Directed Algebraic Topology and Concurrency

Emmanuel Haucourt

MPRI : Concurrency (2.3)

Thursday, the 4^{th} of February 2010

emmanuel.haucourt@cea.fr

Free monoid $W(\mathbb{A})$

words over the alphabet $\mathbb{A}$

- The set $\mathbb{A}$ is called the alphabet, its elements are called the letters
- Given $n \in \mathbb{N}$, a n-word is a finite sequences of letters of length n i.e.

$$w \in \mathcal{S}et\Big[\{0, \ldots, n-1\}, \mathbb{A}\Big]$$

- The elements of $W(\mathbb{A})$ are all the words i.e.

$$\bigcup_{n \in \mathbb{N}} \mathcal{S}et\Big[\{0, \ldots, n-1\}, \mathbb{A}\Big]$$

- The internal law is the concatenation,
  given words w and w' of lengths n and n'

$$w \cdot w' : \; \{0, \ldots, n+m-1\} \longrightarrow \mathbb{A}$$

$$t \longmapsto \begin{cases} w(t) & \text{if } 0 \leqslant t \leqslant n-1 \\ w'(t-n) & \text{if } r \leqslant t \leqslant n+n'-1 \end{cases}$$

- The neutral element is the empty word

The free monoid functor

Remark : if w is a word over the alphabet $\mathbb{A}$ and $f \in \mathcal{S}et[\mathbb{A}, \mathbb{A}']$ then $f \circ w$ is a word over the alphabet $\mathbb{A}'$

$$W : \mathcal{S}et \longrightarrow \mathcal{M}on$$

$$\begin{array}{ccc} \mathbb{A} & & W(\mathbb{A}) \\ \downarrow f & \longmapsto & W(f) \downarrow \\ \mathbb{A}' & & W(\mathbb{A}') \end{array}$$

with

$$W(f) : W(\mathbb{A}) \longrightarrow W(\mathbb{A}')$$
$$w \longmapsto f \circ w$$

Free commutative monoid $C(V)$

Linear combinations with coefficients in $\mathbb{N}$ and variables in V

- Given $\varphi \in \mathcal{S}et[V, \mathbb{N}]$, the support of φ is $\{x \in V \mid \varphi(x) \neq 0\}$
- The elements of $C(V)$ are the linear combinations i.e. the elements of $\mathcal{S}et[V, \mathbb{N}]$ with finite support
- The internal law is the pointwise sum, given polynomials φ and φ'

$$\varphi + \varphi' : \begin{array}{ccc} V & \longrightarrow & \mathbb{N} \\ x & \longmapsto & \varphi(x) + \varphi'(x) \end{array}$$

- The neutral element is the null combination

The free commutative monoid functor $C(-)$

$$\begin{array}{cccc} C(-): & \mathcal{S}et & \longrightarrow & \mathcal{C}mon \\ & V & & C(V) \\ & \Big\downarrow f & \longmapsto & C(f) \Big\downarrow \\ & V' & & C(V') \end{array}$$

with

$$C(f):\ C(V) \longrightarrow C(V')$$

$$\varphi \longmapsto \left\{ \begin{array}{ccc} V' & \longrightarrow & \mathbb{N} \\ x' & \longmapsto & \displaystyle\sum_{\substack{x \in V \\ f(x)=x'}} \varphi(x) \end{array} \right.$$

An example

$V := \{a, b, c\}$ and $V' := \{x, y, z\}$

$\varphi : V \longrightarrow \mathbb{N}$ with $\varphi(a) = 1$, $\varphi(b) = 2$, $\varphi(c) = 3$

The element $\varphi \in C(V)$ can be denoted as a linear combination $a + 2b + 3c$

