Algorithms for computing fundamental categories, and applications to the static analysis of concurrent programs

Eric Goubault

CEA/Saclay

Eric.Goubault@cea.fr

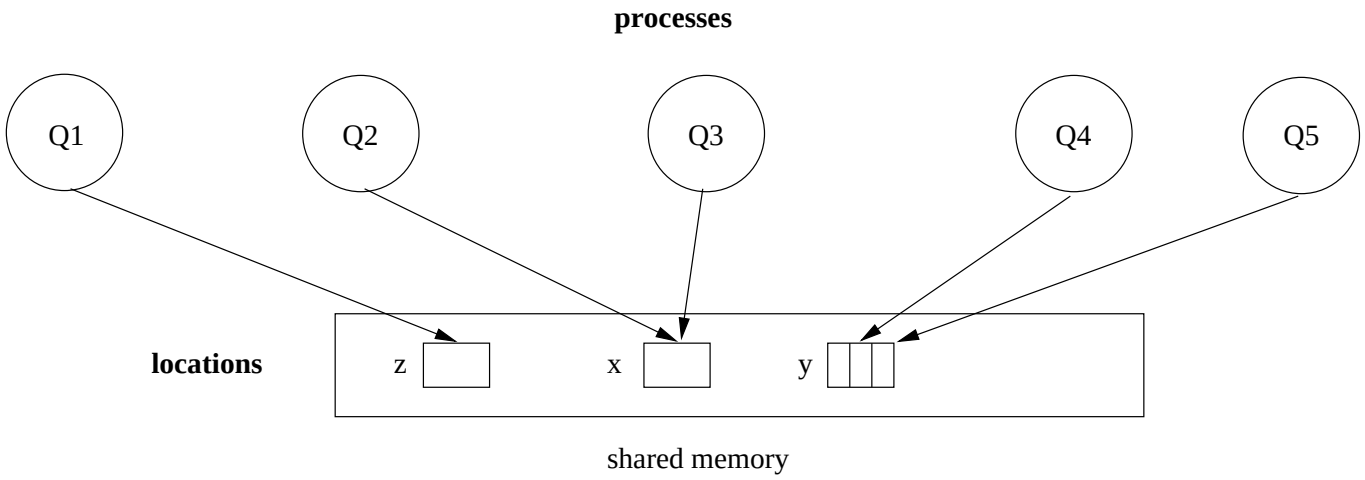
http://www.di.ens.fr/~goubault

ATMCS II, London Ontario, 17th of july 2004

Plan

- Introduction (geometry of concurrency?)
- A construction of component categories (follow up of Martin's talk)
- Using component categories, for static analysis
- Some further remarks (if time permits...)

Geometry of concurrency?



$\neq$ sequential; coordination problems (locks P, V)

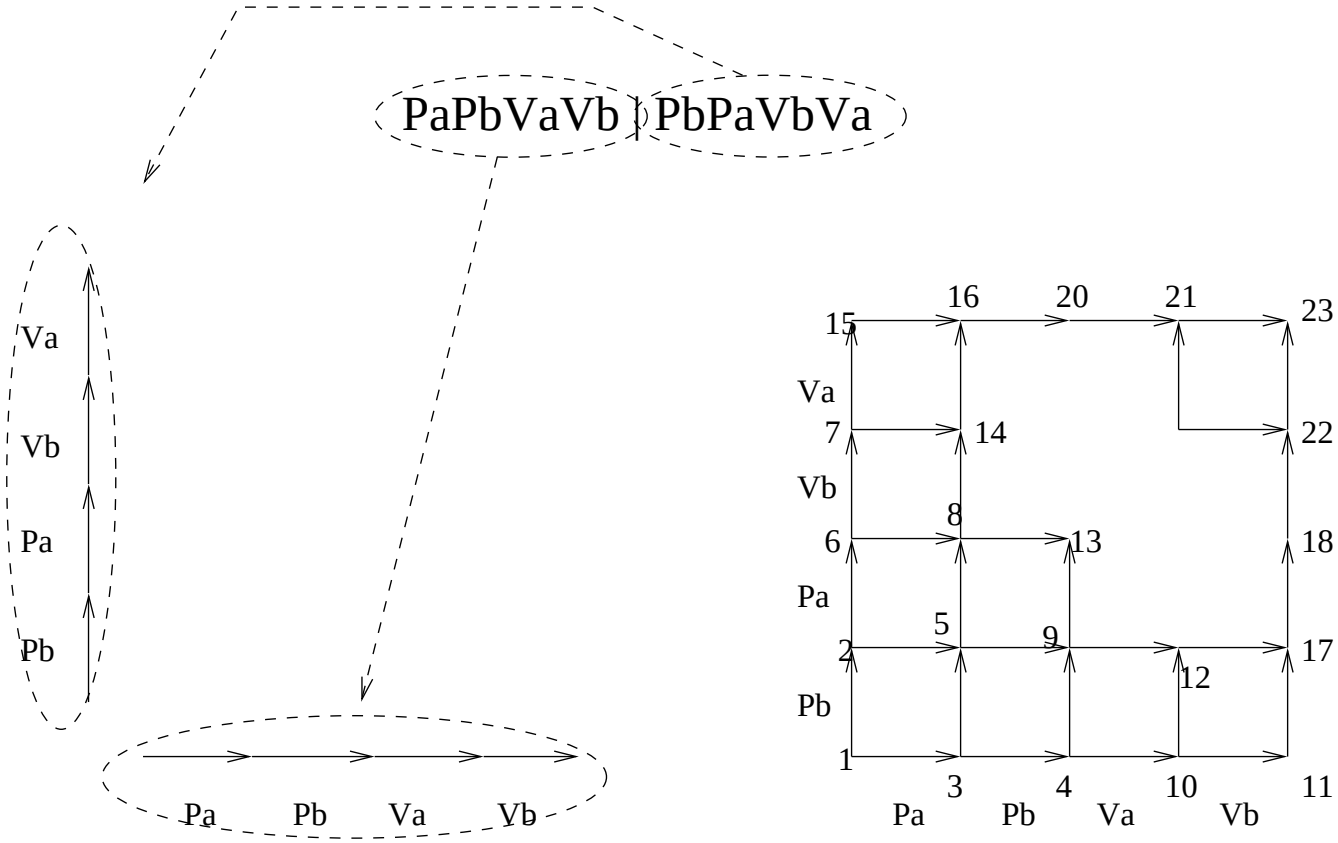
too constrained? $\longrightarrow$ deadlocks

bad states? $\longrightarrow$ reachability

"correct" access sequences? $\longrightarrow$ paths...

First model

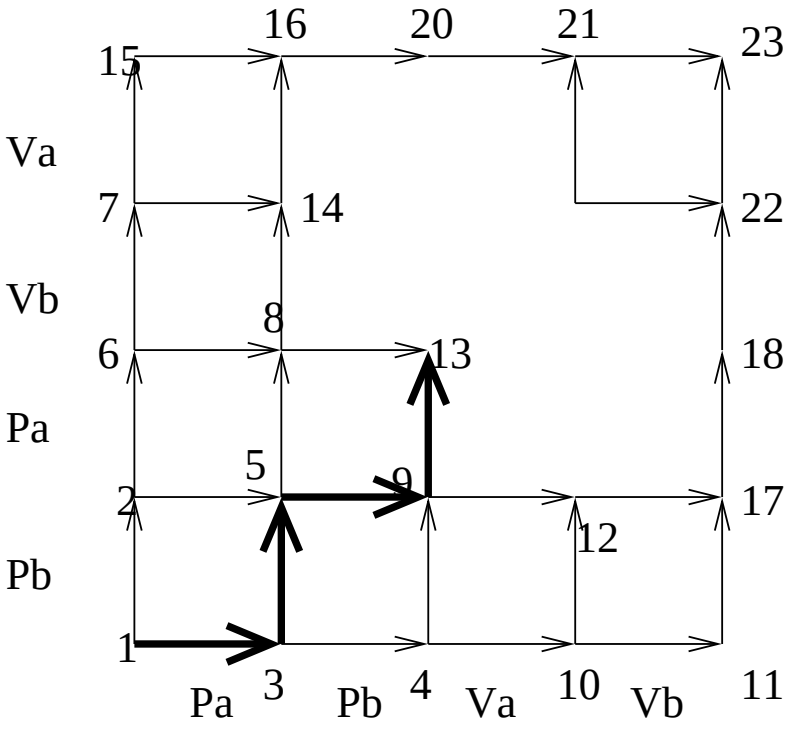
Processes: dynamical systems (discrete in general) = directed graphs of actions:



A potential execution

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

T_1	T_2
Pa	$-$
$-$	Pb
$Pb(?)$	$-$
$-$	$Pa(?)$

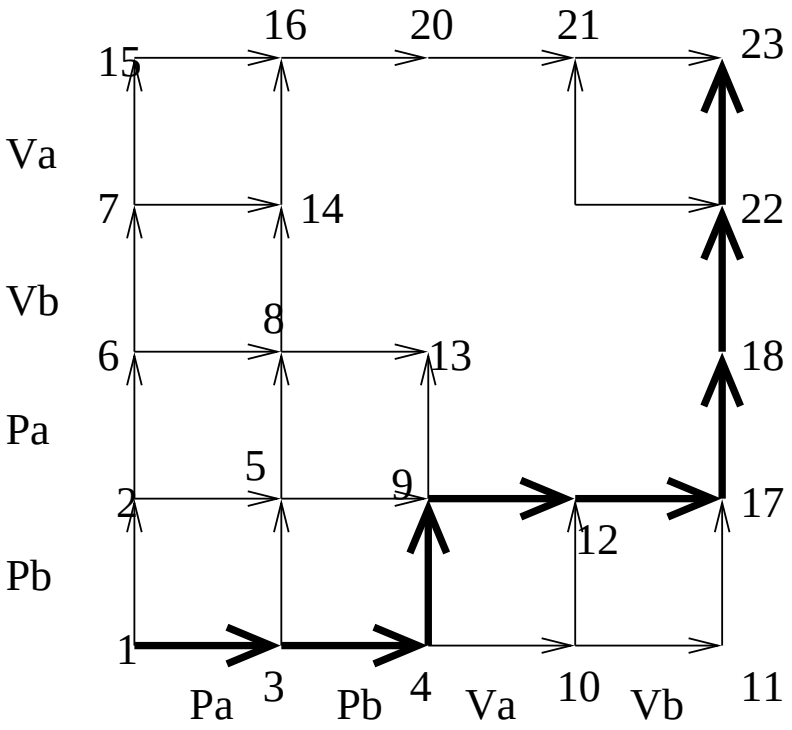


Deadlock!

Example

Another potential execution (16 in total!):

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



Notice that...

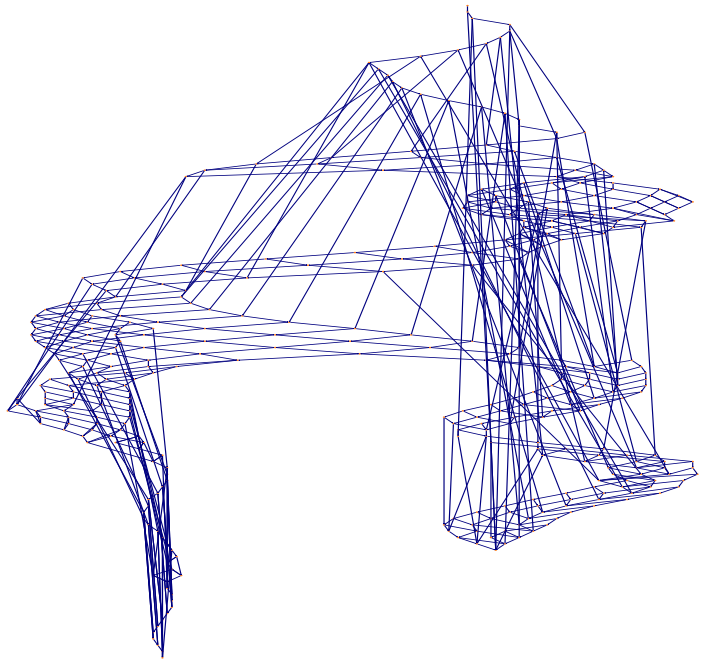
... there are very few "interesting" paths

Suppose, $T_1 = PaPb(b = b + 1)VaVb$ and $T_2 = PbPa(b = b * 2)VbVa$ and, in the beginning, $b = 2$:

- The 5 top left paths are "equivalent" to T_2 then T_1 : they compute $\underline{b = 5}$ (2*2+1)
- The 5 bottom right paths are "equivalent" to T_1 then T_2: they compute $\underline{b = 6}$ ((2+1)*2)
- The 6 paths near the diagonal are equivalent: they do not "terminate"

In general, this is really painful...

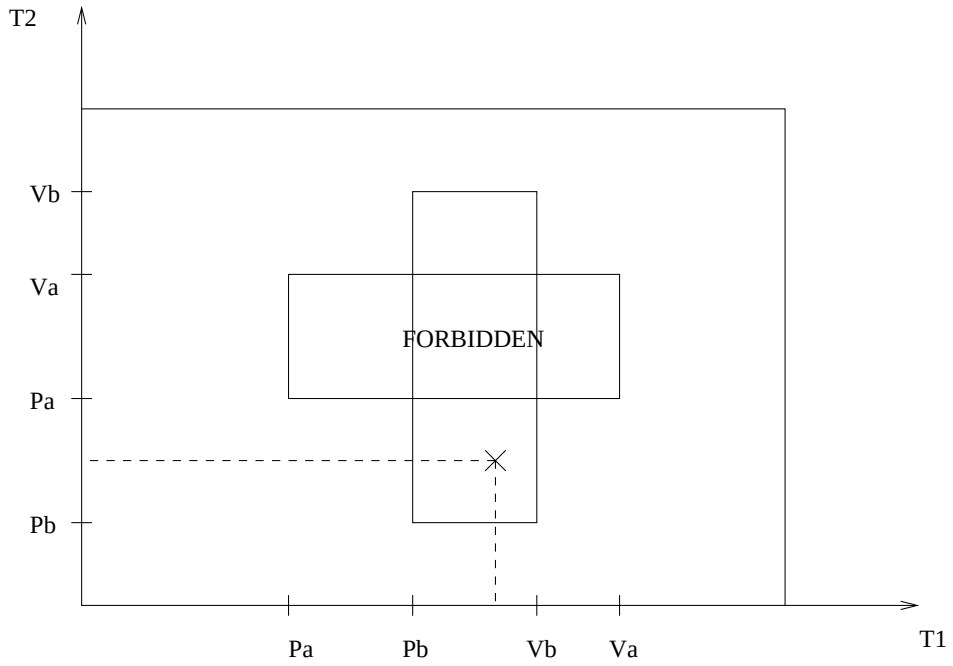
Big number of states: here, Dekker's algorithm, a few lines of C on 2 processes, a few hundreds of paths, only 2 of which are interesting!



Geometry?

