Factorization of cubical areas

Nicolas Ninin

LIX, Ecole polytechnique (France)

March 7, 2017

General context of this work

- **Static analysis** of **concurrent programs**
- Using the tools of **directed** algebraic topology

What is static analysis ?

One definition

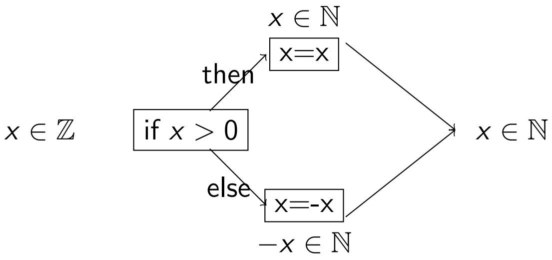
Definition (static analysis)

Static analysis of a program is the analysis of the code of the programs **without actually executing the code**

- It's undecidable in general
- We can try to give approximation to many problems

What is static analysis ?
A baby example

Let P be the following program (absolute function):
Input $x \in \mathbb{Z}$ P:= if $x > 0$ then return x else return -x



Concurrent Programs

A definition

Definition

A concurrent program is a program composed of many sub-process which can use and modify the same ressources. Each process runs at his own pace.

- We write $P = P_1|...|P_n$ for a program with n processus
- Ressources will mean memory for us
- The scheduling of the process is unknown

Concurrent Programs

An example

We consider a program $P = P_1 | P_2$ with:
P_1: $x = 1$; $y = 1x$;
P_2: $y = 2$; $x = 2$;
P_1 and P_2 both shares the ressources x and y.

Different Scheduling

- $P_1\, x = 1;\ y = 1;\ P_2\ y = 2;\ x = 2;$ Output $x = y = 2$
- $P_1\, x = 1;\ P_2\, y = 2;\ P_1 : y = 1;\ P_2 : x = 2$ Output $x = 2\, y = 1$
- P_2 then P_1 Ouput $x = y = 1$
- $P_2 y = 1\ P_1 x = 2; y = 2x\ P_2 x = y$ Output $x = 2, y = 1$

Static analysis of concurrent programs

The problem

The static analysis of a concurrent program is difficult because of the exponential number of executions traces.

In the previous example there were six differents schedulings with sometimes different results. We have to analyze each one of those traces.

Decomposition of programs

concept of decomposition

Definition

We say that P_1 is independent of P_2 if the two process dont interfers with each other.

In that case the analysis of P is reduced to the analysis of P_1 and P_2 separately.
For instance: $P_1 : x = 0$ and $P_2 : y = 0$ are independent since they dont share any variables.

The goal of this talk

decomposition algorithm

key concept

Prior to a complete analysis, trying to decompose a concurrent program into indenpendent parts helps greatly to reduce the complexity.

The goal

In this talk, we are going to present a **decomposition algorithm**.

1. Context
   - static analysis
   - Concurrent Programs

2. Geometric model of concurrency
   - Semaphore and *PV* language
   - Geometric model
   - Cubical Areas

3. Factorization
   - Factorization of cubical areas
   - syntactic algorithm
   - Geometric algorithm

Semaphore and mutex

A locking mechanism

Definition (Semaphore)

A semaphore a of arity $n \geq 1$ is a special ressource that can be taken or released by at most n process at the same time.
There is two special instructions to use them:

- Pa means that the ressource is taken
- Va means that the ressource is released

Thus semaphores act as a **locking** mechanism.

Definition (mutex)

A mutex (MUTual EXclusion) is a semaphore of arity 1 (only one process can take it)

PV programs

Definition

A PV program is a concurrent program where the only allowed instructions are P and V. In particular :

- Our programs won't have loops(for, while) nor branching (if)
- We forget about the other instructions (ie $x = 0$) to abstract the concurrency

Example (back to our first concurrent program)

P_1: $x = 1$; $y = 1$; becomes P_a; $x = 1$; V_a; P_b; $y = 1$; V_b;
P_2: $y = 2$; $x = 2$; becomes P_b; $y = 2$; V_b; P_a; $x = 2$; V_a;
Then we only keeps the P and V

$$P_1 : P_a.V_a.P_b.V_b$$
$$P_2 : P_b.V_b.P_a.V_a$$

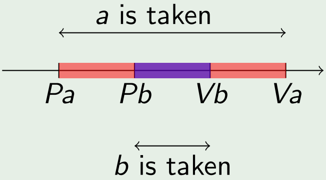
Geometric model of a *PV* program

axis of a process

For a each process we associate a copy of $\mathbb{R}$ where we pin the *PV* instructions.