Consider $f : V \longrightarrow V'$ with $f(a) = f(b) = x$ and $f(c) = z$ then

$$C(f)(\varphi) = C(f)(a + 2b + 3c) = f(a) + 2f(b) + 3f(c) = x + 2x + 3z = 3x + 3z$$

i.e. the mapping

$C(f)(\varphi) : V' \longrightarrow \mathbb{N}$ with

$$C(f)(\varphi)(x) = \varphi(a) + \varphi(b) = 3,\ C(f)(\varphi)(y) = 0,\ C(f)(\varphi)(z) = \varphi(c) = 3$$

Assumption

From now on, all the monoids we consider are supposed to be commutative

unless otherwise stated

Divisibility relation

in a commutative monoid $(M, *, e)$

Given $a, b \in M$ by $a|b$ we mean there exists $q \in M$ s.t. $b = a * q$

The divisibility relation $|$ is a preorder

Prime *vs* Irreducible

Let $(M, *, e)$ be a commutative monoid

$u \in M$ is said to be a unit when there exists $x \in M$ such that $u * x = e$

$p \in M$ is said to be prime when p is not a unit and
for all $a, b \in M$, $p|(a * b) \Rightarrow p|a$ or $p|b$

$i \in M$ is said to be irreducible when for all $a, b \in M$,
if $i = a * b$ then either a or b is a unit (not both)

Prime *vs* Irreducible

Examples

- Denote by $\mathbb{N}[X]$ the collection of one indeterminate polynomials over $\mathbb{N}$ we have $1+X+X^2+X^3+X^4+X^5=(1+X^3)(1+X+X^2)=(1+X)(1+X^2+X^4)$

- $1+X$ is irreducible and not prime since $1+X$ does not divide $1+X^3$ in $\mathbb{N}[X]$

The preceding example is due to *Junji Hashimoto*

- In the monoid $(\{0,1\},\vee,0)$, the element 1 is prime but not irreducible
- In the monoid $(\mathbb{R}_+,+,0)$ there is neither prime element nor irreducible one
- An element φ of the free commutative monoid $C(V)$ is prime iff it is irreducible iff its support is a singleton $\{v\}$ and $\varphi(v)=1$ iff

$$\int_V \varphi := \sum_{v\in V} \varphi(v) = 1$$

Characterization
of the free commutative monoids

Given a commutative monoid M, the following are equivalent

- M is free (i.e. $M \cong C(V)$ for some set V)
- $M \cong C(\mathcal{P})$ with $\mathcal{P}$ the set of prime elements of M
- $M \cong C(\mathcal{I})$ with $\mathcal{I}$ the set of irreducible elements of M
- for all $x \in M$, x is irreducible iff x is prime
  and any element of M is a product of irreducible/prime elements
- any element of M can be written as a product of irreducible elements of M in a unique way (up to permutation)
- any element of M can be written as a product of prime elements of M in a unique way (up to permutation)

The commutative monoid
of isomorphism classes of small categories

- We write $\mathcal{A} \cong \mathcal{B}$ to mean that $\mathcal{A}$ and $\mathcal{B}$ are isomorphic in $\mathcal{C}at$
- The relation $\cong$ is an equivalence relation
- We denote the isomorphism class of $\mathcal{C}$ by $[\mathcal{C}]$
- If $\mathcal{A} \cong \mathcal{A}'$ and $\mathcal{B} \cong \mathcal{B}'$ then $\mathcal{A} \times \mathcal{B} \cong \mathcal{A}' \times \mathcal{B}'$
  so we can define
$$[\mathcal{A}] \times [\mathcal{B}] := [\mathcal{A} \times \mathcal{B}]$$
- Since $\mathcal{A} \times \mathcal{B} \cong \mathcal{B} \times \mathcal{A}$ we have $[\mathcal{A}] \times [\mathcal{B}] = [\mathcal{B}] \times [\mathcal{A}]$
- If we denote the category with one object and one morphism by $\mathbf{1}$ then
$$[\mathbf{1}] \times [\mathcal{A}] = [\mathcal{A}] \times [\mathbf{1}] = [\mathcal{A}]$$
- Hence the collection of (isomorphism classes of)[1] small categories forms a commutative monoid