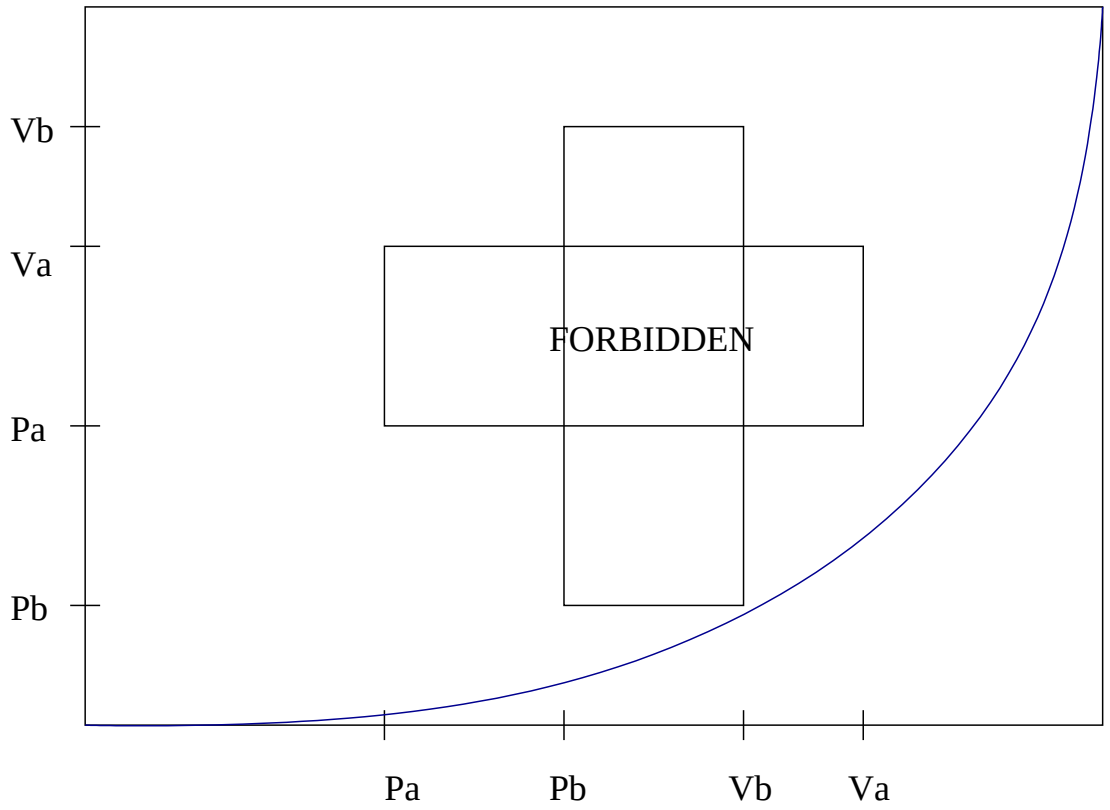
First idea: "Progress graphs" (E.W.Dijkstra (1968))

T1=Pa.Pb.Vb.Va in parallel with T2=Pb.Pa.Va.Vb



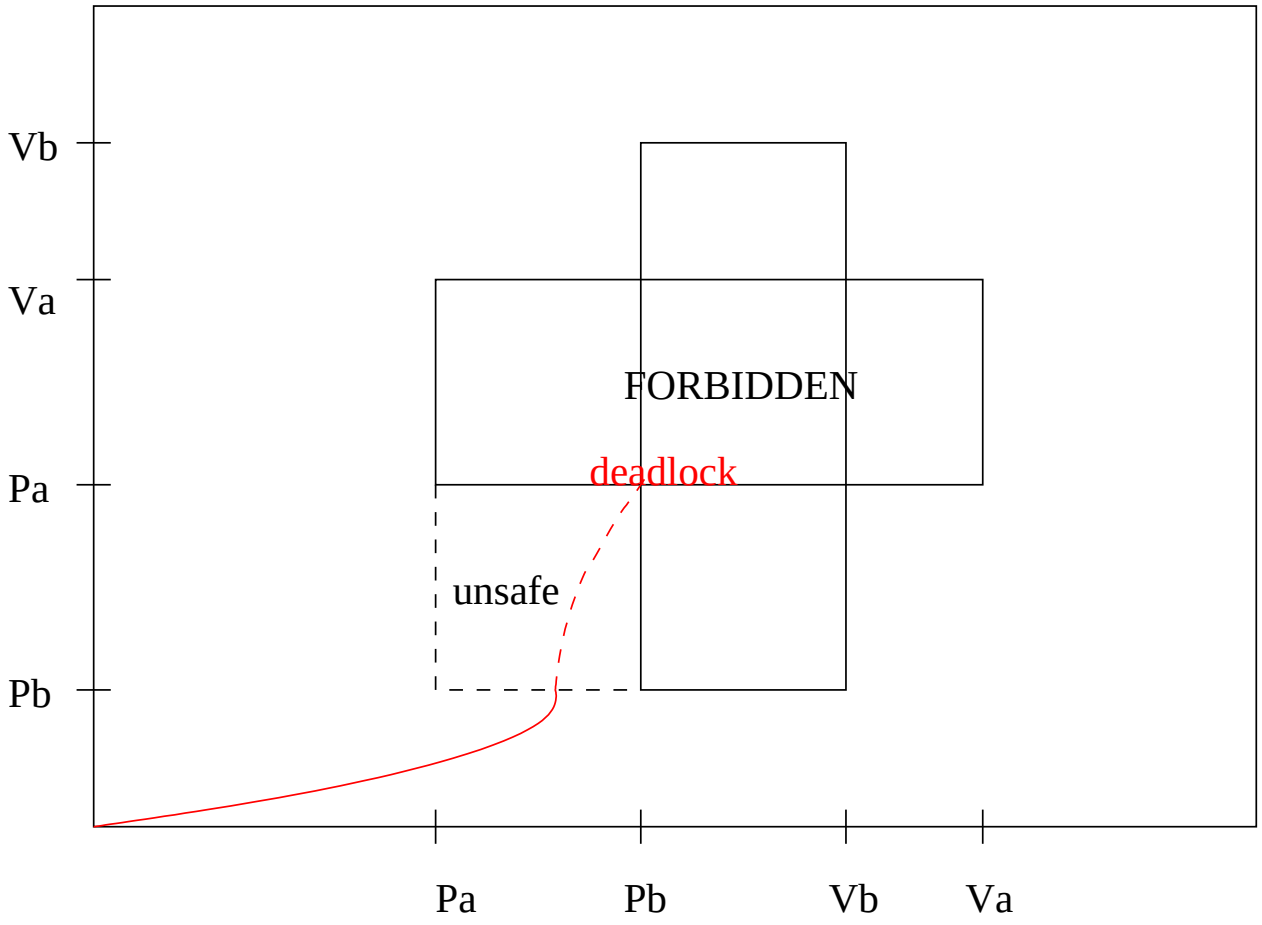
"Continuous image": x_i = local time; dashed rectangles=forbidden!

Execution paths



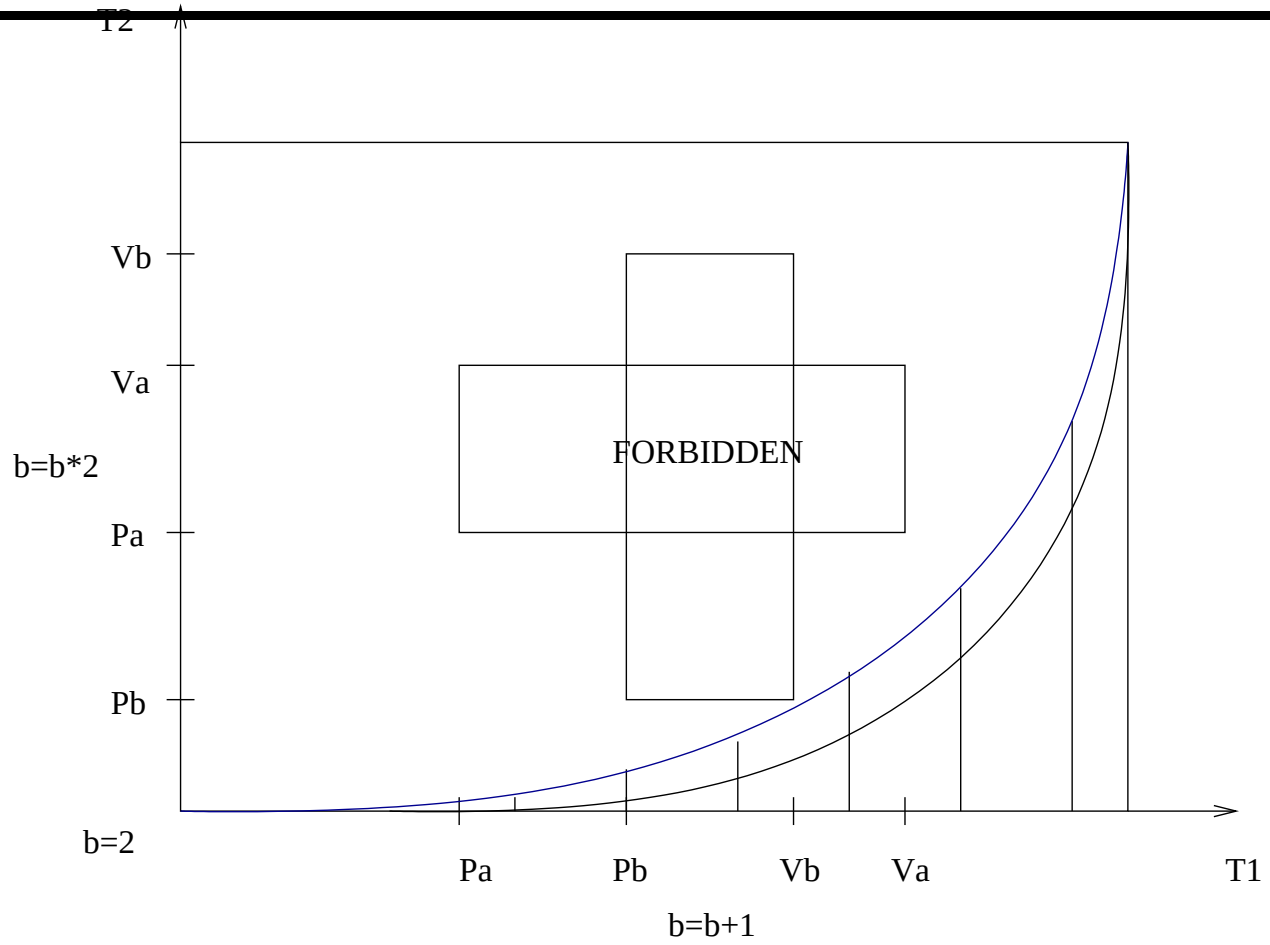
Traces = continuous paths increasing in each coordinate = “di-paths”

First remark



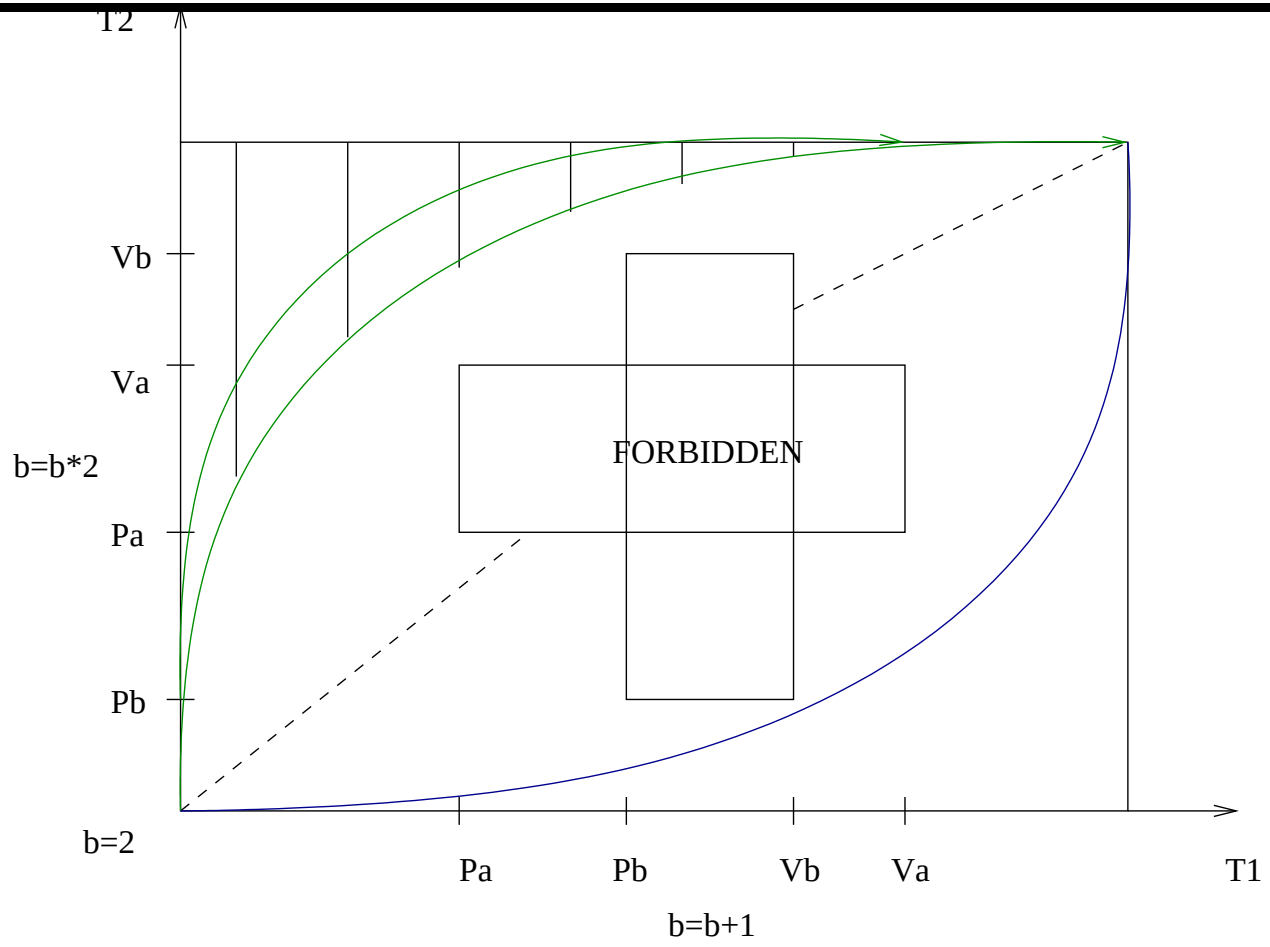
And dually...