Example

$$Pa.Pb.Vb.Va$$



The *PV* instructions have to be in the same order on the axis as in the process. Distances dont matter .

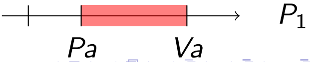
Geometric model of a *PV* program
the construction

Definition

The geometric model $[P]$ of a *PV* program P with n process is a subset of $\mathbb{R}^n$ where we remove areas where the semaphore are taken too many times.

$$Pa.Va|Pa.Va$$

- Axis represent process



Geometric model of a *PV* program

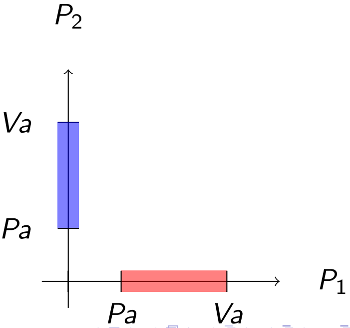
the construction

Definition

The geometric model $[P]$ of a PV program P with n process is a subset of $\mathbb{R}^n$ where we remove areas where the semaphore are taken too many times.

$$Pa.Va|Pa.Va$$

- Axis represent process
- Points in space are states of the program



Geometric model of a PV program

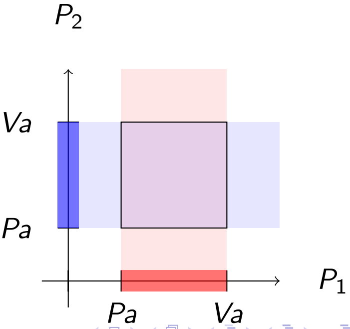
the construction

Definition

The geometric model $[P]$ of a PV program P with n process is a subset of $\mathbb{R}^n$ where we remove areas where the semaphore are taken too many times.

$$Pa.Va|Pa.Va$$

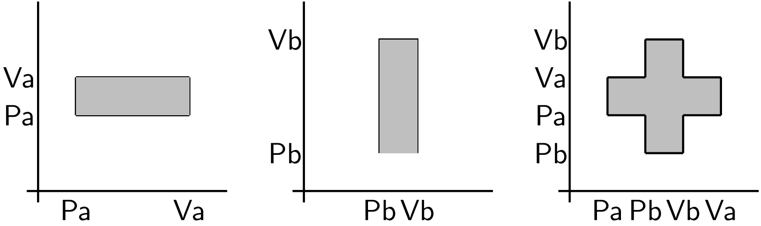
- Axis represent process
- Points in space are states of the program
- Points where semaphore are taken too many times are removed



Geometric model

swiss-cross

$$Pa.Pb.Vb.Va|Pb.Pa.Vb.Va$$



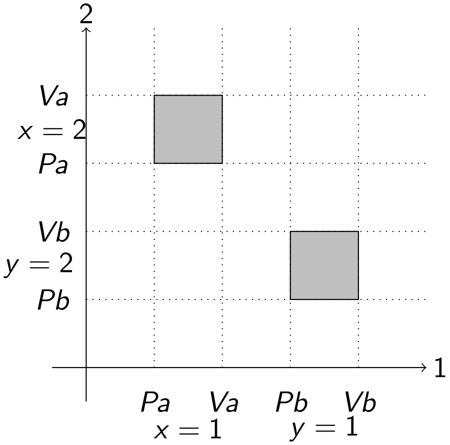
Notice the deadlock situation when P_1 took a and P_2 took b

Geometric model

back to our first concurrent program: directed homotopy

Notion of **directed** homotopy

- directed paths correspond to execution traces
- path that can be continuously deformed to another are equivalent

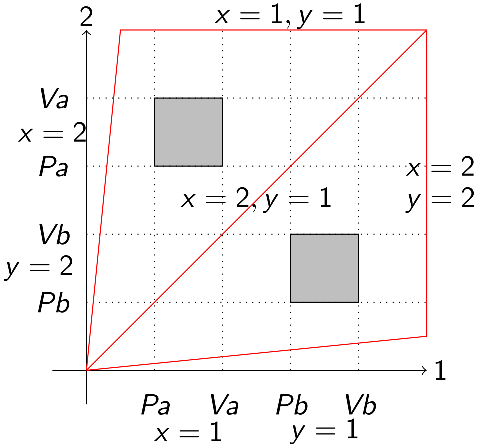


Geometric model

back to our first concurrent program: directed homotopy

Notion of **directed** homotopy

- directed paths correspond to execution traces
- path that can be continuously deformed to another are equivalent