[1] in the sequel we identify a small category with its isomorphism class, therefore omit to write "isomorphism classes of"

Size of an isomorphism class

- The size of small category $\mathcal{A}$ is defined as the cardinal of the set $\mathrm{Mo}(\mathcal{A})$
- Given small categories $\mathcal{A}$ and $\mathcal{B}$ we have

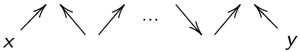
$$\mathit{size}(\mathcal{A} \times \mathcal{B}) := \mathit{size}(\mathcal{A}) \times \mathit{size}(\mathcal{B})$$

- If $\mathcal{A} \cong \mathcal{B}$ then $\mathit{size}(\mathcal{A}) = \mathit{size}(\mathcal{B})$ so we can define

$$\mathit{size}\big([\mathcal{A}]\big) := \mathit{size}(\mathcal{A})$$

Connected categories

- Given two objects x and y of a category $\mathcal{C}$, write $x \leftrightarrow y$ when there exists a zigzag of morphisms between x and y i.e.



- The relation $\leftrightarrow$ is a preorder
- A category $\mathcal{C}$ is said to be connected when the preorder $\leftrightarrow$ is chaotic i.e. for all objects x and y we have $x \leftrightarrow y$

Loop-free categories

This notion has been introduced by *André Haefliger*

A category $\mathcal{C}$ is said to be loop-free when for all objects x and y

$\mathcal{C}[x,y] \neq \emptyset$ and $\mathcal{C}[y,x] \neq \emptyset$ implies $x = y$ and $\mathcal{C}[x,x] = \{\mathrm{id}_x\}$

The fundamental category of any pospace is loop-free

Some properties
preserved under isomorphisms

Let $\mathcal{A}$ and $\mathcal{B}$ be isomorphic categories

- $\mathcal{A}$ is finite iff so is $\mathcal{B}$
- $\mathcal{A}$ is loop-free iff so is $\mathcal{B}$
- $\mathcal{A}$ is connected iff so is $\mathcal{B}$

So we can say that an isomorphism class of categories is
finite/loop-free/connected
when any of its representative is so

Some properties

preserved and reflected by Cartesian product

Let $\mathcal{A}$ and $\mathcal{B}$ be non empty categories

- $\mathcal{A} \times \mathcal{B}$ is finite iff $\mathcal{A}$ and $\mathcal{B}$ are so
- $\mathcal{A} \times \mathcal{B}$ is loop-free iff $\mathcal{A}$ and $\mathcal{B}$ are so
- $\mathcal{A} \times \mathcal{B}$ is connected iff $\mathcal{A}$ and $\mathcal{B}$ are so

The monoid $\mathbb{M}$

The collection of non-empty connected loop-free finite categories forms a sub-monoid $\mathbb{M}$ of the monoid of small categories

$\mathbb{M}$ is pure[2] which means that for all small categories $\mathcal{A}$ and $\mathcal{B}$, if $[\mathcal{A}] \times [\mathcal{B}] \in \mathbb{M}$ then $[\mathcal{A}] \in \mathbb{M}$ and $[\mathcal{B}] \in \mathbb{M}$

The size function induces a morphism of monoids from $\mathbb{M}$ to $(\mathbb{N}\backslash\{0\}, \times, 1)$

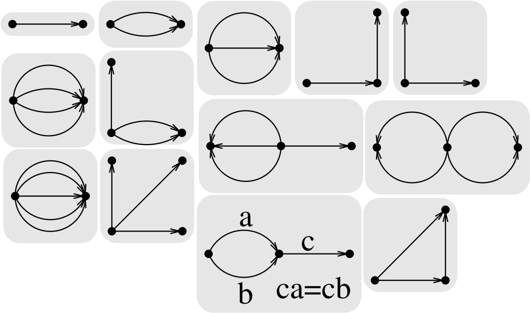
Theorem

$\mathbb{M}$ is a free commutative monoid.
The set of prime/irreducible elements of $\mathbb{M}$ is countable and infinite.