Deformation of execution traces



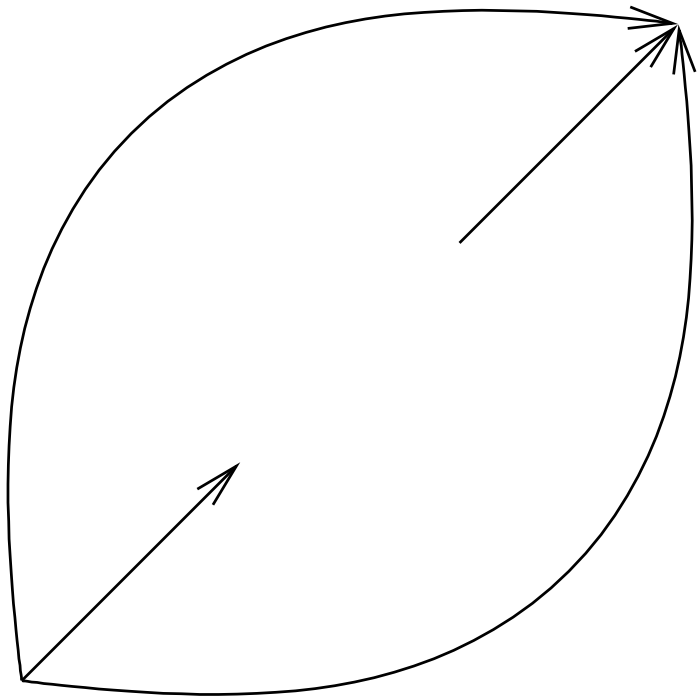
T1 gets a and b before T2 does: b=6

Non deformable paths

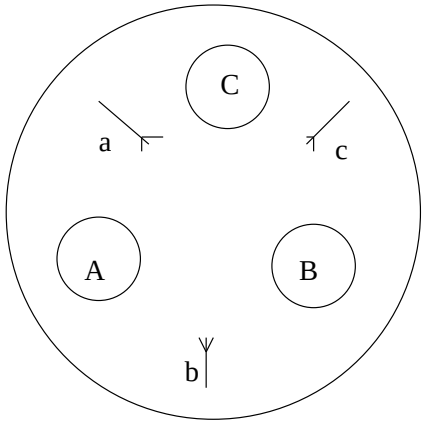


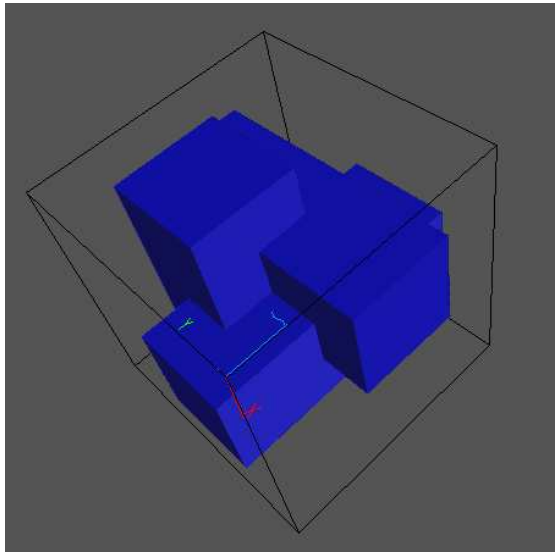
T2 gets a and b before T1 does: b=5!

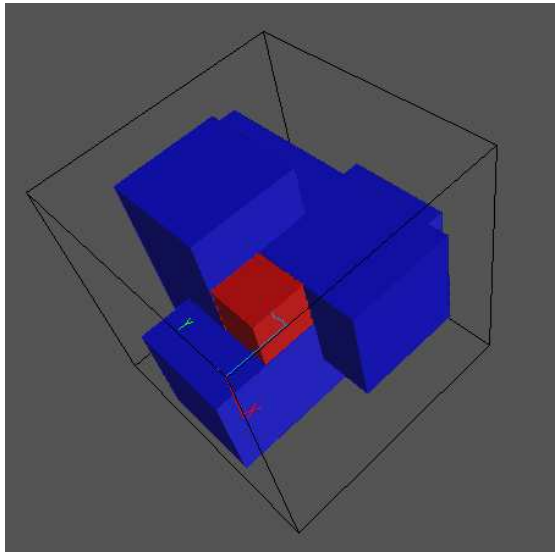
Ideally (not quite true though)...



In higher dimension





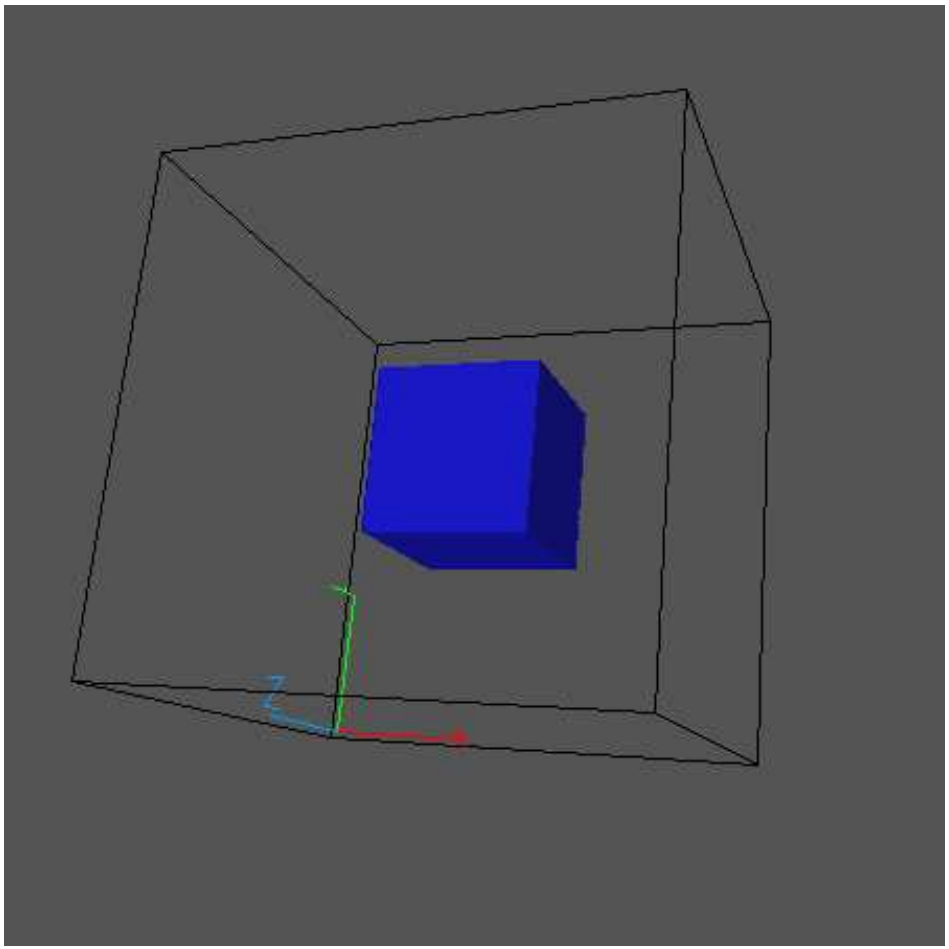


A=Pa.Pb.Va.Vb

B=Pb.Pc.Vb.Vc

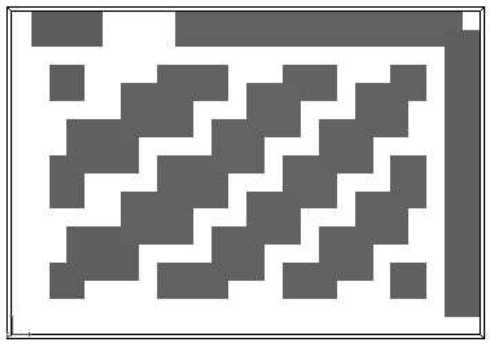
C=Pc.Pa.Vc.Va

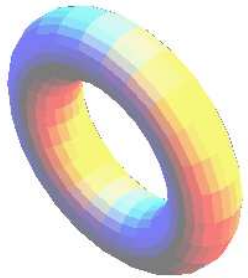
A counting semaphore



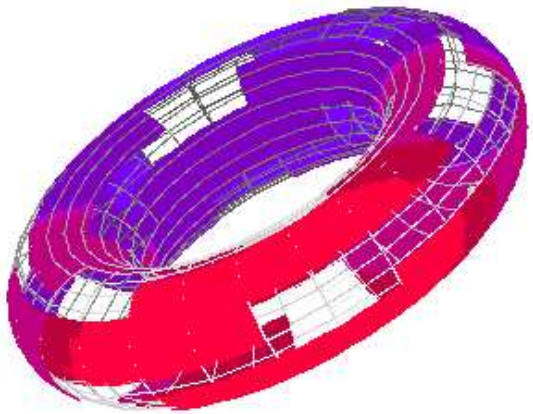
Loops?

```
A=Pd.Pa.(Pb.Va.Vd.Pc.Vb.Pa.Pd.Vc.Pb.Va.Pc.Vb.Pa.Vc.Pb.
                                    Va.Pc.Vb.Pa.Vc)*.Va.Pe.Vd.Ve
B=Pe.Pa.(Pb.Va.Pc.Vb.Pa.Vc.Pb.Va.Pc.Vb.Pa.Vc)*.
                                    Va.Pd.Ve.Vd
```





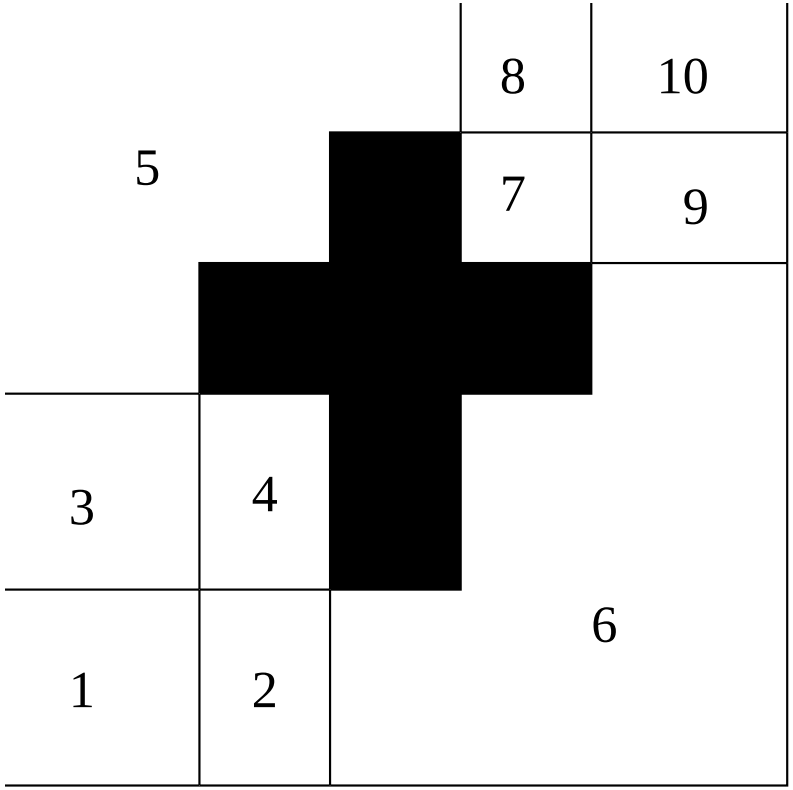
Gives...

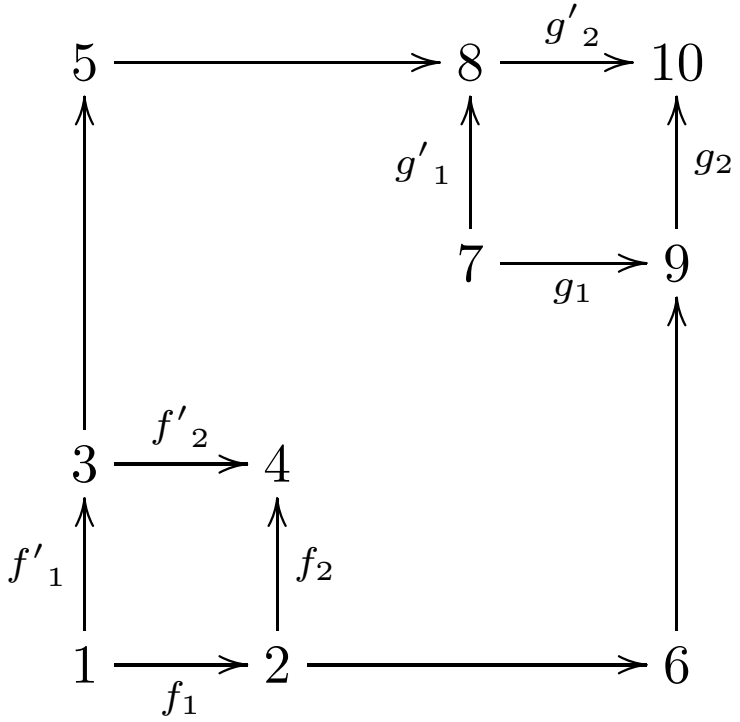


No deadlock, but 2 turns in the loop T_1 / 3 turns for T_2

Model [discrete]	combinatorial complex
Model [continuous]	topological space
Relation discrete/continuous	geometric realisation
Parallel composition	product
Action refinement	subdivision
Compositionality	Seifert/van Kampen
Deadlocks/reachability	connected components
Scheduling properties	fundamental group
Observational equivalence	homotopy equivalence (weak/strong)
Computable properties	topological invariants (homology etc.)
Models/Good structures	Quillen model categories

We are aiming at...





and the relations

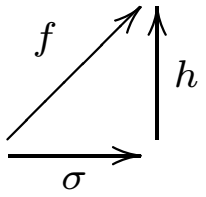
$g'_2 \circ g'_1 = g_2 \circ g_1$ and

$f'_2 \circ f'_1 = f_2 \circ f_1$

Future components

~~Let $\mathcal{C}$ be a directed category. Σ is a subcategory of future weak~~ equivalences (f-WE) if:

- Σ contains all isos of $\mathcal{C}$.
- all σ in Σ are epis in $\mathcal{C}$.
- Σ is stable under pushout (which always exists, with any morphim in $\mathcal{C}$)[a].
- If there is $u : \beta \to \gamma$ in $\mathcal{C}$, then for all $\sigma : \alpha \to \beta$ in Σ, and all $f : \alpha \to \gamma$ in $\mathcal{C}$, f factors through σ, that is, there exists $h : \beta \to \gamma$ such that the following diagram commutes



[a] Alternative, weaker condition, by Lisbeth Fajstrup.

Duals and existence

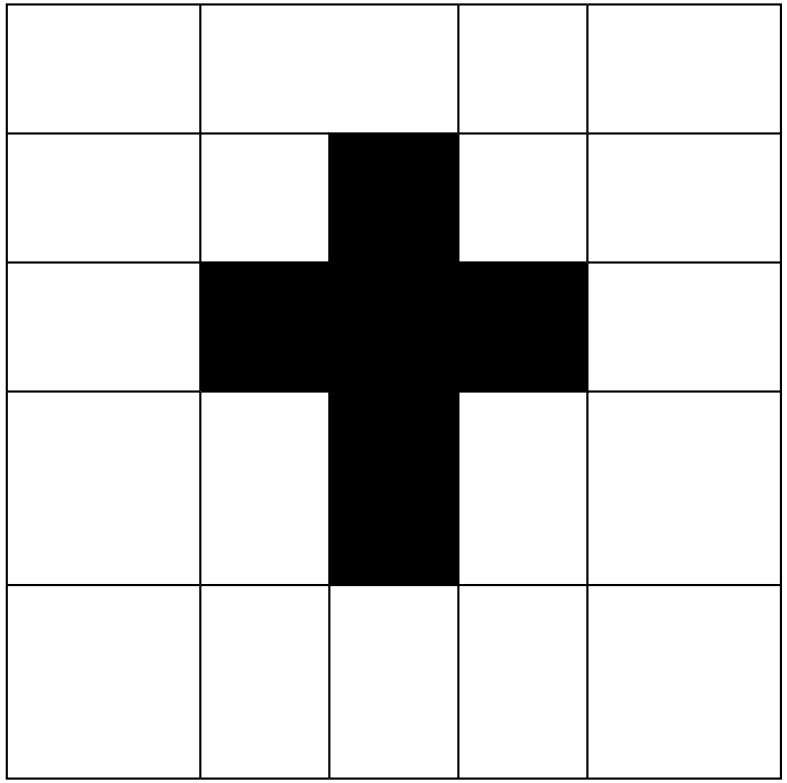
- Dual notion p-WE, bi-notion WE.
- In directed categories, there always exists f-WE, p-WE and WE subcategories, namely the subcategory of isomorphims of $\mathcal{C}$.

