles persistents)
 - (plus long terme) simplification du retract en considérant aussi les $\vec{\pi}_n$ ["résolution homotopique", ou "higher-order syzygies"]
- Plus toutes améliorations classiques: parcours à la volée, compression de l'espace d'états etc.