[2] in the monoid of small categories

Prime elements of $\mathbb{M}$

of size at most 7 (up to opposite)



The motivating example

```
#mutex a b

p = P(a).V(a)
q = P(b).V(b)

init: p q p
```

```
  [0,1[*[0,-[*[0,-[
| [2,-[*[0,-[*[0,-[
| [0,-[*[0,-[*[0,1[
| [0,-[*[0,-[*[2,-[
```

```
#mutex a b

p = P(a).V(a)
q = P(b).V(b)

init: p p q
```

```
  [0,1[*[0,-[*[0,-[
| [2,-[*[0,-[*[0,-[
| [0,-[*[0,1[*[0,-[
| [0,-[*[2,-[*[0,-[
```

```
([0,1[*[0,-[ | [2,-[*[0,-[ | [0,-[*[0,1[ | [0,-[*[2,-[)*[0,-[
```

Semi-lattice of Intervals of $\mathbb{R}^+$

$\emptyset$ empty interval

$\{a\}$	singleton	for $a \in \mathbb{R}_+$
$[a, +\infty[$	closed unbounded	
$]a, +\infty[$	open unbounded	
$[a, b]$	closed bounded (compact)	for $a, b \in \mathbb{R}_+$ and $a < b$
$]a, b[$	open bounded	
$[a, b[$	half-open of the right bounded	
$]a, b]$	half-open of the left bounded	

This collection forms a semi-lattice with $\bigcap$ as product and $[0, +\infty[$ as neutral element

Semi-lattice of cubes

of dimension $n \in \mathbb{N}$

Case $n = 0$: semi-lattice $(\{0,1\}, \wedge, 1)$
Case $n \neq 0$: semi-lattice of Cartesian products

$$\prod_{k=1}^{n} \mathbb{I}_k$$

where $\mathbb{I}_k$ is an interval for all k in $\{1, \ldots, n\}$.

$$\Big(\prod_{k=1}^{n} \mathbb{I}_k\Big) \cap \Big(\prod_{k=1}^{n} \mathbb{I}'_k\Big) = \prod_{k=1}^{n} (\mathbb{I}_k \cap \mathbb{I}'_k)$$

This collection forms a semi-lattice with $\bigcap$ as product and $[0, +\infty[^n$ as neutral element

$$\Big(\prod_{k=1}^{n} \mathbb{I}_k\Big) \times \Big(\prod_{k=n+1}^{n+p} \mathbb{I}_k\Big) = \prod_{k=1}^{n+p} \mathbb{I}_k$$

Boolean algebra of cubical areas

of dimension $n \in \mathbb{N}$

Cas $n = 0$: Boolean algebra $\{0, 1\}$
Cas $n \neq 0$: Boolean algebra of sets $X \subseteq \mathbb{R}_+^n$ which can be written as

$$\bigcup_{i=1}^{p} C_i$$

where $p \in \mathbb{N}$ and for all i in $\{1, \ldots, p\}$, the cube C_i is n-dimensional.

$$\Big(\bigcup_{i=1}^{p} C_i\Big) \cap \Big(\bigcup_{j=1}^{p'} C'_j\Big) = \bigcup_{i=1}^{p}\bigcup_{j=1}^{p'} (C_i \cap C'_j)$$

$$\Big(\bigcup_{i=1}^{p} C_i\Big)^c = \bigcap_{i=1}^{p} C_i^c$$

$$C_i^c = \Big(\prod_{k=1}^{n} \mathbb{I}_k\Big)^c = \bigcup_{k=1}^{n} \underbrace{\mathbb{R}^+ \times \cdots \times \mathbb{R}^+}_{k-1 \text{ times}} \times \mathbb{I}_k^c \times \underbrace{\mathbb{R}^+ \times \cdots \times \mathbb{R}^+}_{n-k \text{ times}}$$

$$\Big(\bigcup_{i=1}^{p} C_i\Big) \times \Big(\bigcup_{j=1}^{p'} C'_j\Big) = \bigcup_{i=1}^{p}\bigcup_{j=1}^{p'} (C_i \times C'_j)$$