In general, nice framework: directed categories

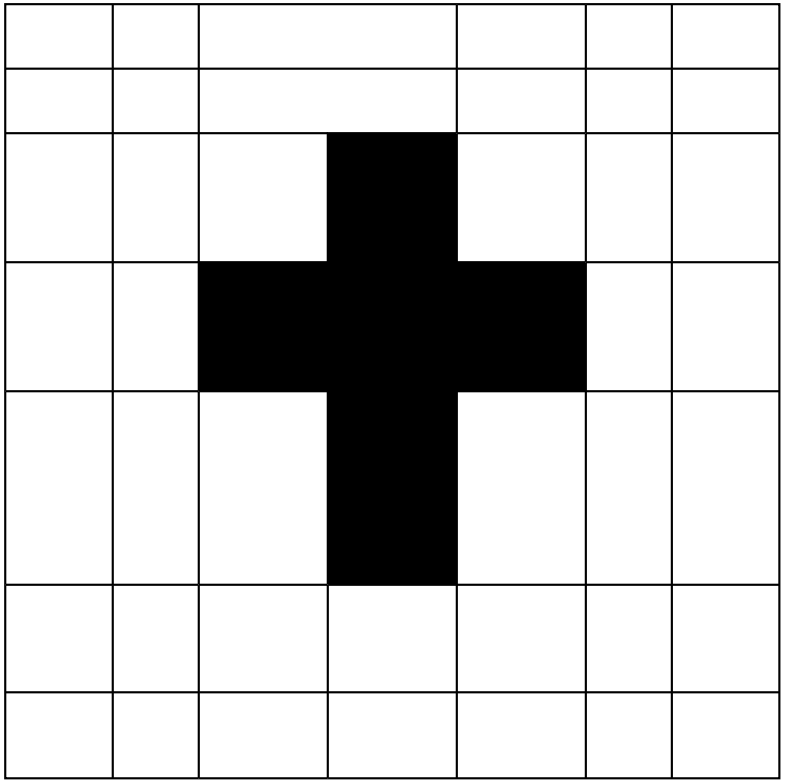
- Let $\mathcal{C}$ be a category, Σ a subcategory of $\mathcal{C}$. Σ is pure in $\mathcal{C}$ if for all f, g in $\mathcal{C}$, $f \circ g \in \Sigma$ implies $f \in \Sigma$ and $g \in \Sigma$.
- A directed category is a category whose subcategory of isomorphisms is pure.
- For X a po-space, local po-space etc. $\vec{\pi}_1(X)$ is a directed category.

Important property

There exists a maximal one (the "optimal" state-space reduction)



(interleaving - suboptimal)



Maximal WE subcategories

Let $\mathcal{C}$ be a directed category. There exists a maximal subcategory of WE (f-WE, p-WE respectively) in $\mathcal{C}$.

Sketch of proof.

There exists one, as we saw earlier.

Now, it suffices to show that if τ is a morphism of $\mathcal{C}$ satisfying all axioms of a subcategory of WE (as the category generated by τ) and if Σ is a subcategory of WE, then all composites $\tau \circ \sigma$ and $\sigma \circ \tau$, together with Σ, is a subcategory of WE. By induction, we see that being WE is inductive. Using Zorn's lemma, we find the maximal WE.

WE and left/right calculi

- p-WE form right calculi, and
- f-WE form left calculi,
- WE form left and right calculi.

Sketch of proof.

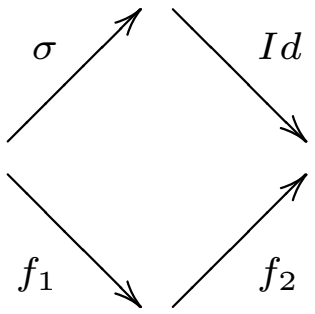
It is immediate to see that the fact that all σ in Σ are monos imply (iv). Then having pullbacks between any σ in Σ and f in $\mathcal{C}$ trivially implies the extension property (iii).

Purity

Let Σ be a subcategory of WE (respectively, f-WE, p-WE) in $\mathcal{C}$, then Σ is (respectively left, right) pure.

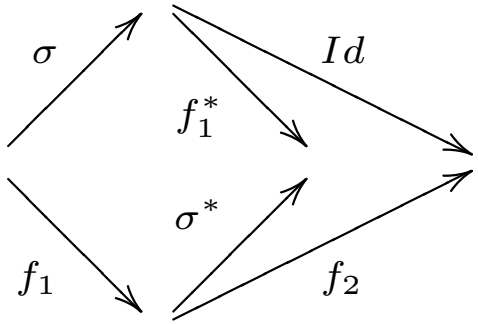
Sketch of proof.

Suppose σ in Σ is equal to $f_2 \circ f_1$ with f_1 and f_2 in $\mathcal{C}$:



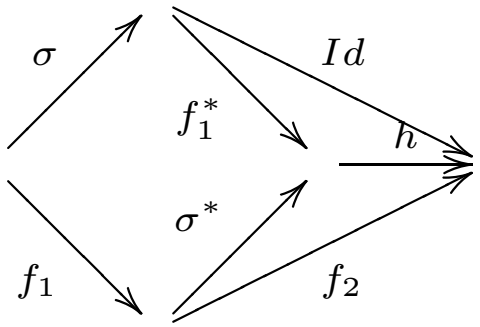
Sketch of proof

As σ is in Σ, we have a pushout between σ and f_1, hence we have the commutative diagram



Sketch of proof

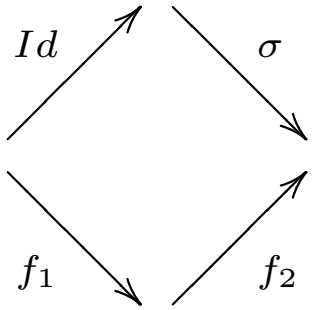
hence, by the universality of the pushout, we have a unique h in $\mathcal{C}$ such that the diagram below commutes



This implies that $Id = h \circ f_1^*$. Hence Id (an iso) is factored - by purity of isos in directed categories, h and f_1^* are isos. Hence h is in Σ. This also implies that $f_2 = h \circ \sigma^*$, a composite of two elements of Σ, hence is in Σ.

Sketch of proof

Now we can rewrite $\sigma = f_2 \circ f_1$ as the commutative diagram



and use the universal property of the pullback of σ along f_2, to conclude in the same manner that f_1 is in Σ.

Consequence...

~~Using the maximal WE, we have all the results from APCS (see~~ Martin's talk), and in particular:

Let $\mathcal{C}$ be a category in which all endomorphisms are identities. Then there is a maximal WE subcategory Σ in $\mathcal{C}$. Furthermore, let $C_1, C_2 \subset Ob(\mathcal{C})$ denote two components such that the set of morphisms (in $\mathcal{C}/\Sigma$) is *finite*[a]. Then, for every $x_1 \in C_1$ there exists $x_2 \in C_2$ such that the quotient map

$$\mathcal{C}(x_1, x_2) \to \mathcal{C}/\Sigma(C_1, C_2), \ ; \ f \mapsto [f]$$

is *bijective*.

Similarly with just f-WE (we can choose the target point only) and the p-WE (we can choose the source point only).

[a] If we ask for a little more on Σ, no restriction here!

How can we solve the problem...

... of the characterisation of the schedulings (or "essential" traces)?

- By applying some general theorems (van Kampen, see E. Haucourt's [CEA & Paris 7] work etc.)
- By some specific algorithms (in the case of $\mathbf{R}^n$): implemented, with some algorithmic improvements to come.

Component categories generated by bidimensional cubical sets

For the mutual exclusion case ($X = \overrightarrow{I}^n \setminus R$), the component category is generated by a bidimensional cubical set.

$$F : \Upsilon_2 \to \mathcal{C}$$

- let D be the free category generated by the graph (X_0, X_1) underlying X.
- $F(X)$ is the quotient of D by relations:
  $[d^1_0(A)] \circ [d^0_1(A)] = [d^1_1(A)] \circ [d^0_0(A)]$ where $A \in X_2$ is a 2-cell.

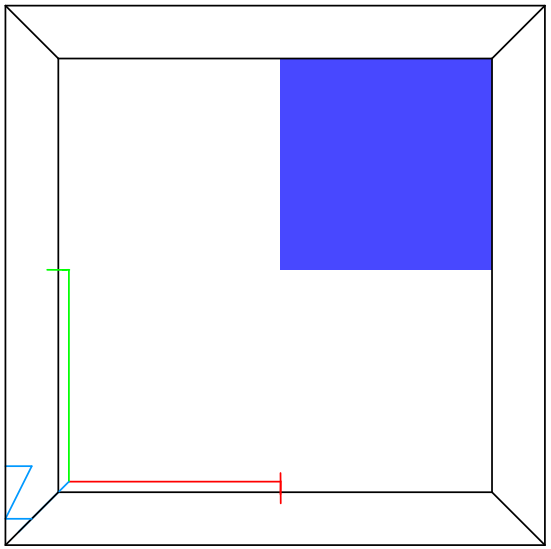
A simple case

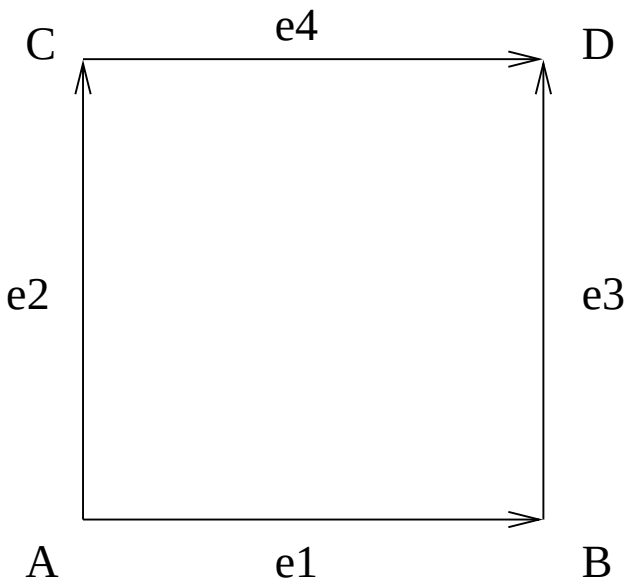
The component category for the following program:

```
A=Pa.Va
B=Pa.Va
PROG=A|B
```

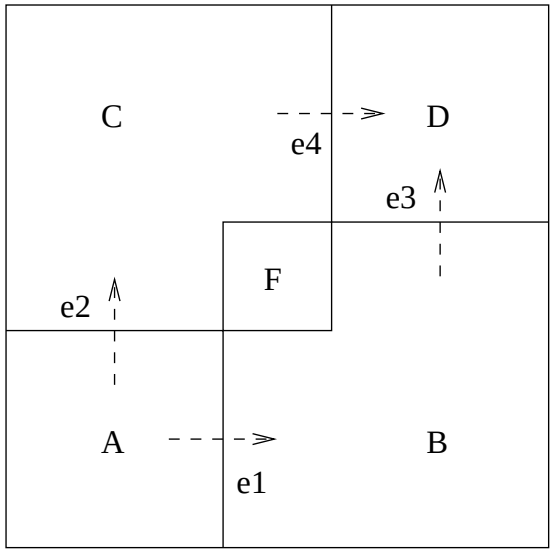
is given by the following picture:

Simple case





The component category, geometrically



The components A, B, C and D correspond to the squares separated by the horizontal and vertical lines from the min and max points of the forbidden region F.

The component category: "duality"

- The unique morphism from A to B (e_1) is the "orthogonal" of the vertical line coming from the min point of F.
- We identify e_1 with the codimension 1 linear variety (here, the vertical segment)
- Similarly, e_2 is identified with the horizontal line left of the min point of F etc.
- There is no interesting codimension 2 linear variety here, hence no relations between morphisms.

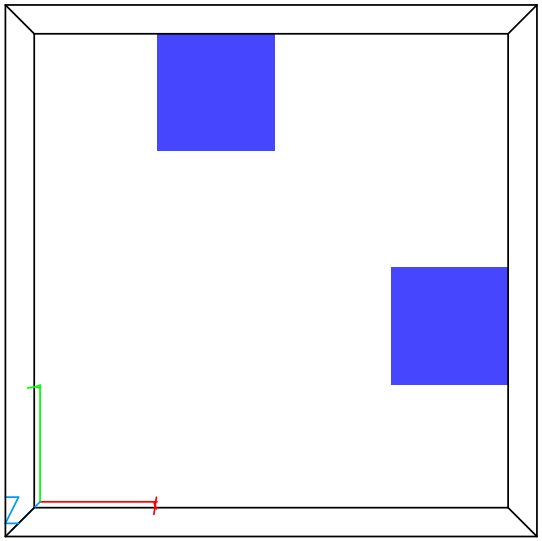
Digging a hole

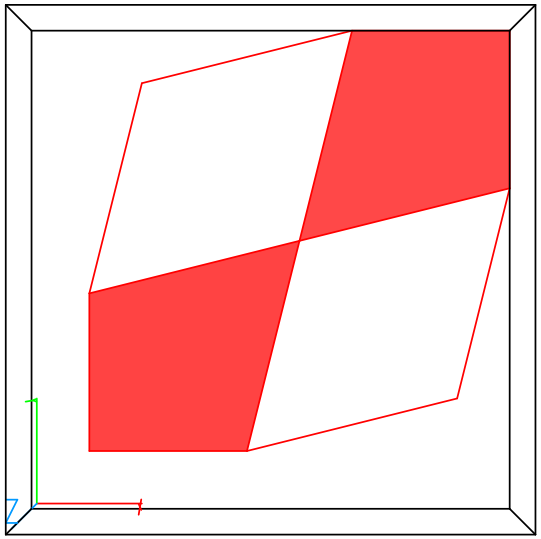
This corresponds here to the program:

```
A=Pa.Va.Pb.Vb
B=Pb.Vb.Pa.Va
PROG=A|B
```

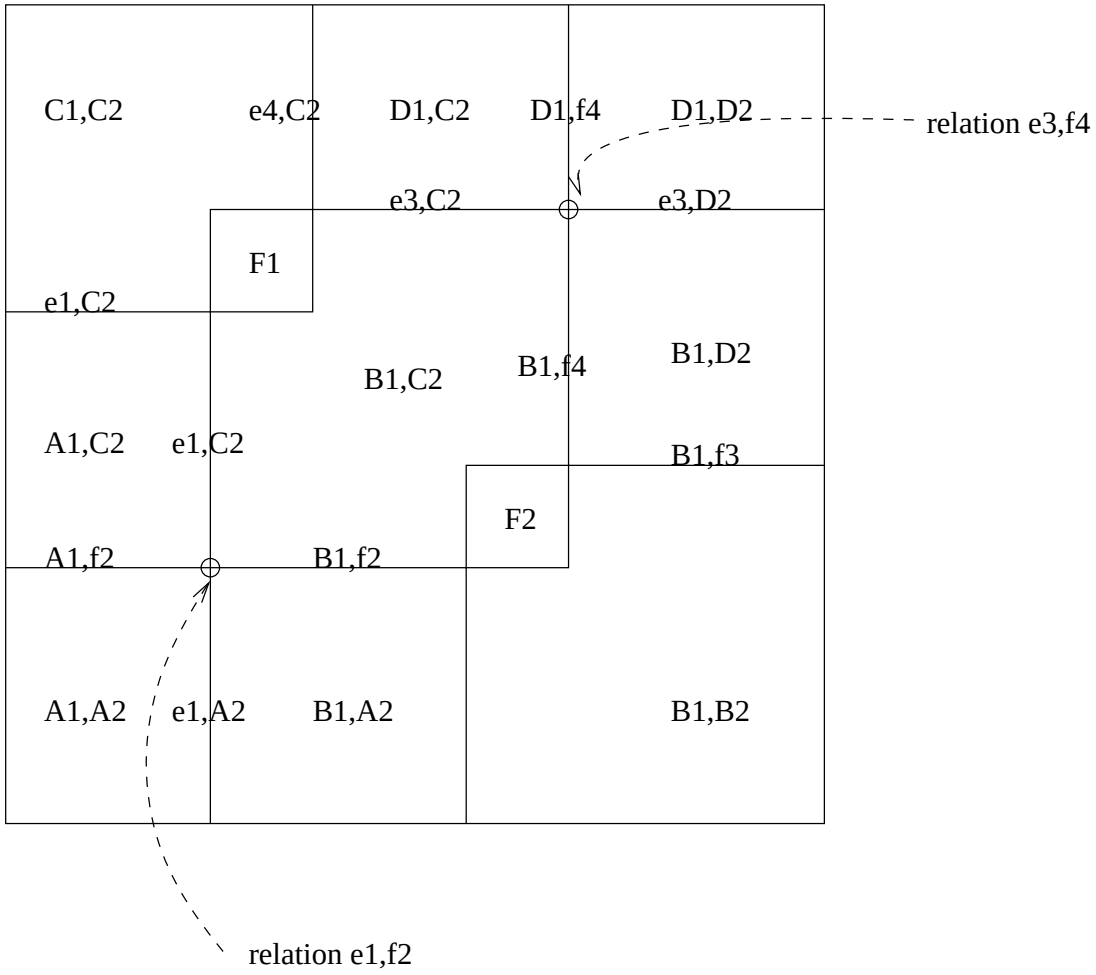
and the component category is:

Component category





Inductive formula



Inductive formula

Inductive formula: based on the intersections of codimension 0, 1 and 2 linear varieties, for all holes dug:

- Starting with (e_1, e_2, e_3, e_4) (for F_1) and (f_1, f_2, f_3, f_4) (for F_2)
- We compute all intersections of objects (regions) (A_1, B_1, C_1, D_1) with morphisms of F_2 and of morphisms of F_1 with objects (A_2, B_2, C_2, D_2) of F_2
- We have for instance the relation (e_1, f_2) (intersection of two morphisms):

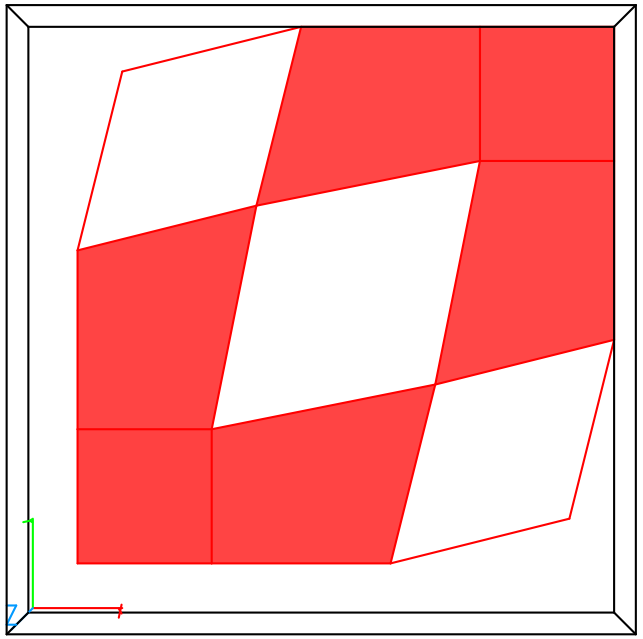
$$(e_1, C_2) \circ (A_1, f_2) = (B_1, f_2) \circ (e_1, A_2)$$

Let us dig again...

...in order to compute the component category of:

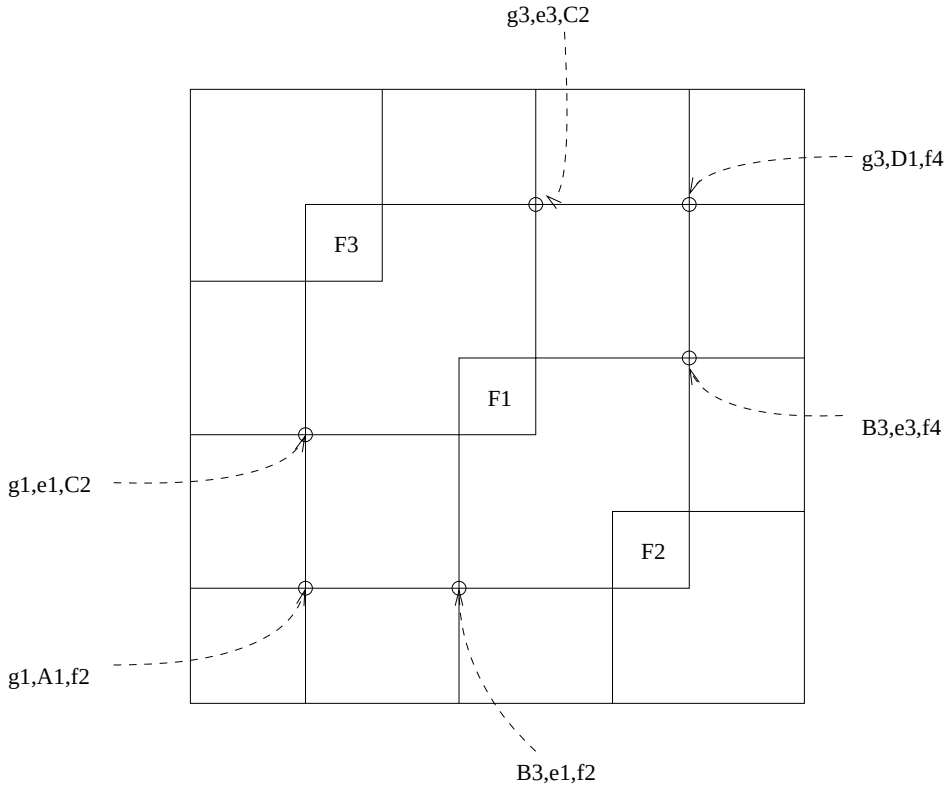
```
A=Pa.Va.Pb.Vb.Pc.Vc
B=Pc.Vc.Pb.Vb.Pa.Va
PROG=A|B
```

Component category

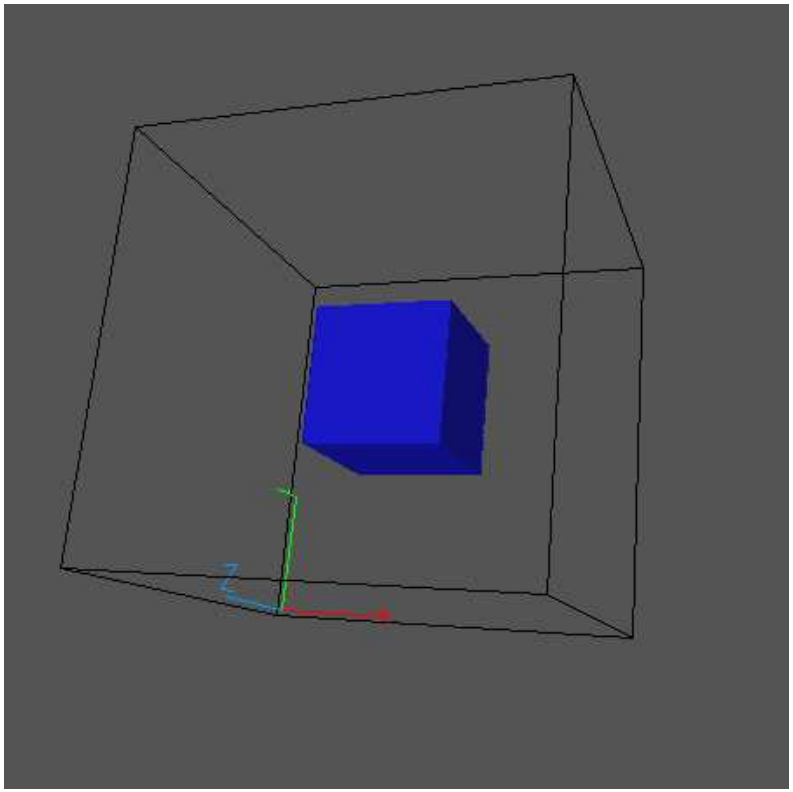


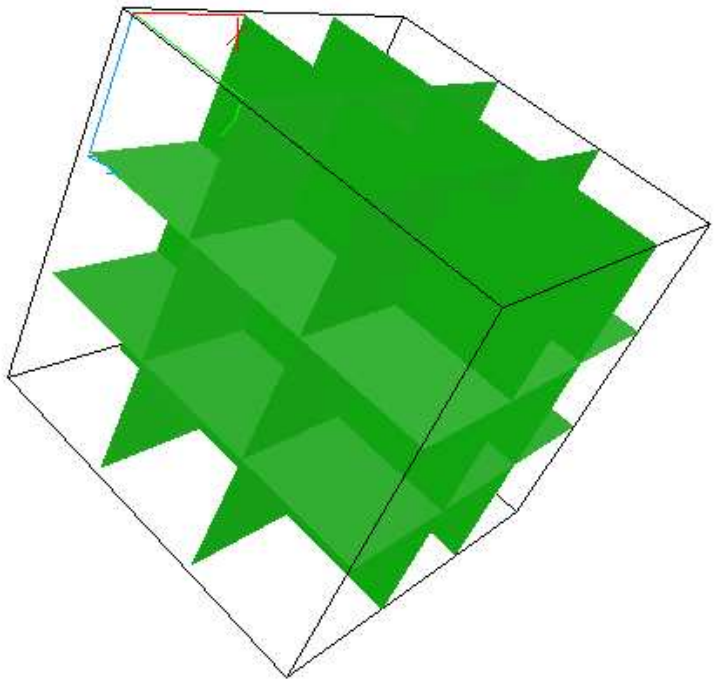
Inductive "explanation"

The six relations we find are given in the picture (g_1, g_2, g_3 and g_4 are the morphisms of the component category of F_3 alone).



2 semaphore





Inductive presentation of the component category

We already know that the component category of $[0,1]^n \backslash R$ is generated by a 2-dimensional precubical set, that we write as $(Y_0, Y_1, Y_2, \delta^0, \delta^1)$. We now define a new structure $(Z_0, Z_1, Z_2, \partial^0, \partial^1)$ which will generate (an "approximation" of) the component category of $U \backslash R$:

- $Z_0 = \{A \cap B \mid A \in X_0, B \in Y_0, A \cap B \neq \emptyset\}$
- $Z_1 = \begin{array}{l} \{A \cap f \mid A \in X_0, B \in Y_1, A \cap f \neq \emptyset\} \\ \cup \{e \cap B \mid e \in X_1, B \in Y_0, e \cap B \neq \emptyset\} \end{array}$

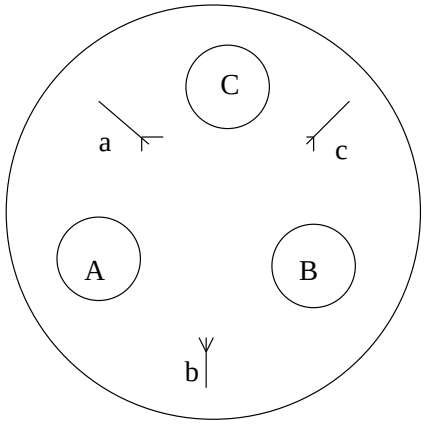
Inductive presentation of the component category

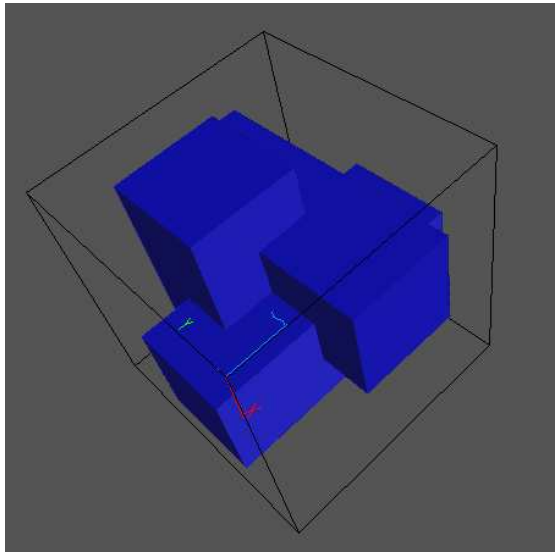
- $Z_2 = \begin{array}{l} \{e \cap f \mid e \in X_1, f \in Y_1, e \cap f \neq \emptyset\} \\ \cup\{R \cap B \mid R \in X_2, B \in Y_0, R \cap B \neq \emptyset\} \\ \cup\{A \cap S \mid A \in X_0, S \in Y_2, A \cap S \neq \emptyset\} \end{array}$
- $\partial^*_* : Z_1 \to Z_0$ are defined by:
  - $\partial^0_0(A \cap f) = A \cap \delta^0_0(f)$,
  - $\partial^1_0(A \cap f) = A \cap \delta^1_0(f)$,
  - $\partial^0_0(e \cap B) = d^0_0(e) \cap B$,
  - $\partial^1_0(e \cap B) = d^1_0(e) \cap B$.

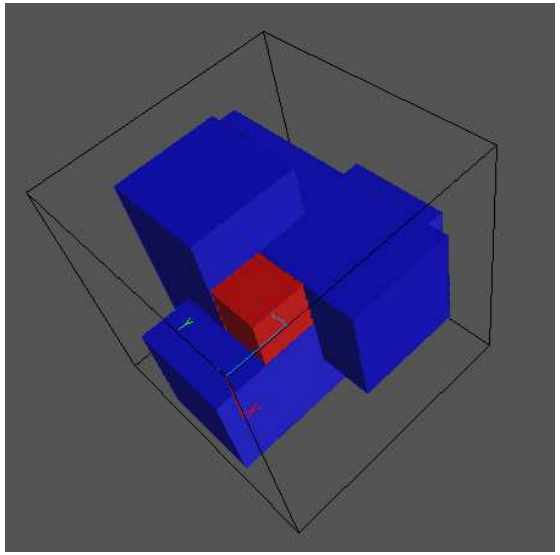
Inductive presentation of the component category

- $\partial^*_* : Z_2 \to Z_1$ are defined by:
  - $\partial^0_0(e \cap f) = d^0_0(e) \cap f$,
  - $\partial^0_1(e \cap f) = e \cap \delta^0_0(f)$,
  - $\partial^1_0(e \cap f) = d^1_0(e) \cap f$,
  - $\partial^1_1(e \cap f) = e \cap \delta^1_0(f)$,
  - $\partial^k_l(R \cap B) = d^k_l(R) \cap B$ (for $k, l = 0, 1$),
  - $\partial^k_l(A \cap S) = A \cap \delta^k_l(S)$.