Semi-lattice of cubical coverings

of dimension $n \in \mathbb{N}$

A cubical covering $\mathcal{C}$ of dimension n is a finite set of n-dimensional cubes

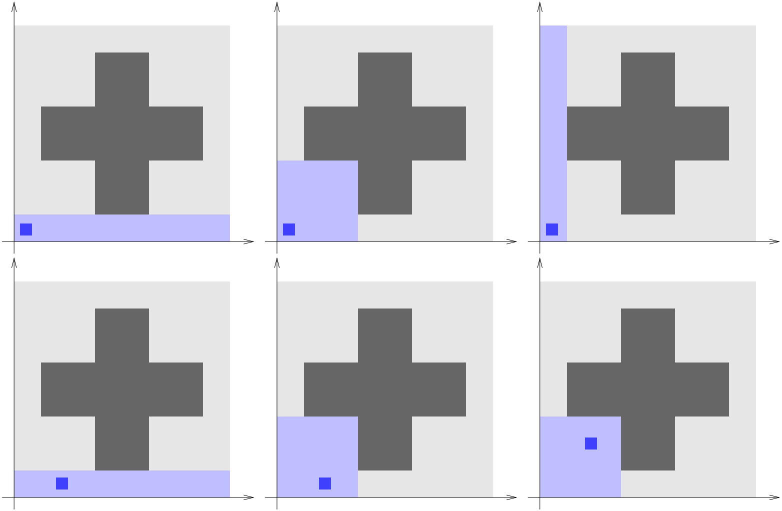
$$\mathcal{C} \sqsubseteq \mathcal{C}' \quad \text{iff} \quad \forall x \in \mathcal{C}\ \exists x' \in \mathcal{C}' \text{ s.t. } x \subseteq x'$$

$$\mathcal{C} \wedge \mathcal{C}' = \left\{ x \cap x' \mid x \in \mathcal{C}; x' \in \mathcal{C}' \right\}$$

$$\mathcal{C} \times \mathcal{C}' = \left\{ x \times x' \mid x \in \mathcal{C}; x' \in \mathcal{C}' \right\}$$

CPO of sub-cubes of a cubical area

Maximal sub-cubes of a cubical area



A cubical area is the union of its maximal sub-cubes

Cubique areas *vs* Cubique coverings

A *Galois* connection

$$\{\text{Cubical coverings}\} \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \{\text{Cubical areas}\}$$

$$\alpha(\mathcal{C}) := \bigcup_{x \in \mathcal{C}} x$$

$$\gamma(X) := \{\text{maximal sub-cubes of } X\}$$

$$\alpha \circ \gamma = \text{id} \qquad \text{and} \qquad \text{id} \sqsubseteq \gamma \circ \alpha$$

If the cubical coverings $\mathcal{C}_1$ and $\mathcal{C}_2$ contain all the maximal sub-cubes of $\alpha(\mathcal{C}_1)$ and $\alpha(\mathcal{C}_2)$, then $\mathcal{C}_1 \wedge \mathcal{C}_2$ contains all the maximal sub-cubes of $\alpha(\mathcal{C}_1) \cap \alpha(\mathcal{C}_2)$