The 3 philosophers





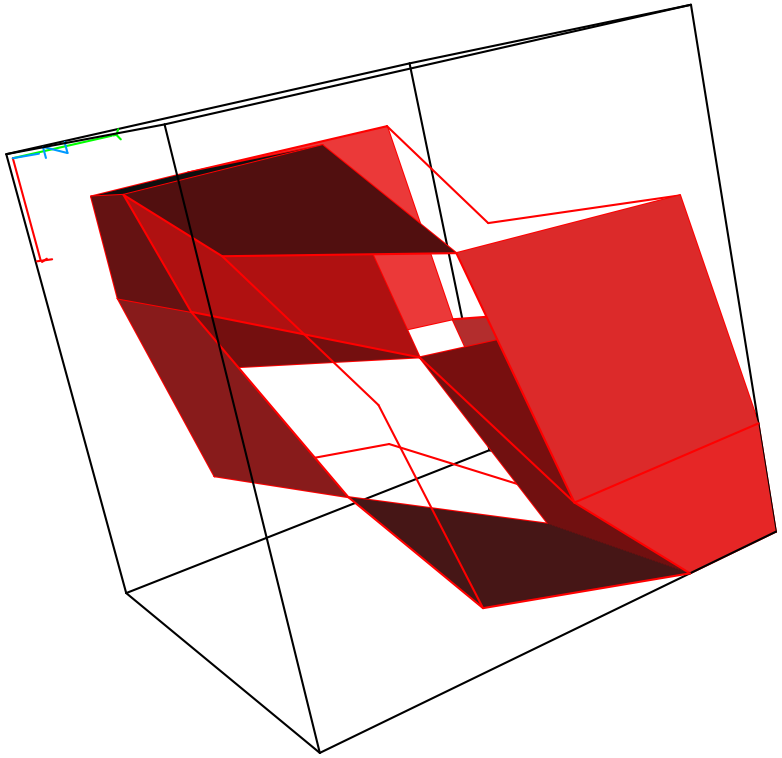


A=Pa.Pb.Va.Vb

B=Pb.Pc.Vb.Vc

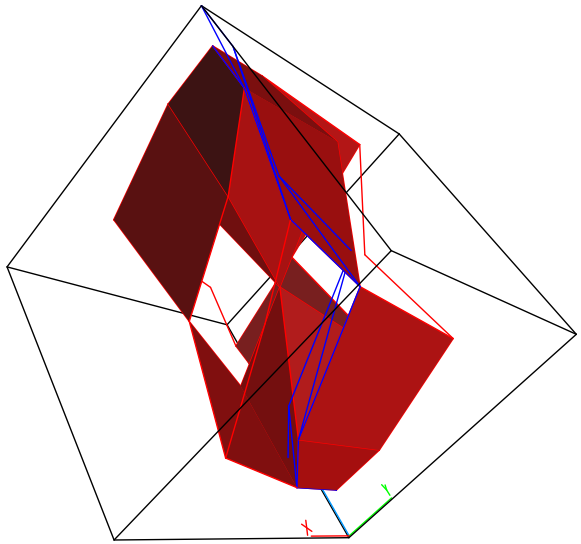
C=Pc.Pa.Vc.Va

Component category

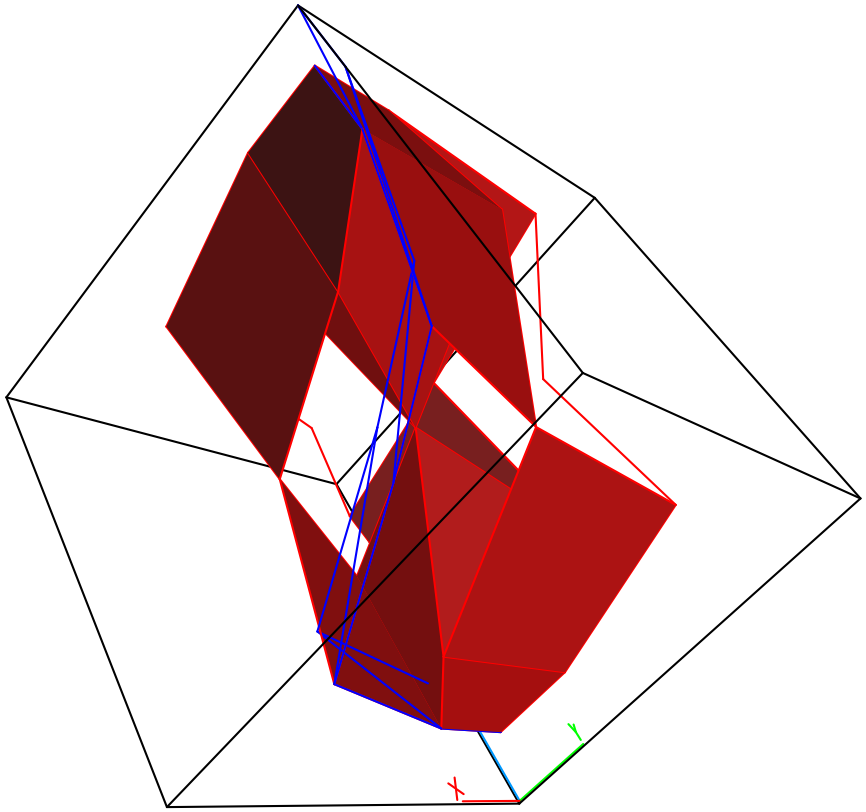


Example

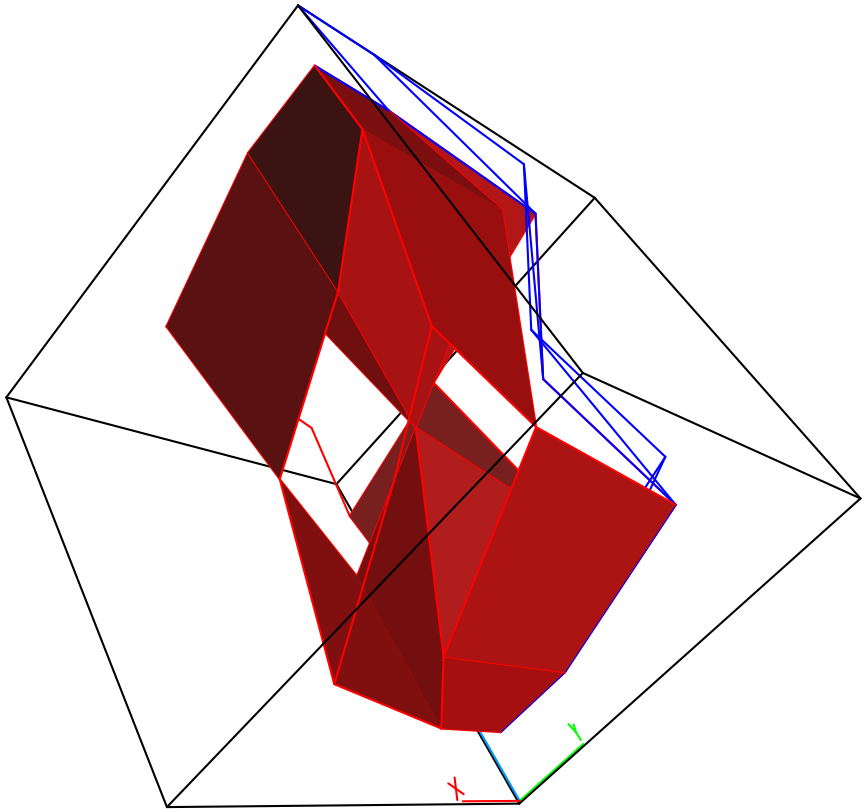
In the case of the 3 philosophers, we find 6 (=3!) maximal legal paths (all possibles serializations in fact):



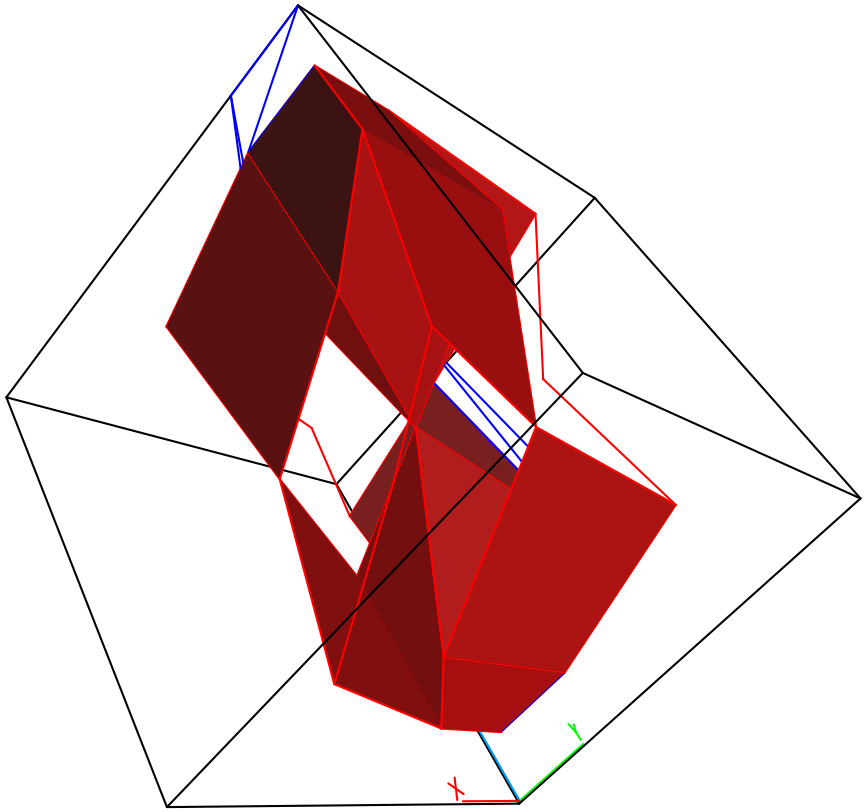
Path 3



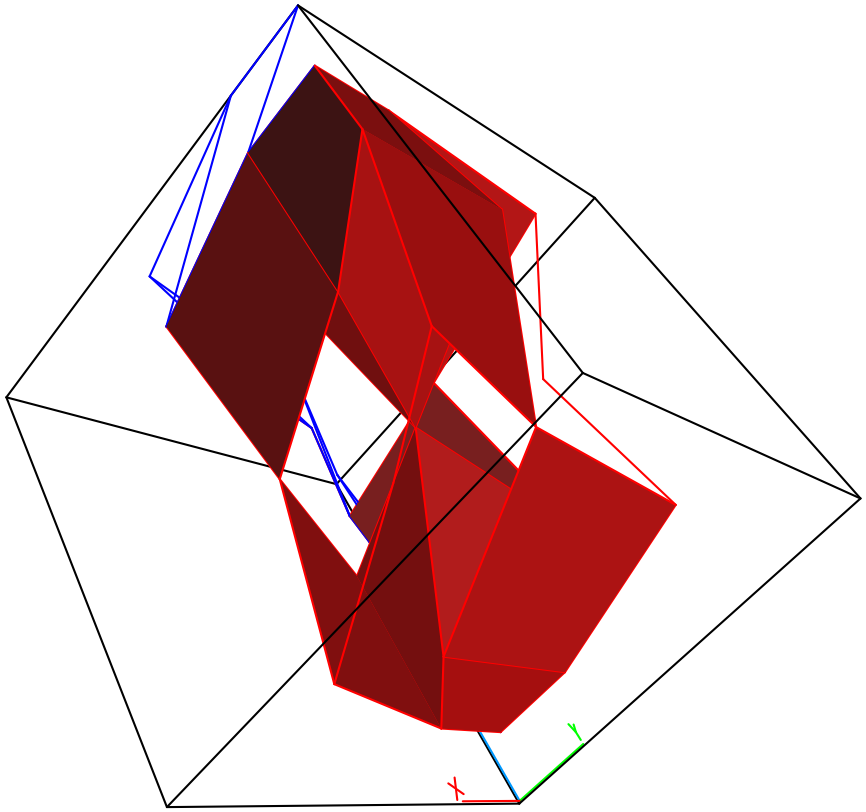
Path 4



Path 5



Path 6



Some figures...

- new3phil.pv: (0.05s) Objects: 27, Morphisms: 48, Relations: 18
- new4phil.pv: (0.07s) Objects: 85, Morphisms: 200, Relations: 132
- new7phil.pv: 4s; 42 Mo; (about one million transitions in a standard model) Objects: 2467, Morphisms: 10094, Relations: 15484
- new8phil.pv: 21s; 100Mo; (about 10 million transitions in a standard interleaving model) Objects: 7533, Morphisms: 35216, Relations: 64312
- new9phil.pv: 106s, 319Mo (about 100 million transitions in a standard model) Objects: 22995, Morphisms: 120924, Relations: 256158 for 9!=362880 maximal paths...

Syntactic representation of a path...

...or how to choose a "good" lift:

- From a continuous path, we want to get back to a "discrete" path
- This "discrete" path is an interleaving path corresponding to this idealized execution
- This can be then analyzed by any standard sequential analyzer

Example

```
> more simple.pv
A=Pa.Va.Pb.Vb
B=Pb.Vb.Pa.Va
PROG=A|B
> essentialpaths simple.pv
(...)
Number of paths=3
```

Example

```
> more simple.pv_1.sp
{1}P(a);{1}V(a);{1}P(b);{1}V(b);{2}P(b);
  {2}V(b);{2}P(a);{2}V(a);
> more simple.pv_2.sp
{2}P(b);{2}V(b);{2}P(a);{2}V(a);{1}P(a);
  {1}V(a);{1}P(b);{1}V(b);
> more simple.pv_3.sp
{1}P(a);{2}P(b);{1}V(a);{2}V(b);{2}P(a);
  {2}V(a);{1}P(b);{1}V(b);
```

For this...

We remark that:

- (1) Every component has a trivial $\vec{\pi}_1$
- (2) There exists a path (unique) from the minimum (or infimum in general) from a component to the minimum of the next component (essentially by the lifting property)

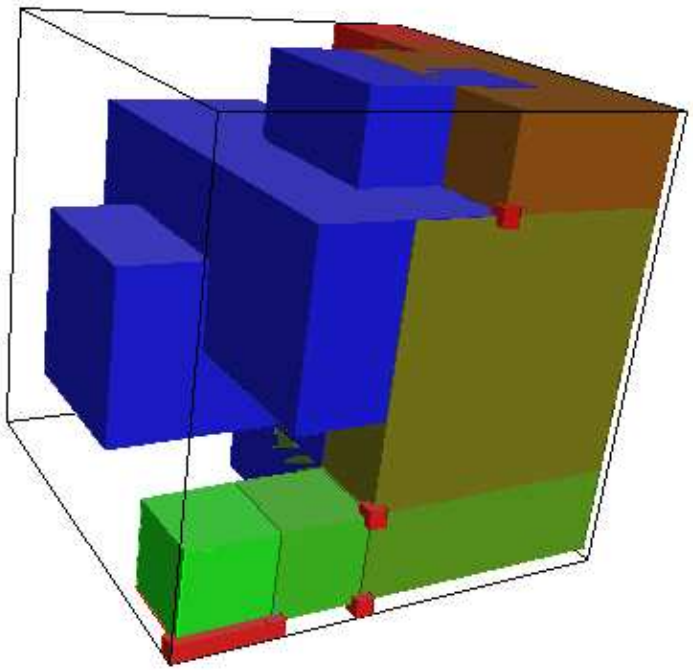
  (modulo some technicalities I am hiding here...)

For this...

Given the morphisms of the component category, we compute:

- (a) the minimum of the components (i.e. of hyperrectangles minus the forbidden region)
- (b) the program comprising the possible executions between the minimum of a component, and the minimum of the next
- (c) we use the interleaving semantics for finding *1* path in this program (using 1, in a very economical manner)

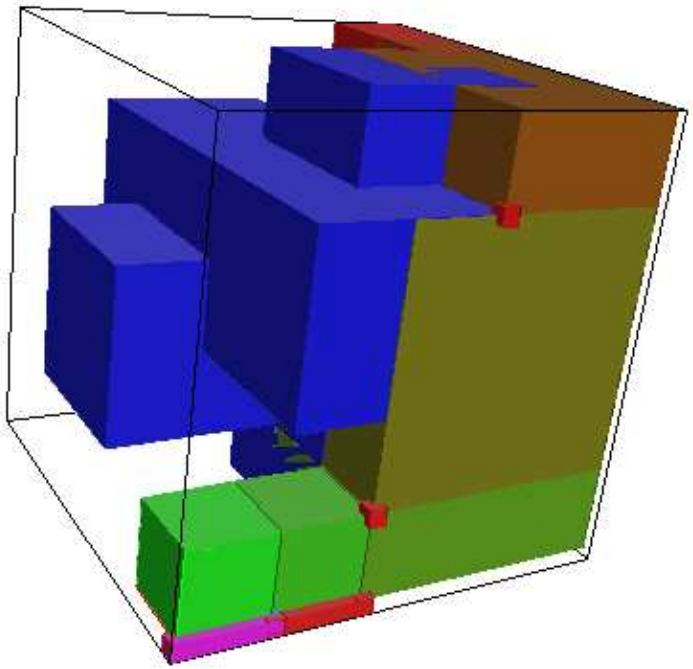
Discretization of paths



`0 | 0 | P(c)`

In context `#sem c 1`, `#sem b 1`, and `#sem a 1`.

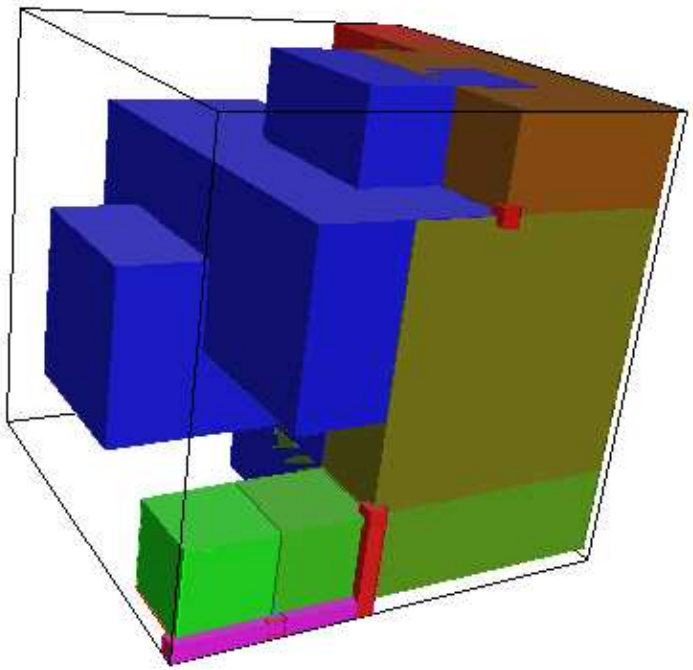
Discretization of paths



`0 | 0 | P(a)`

In context `#sem c 0`, `#sem b 1`, `#sem a 1`.

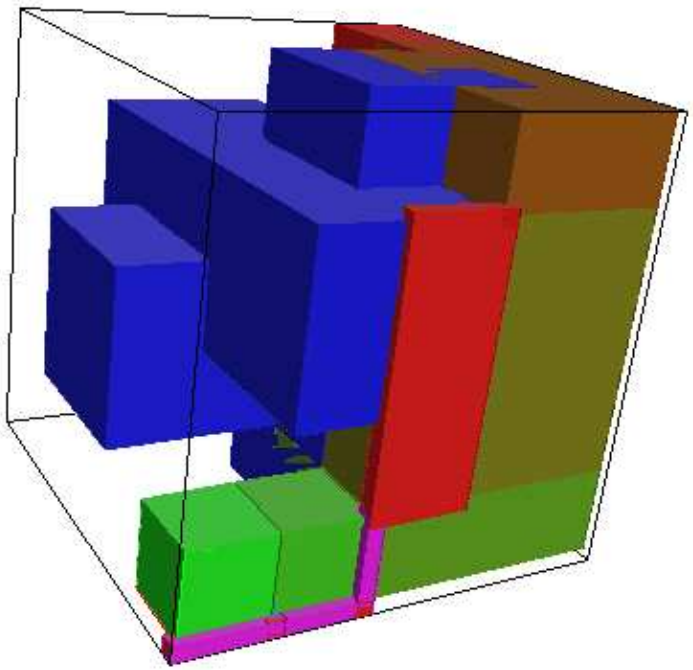
Discretization of paths



`0 | P(b) | 0`

In context `#sem c 0`, `#sem b 1`, `#sem a 0`.

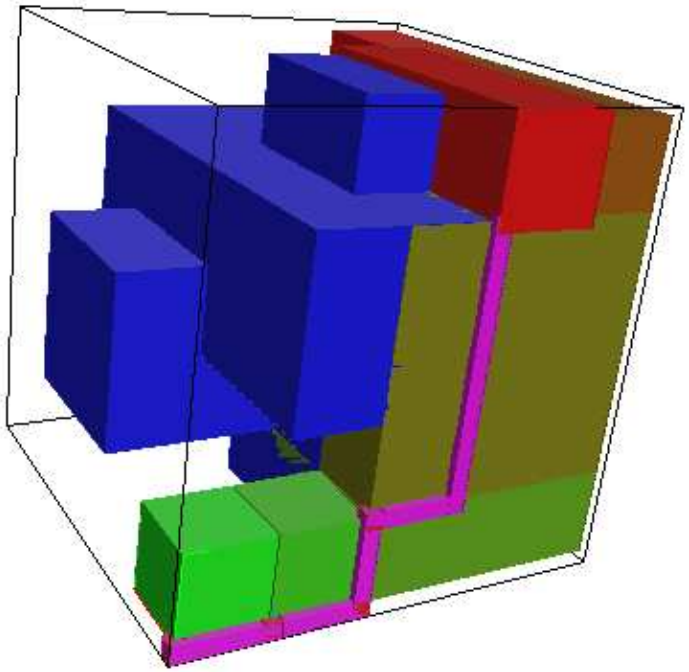
Discretization of paths



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

In context `#sem c 0`, `#sem b 0`, `#sem a 0`.

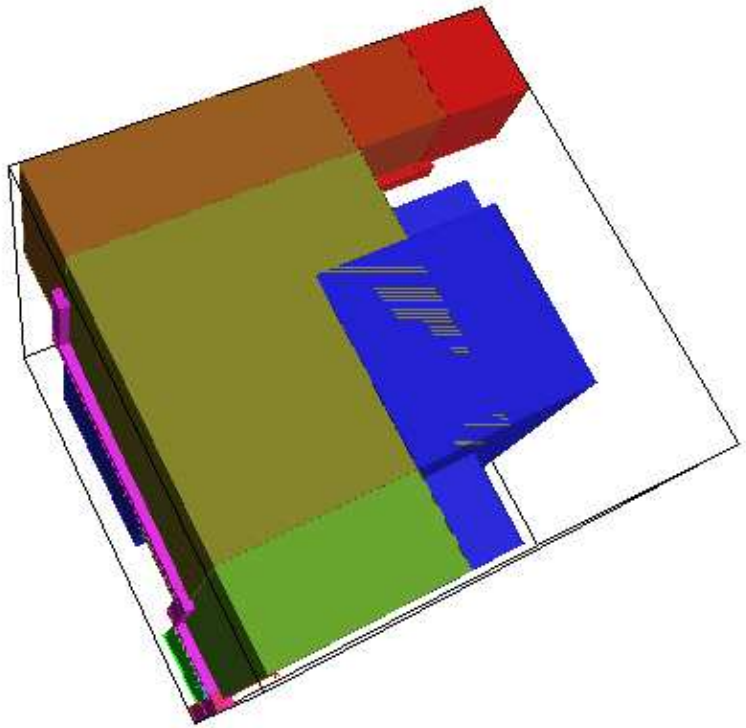
Discretization of paths



`P(a).P(b).V(a) | 0 | V(a)`

In context `#sem c 1`, `#sem b 1`, `#sem a 0`.

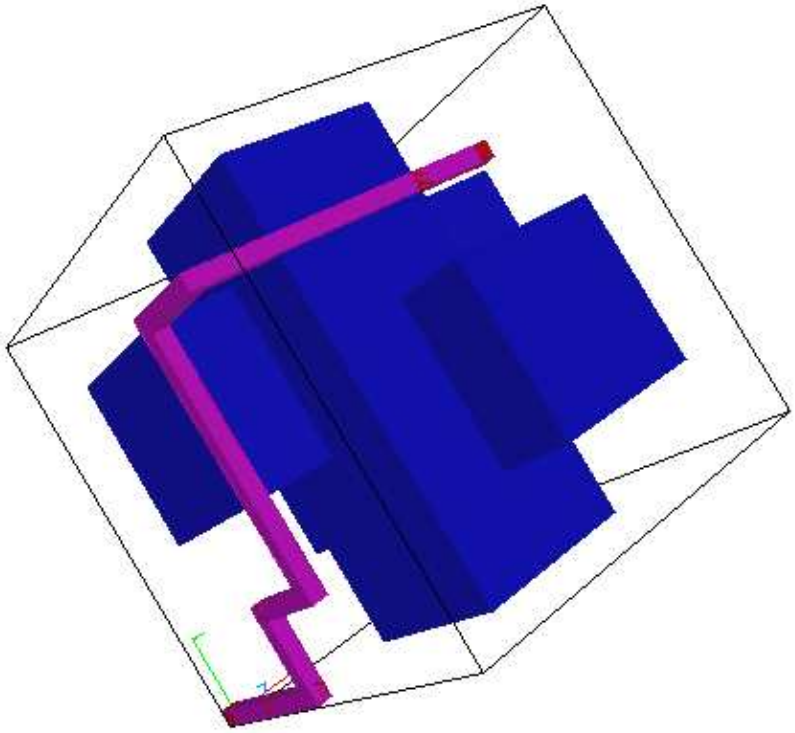
Discretization of paths



`V(b) | 0 | 0`

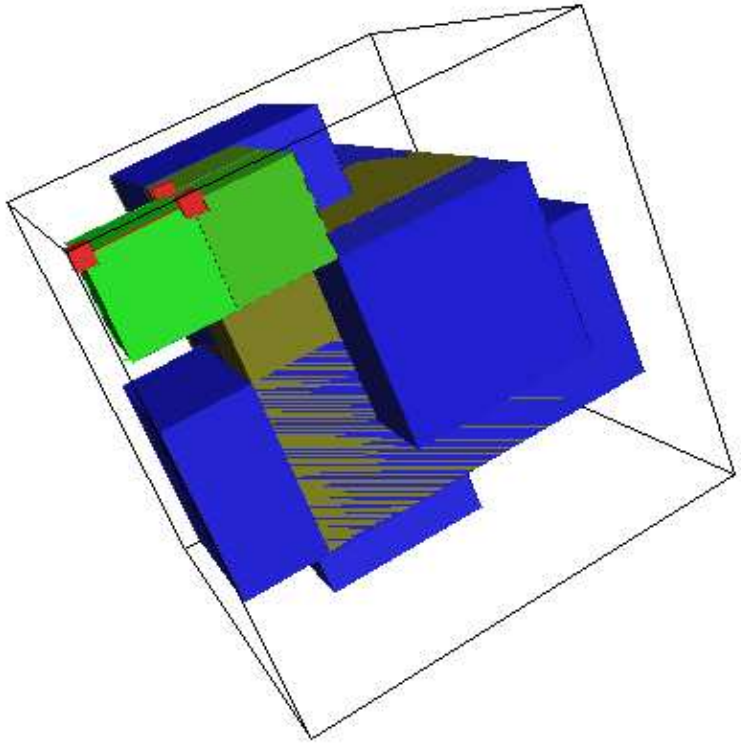
In context `#sem c 1`, `#sem b 0`, `#sem a 1`.

Discretization of paths



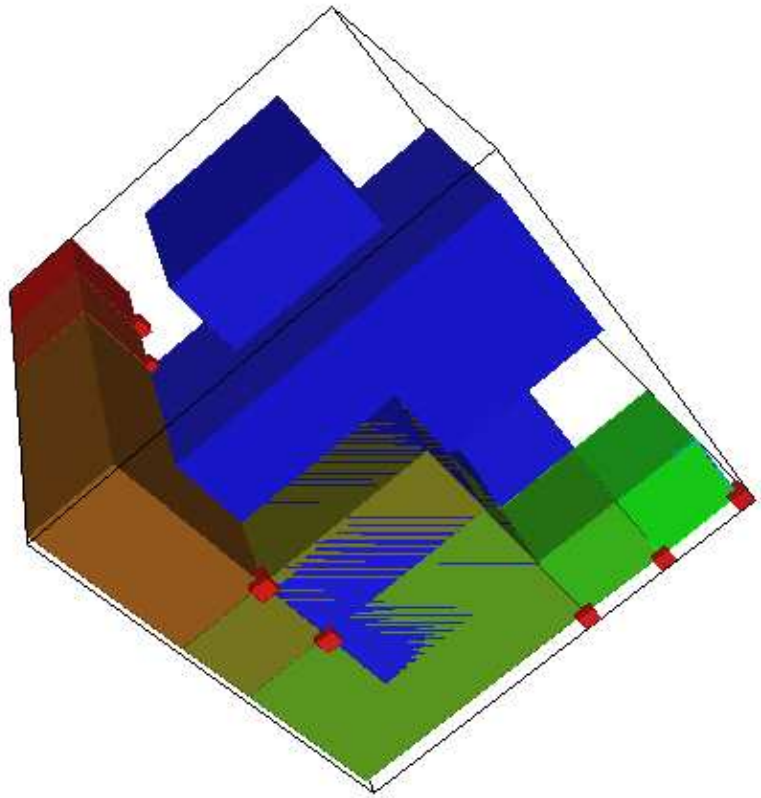
```
{3}P(c);{3}P(a);{2}P(b);{3}V(c);{2}P(c);{2}V(b);
{2}V(c);{3}V(a);{1}P(a);{1}P(b);{1}V(a);{1}V(b);
```

Discretization of paths (2 - deadlock)