Implementation of the graded Boolean structure

of the cubical areas of all dimensions

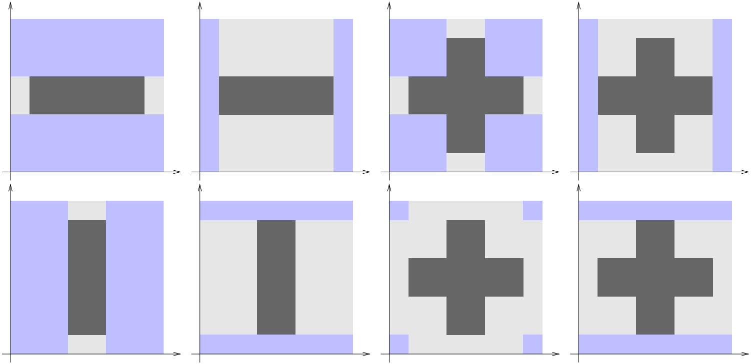
The Boolean algebra of cubical areas is isomorphic to collection of cubical coverings whose elements are maximal sub-cubes of the area it covers. Concretely, the non-empty n-cubes are words of non-empty intervals of length n. If C and C' are two non-empty cubes of dimension n and m, then their Cartesian product $C \times C'$ is given by the concatenation of words of intervals. Since we gather all the Boolean algebras of n-cubical areas in a single graded one, we need to pay some attention to the empty sets! Indeed, the empty set $\emptyset_n$ of dimension n differs from the empty set $\emptyset_m$ of dimension m as soon as $n \neq m$ since their complements (respectively $\mathbb{R}_+^n$ and $\mathbb{R}_+^m$) do. In particular if C is a m-cube, then $\emptyset_n \times C = \emptyset_{n+m}$.

Yet, recall that the Boolean algebra of 0-dimensional cubical areas is $\{0, 1\}$. Then 1 is the neutral element of the Cartesian product, this fact comes naturally if we represent it by the singleton whose unique element is the empty word $\{()\}$.

This product obviously extends to cubical area which are represented by sets of cubes. Intersection and Cartesian product are easily computed. The union requires we apply the operator $\gamma \circ \alpha$: if $\mathcal{C}$ and $\mathcal{C}'$ represent the cubical areas X and X', then $X \cup X'$ is represented by $\gamma \circ \alpha(\mathcal{C} \cup \mathcal{C}')$.

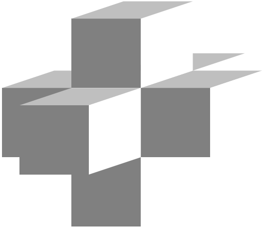
Complement of a cubical area

a planar example

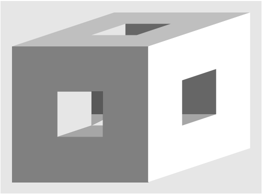


Complement of a cubical area

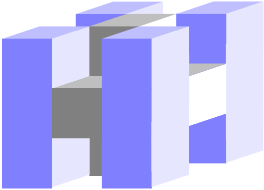
a spatial example



forbidden area



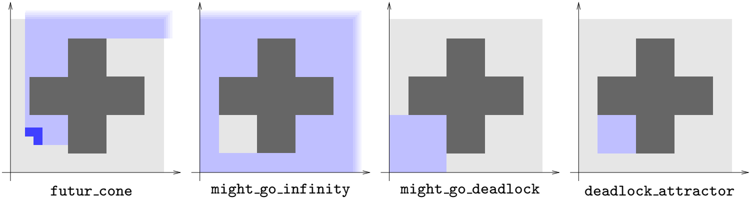
state space



the four “vertical” maximal cubes

Examples

of functions implemented in the OCaml library `area.ml`



Graded action

of the symmetrical groups $\mathfrak{S}_n$ for $n \in \mathbb{N}$ over cubical algebra