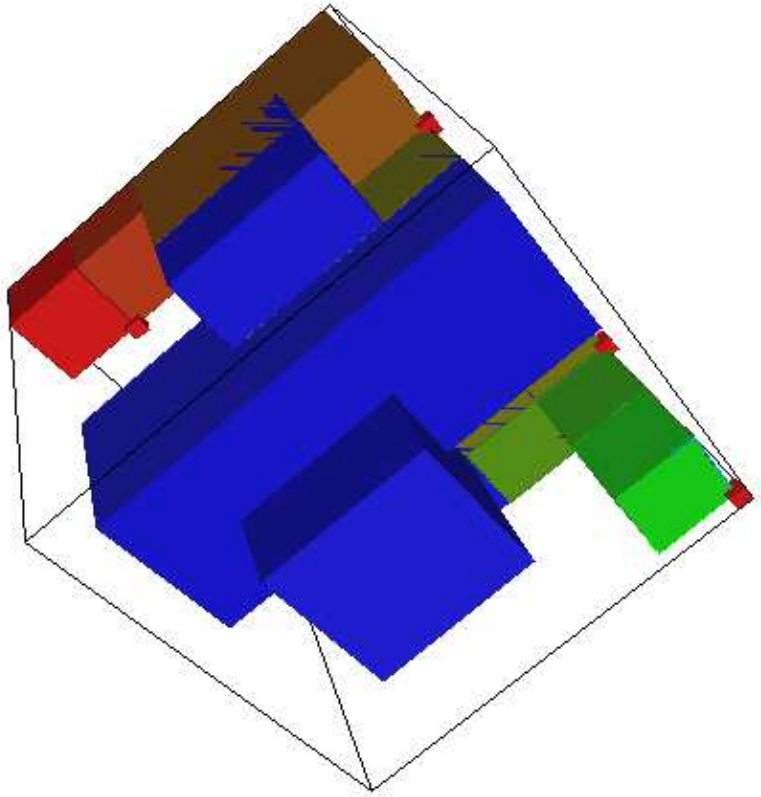
```
{1}P(a);{3}P(c);{2}P(b);
```

Discretization of paths (3)



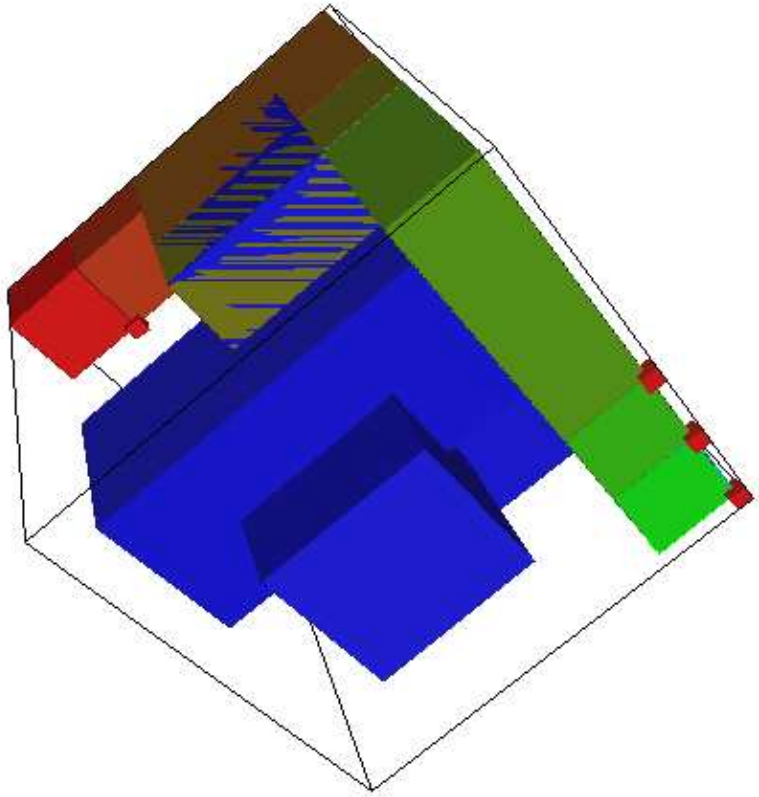
```
{2}P(b);{2}P(c);{2}V(b);{2}V(c);{3}P(c);{3}P(a);
{3}V(c);{3}V(a);{1}P(a);{1}P(b);{1}V(a);{1}V(b);
```

Discretization of paths (4)



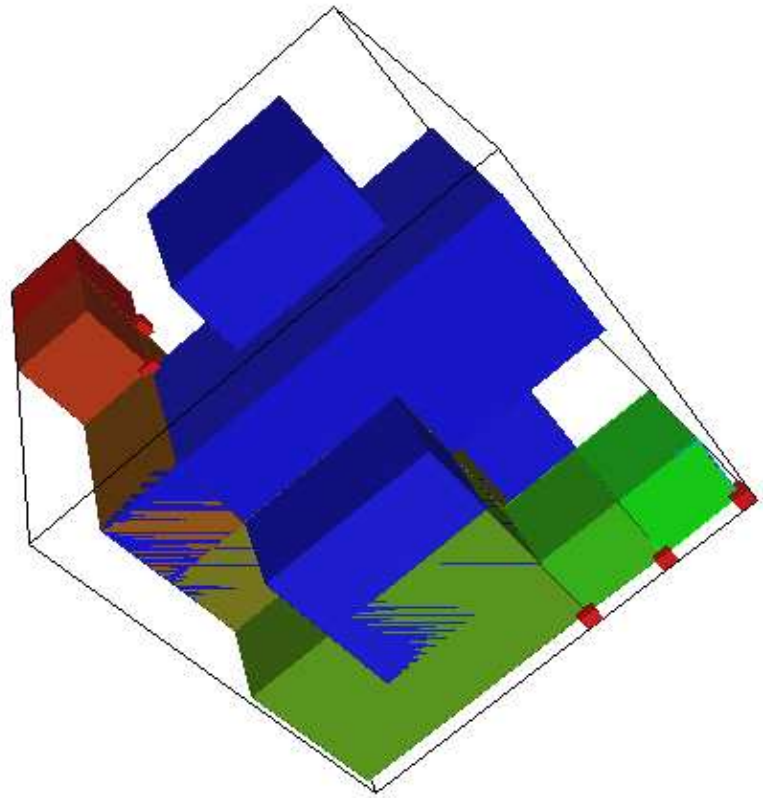
```
{1}P(a);{1}P(b);{3}P(c);{1}V(a);{3}P(a);{3}V(c);
{3}V(a);{1}V(b);{2}P(b);{2}P(c);{2}V(b);{2}V(c);
```

Discretization of paths (5)



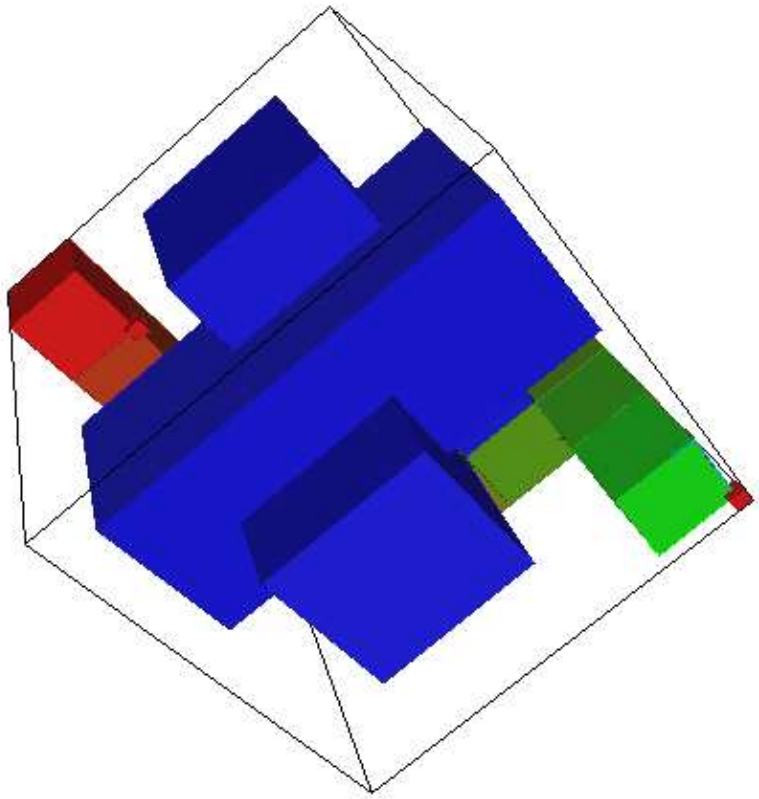
```
{3}P(c);{3}P(a);{3}V(c);{3}V(a);{1}P(a);{1}P(b);
{1}V(a);{1}V(b);{2}P(b);{2}P(c);{2}V(b);{2}V(c);
```

Discretization of paths (6)



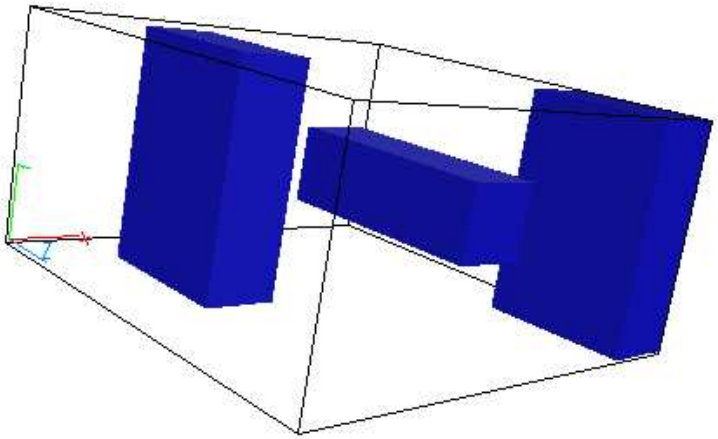
```
{2}P(b);{2}P(c);{1}P(a);{2}V(b);{1}P(b);{1}V(a);
{2}V(c);{3}P(c);{3}P(a);{3}V(c);{3}V(a);{1}V(b);
```

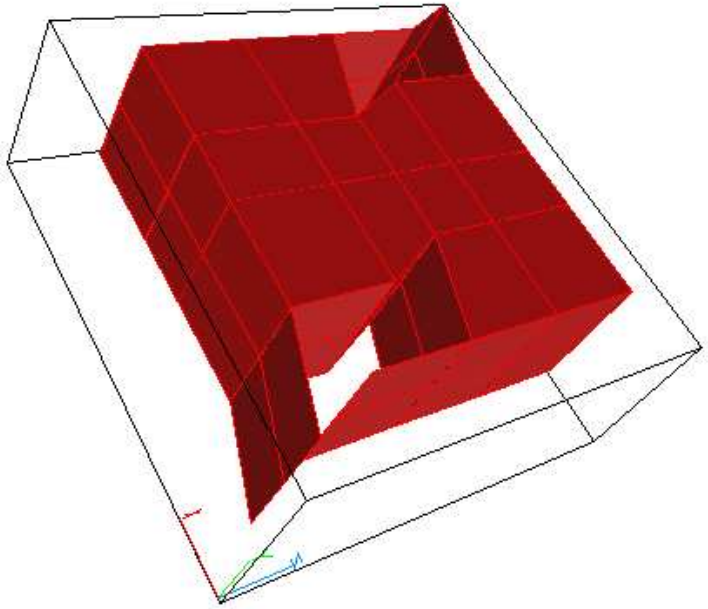
Discretization of paths (7)



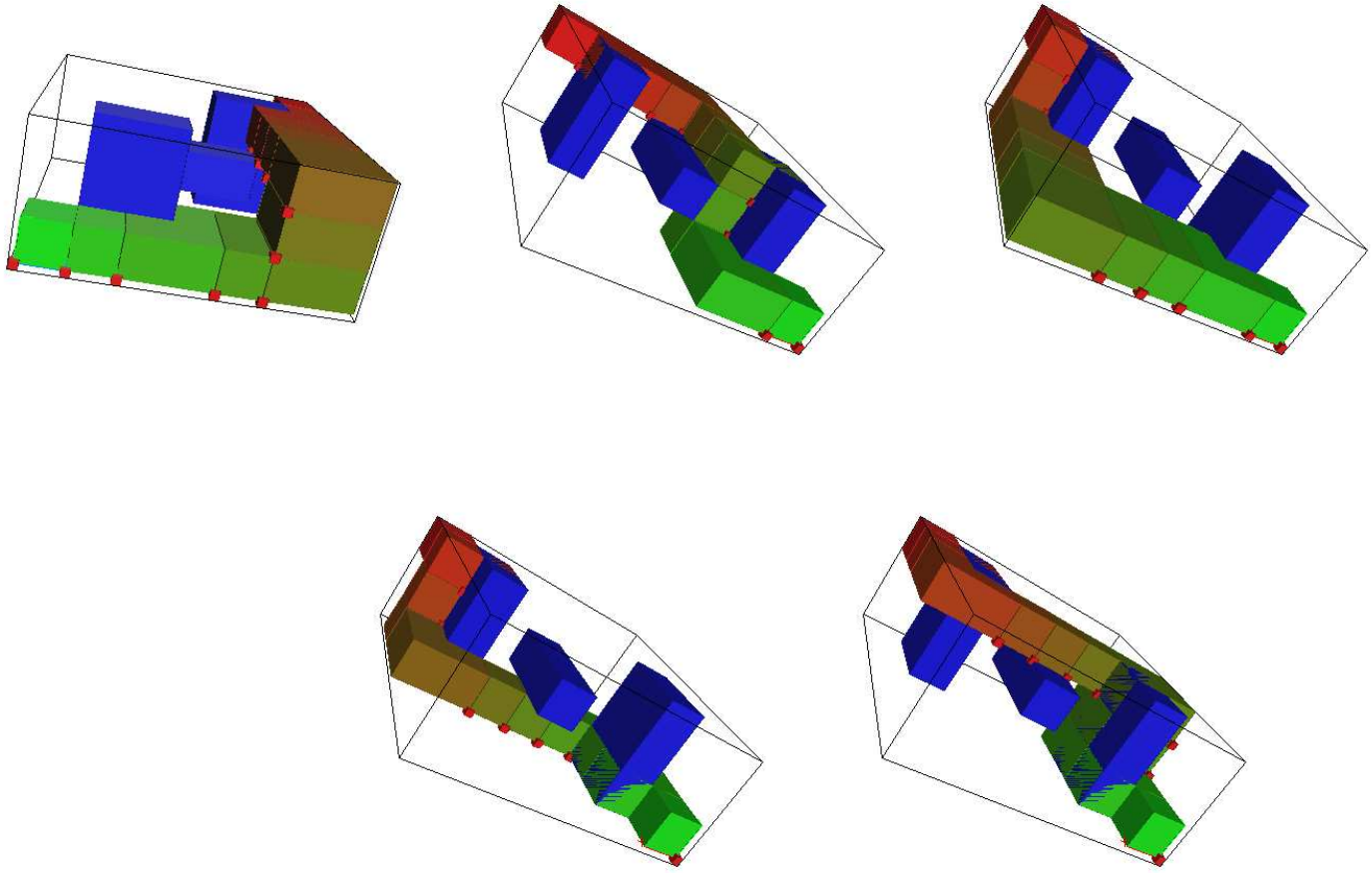
```
{1}P(a);{1}P(b);{1}V(a);{1}V(b);{2}P(b);{2}P(c);
{2}V(b);{2}V(c);{3}P(c);{3}P(a);{3}V(c);{3}V(a);
```

Component category





Discretization of paths



Algorithmic improvements

- Computation of components:
  - still sub-optimal (in size)
  - use of static/dynamic segment trees to improve the computation of intersections
  - (simpler) use of geometric simple geometric constraints to improve the computation of intersections
  - etc.

Algorithmic improvements

- Computation of morphisms in the component category:
  - By computing the first homology group
  - (simpler) approximation of relations (persistent sets)
  - (on the longer run) simplification of the retract by considering also the $\overrightarrow{\pi}_n$ ["homotopical resolution", or "higher-order syzygies"]
- Plus all classical improvements: symbolic terms, symmetry, on the fly traversal, state space hashing etc.

Thanks for your attention...

...Any questions?