- Given an n-cube $x = \mathbb{I}_1 \times \cdots \times \mathbb{I}_n$ and a permutation $\sigma \in \mathfrak{S}_n$ we define $\sigma \cdot x = \sigma \cdot (\mathbb{I}_1 \times \cdots \times \mathbb{I}_n) := \mathbb{I}_{\sigma(1)} \times \cdots \times \mathbb{I}_{\sigma(n)}$
- The preceding definition extends to cubical covering $\sigma \cdot \mathcal{C} := \{\sigma \cdot x \mid x \in \mathcal{C}\}$
- If $\mathcal{C} \sqsubseteq \mathcal{C}'$ then $\sigma \cdot \mathcal{C} \sqsubseteq \sigma \cdot \mathcal{C}'$
- Given two cubical coverings $\mathcal{C}_1$ and $\mathcal{C}_2$, if $\alpha(\mathcal{C}_1) = \alpha(\mathcal{C}_2)$ then $\alpha(\sigma \cdot \mathcal{C}_1) = \alpha(\sigma \cdot \mathcal{C}_2)$ therefore we can define $\sigma \cdot X = \alpha(\sigma \cdot \mathcal{C})$ where $\mathcal{C}$ is any cubical covering such that $\alpha(\mathcal{C}) = X$

The monoid of cubical areas

- We identify each cubical area X with is set of maximal sub-cubes since $\gamma(X \times Y) = \gamma(X) \times \gamma(Y)$
- The non-empty cubical areas with Cartesian product forms a free monoid (it is not commutative)

The commutative monoid of cubical areas

- Given n-cubical areas X and Y, write $X \sim Y$ when there exists $\sigma \in \mathfrak{S}_n$ s.t. $\sigma \cdot X = Y$
- $\sim$ is a congruence over the monoid of cubical areas i.e. $\sim$ is an equivalence relation and $X \sim X'$ and $Y \sim Y'$ implies $X \times X' \sim Y \times Y$
- The quotient of the monoid of cubical areas by $\sim$ is commutative free

The motivating example

```
#mtx a b
#sem c 3
pa = P(a).P(c).V(c).V(a)
pb = P(b).P(c).V(c).V(b)
init:  pa pb pa pb
```

```
  [0,1[*[0,1[*[0,-[*[0,-[
| [0,1[*[4,-[*[0,-[*[0,-[
| [0,1[*[0,-[*[0,-[*[0,1[
| [0,1[*[0,-[*[0,-[*[4,-[
| [4,-[*[0,1[*[0,-[*[0,-[
| [4,-[*[4,-[*[0,-[*[0,-[
| [4,-[*[0,-[*[0,-[*[0,1[
| [4,-[*[0,-[*[0,-[*[4,-[
| [0,-[*[0,1[*[0,1[*[0,-[
| [0,-[*[0,1[*[4,-[*[0,-[
| [0,-[*[4,-[*[0,1[*[0,-[
| [0,-[*[4,-[*[4,-[*[0,-[
| [0,-[*[0,-[*[0,1[*[0,1[
| [0,-[*[0,-[*[0,1[*[4,-[
| [0,-[*[0,-[*[4,-[*[0,1[
| [0,-[*[0,-[*[4,-[*[4,-[
```

The motivating example

```
#mtx a b
#sem c 3
pa = P(a).P(c).V(c).V(a)
pb = P(b).P(c).V(c).V(b)
init:  pa pa pb pb
```

```
  [0,1[*[0,-[*[0,1[*[0,-[
| [0,1[*[0,-[*[4,-[*[0,-[
| [0,1[*[0,-[*[0,-[*[0,1[
| [0,1[*[0,-[*[0,-[*[4,-[
| [4,-[*[0,-[*[0,1[*[0,-[
| [4,-[*[0,-[*[4,-[*[0,-[
| [4,-[*[0,-[*[0,-[*[0,1[
| [4,-[*[0,-[*[0,-[*[4,-[
| [0,-[*[0,1[*[0,1[*[0,-[
| [0,-[*[0,1[*[4,-[*[0,-[
| [0,-[*[0,1[*[0,-[*[0,1[
| [0,-[*[0,1[*[0,-[*[4,-[
| [0,-[*[4,-[*[0,1[*[0,-[
| [0,-[*[4,-[*[4,-[*[0,-[
| [0,-[*[4,-[*[0,-[*[0,1[
| [0,-[*[4,-[*[0,-[*[4,-[
```