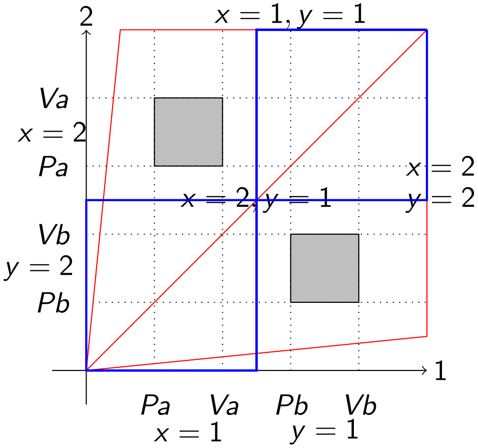


Geometric model

back to our first concurrent program: directed homotopy

Notion of **directed** homotopy

- directed paths correspond to execution traces
- path that can be continuously deformed to another are equivalent
- four schedulings are equivalent



Cubical areas
defition

Definition (cube)

A n-cube is the cartesian product of n intervals:

$$C = I_1 \times ... \times I_n$$

In particular those intervals can be equal to $\mathbb{R}$

Definition

A cubical area of $\mathbb{R}^n$ is a subset of $\mathbb{R}^n$ who can be covered by a finite number of cubes.

The geometric model of a *PV* program is a cubical area, as well as its complement in $\mathbb{R}^n$

Cubical areas

covering families

Let $\mathcal{M} = \{C_1, ..., C_k\}$ a finite family of n-cubes. Then its corresponding cubical area is

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

Since many families can cover the same cubical area, we need some normal form.

Definition

A cube of X is maximal if he is not contained in another cube of X. We define $\gamma(X)$ the set of maximal cubes of the cubical area X

$$\alpha \circ \gamma(X) = X$$

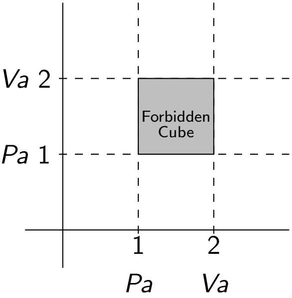
$$\gamma \circ \alpha(\mathcal{M}) = \text{maximal cubes of the area covered by } \mathcal{M}$$

Cubical areas

Example of maximal cubes

Maximal cubes of X

- $C_l =]-\infty, 1] \times \mathbb{R}$
- $C_r = [1, \infty[\times \mathbb{R}$
- $C_u = \mathbb{R} \times]-\infty, 1[$
- $C_d = \mathbb{R} \times [1, \infty[$

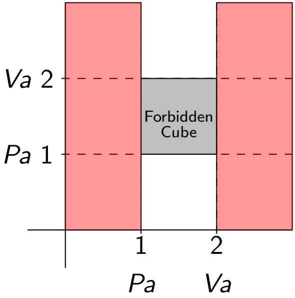


Cubical areas

Example of maximal cubes

Maximal cubes of X

- $C_l =]-\infty, 1] \times \mathbb{R}$
- $C_r = [1, \infty[\times \mathbb{R}$

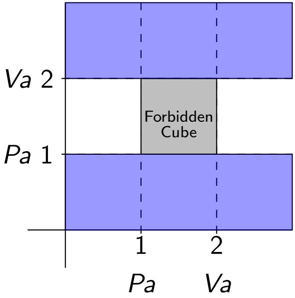


Cubical areas

Example of maximal cubes

Maximal cubes of X

- $C_u = \mathbb{R} \times]-\infty, 1[$
- $C_d = \mathbb{R} \times [1, \infty[$



Cubical area

complement of the cubical area

From a *PV* program P (with n threads), we get a family $\mathcal{F} = \{C_1, ..., C_k\}$ of **forbidden** cubes, by looking at all the ressources.

The cubical area $[P]$ is

$$X = \mathbb{R}^n \setminus (C_1 \cup ... \cup C_k)$$

Its complement is the **forbidden region** (also a cubical area) :

$$X^c = C_1 \cup ... \cup C_k = \alpha(\mathcal{F})$$

Factorization of cubical areas

Definition

A factorization of a cubical area X of $\mathbb{R}^n$ is a decomposition of X as a cartesian product (up to permutations of coordinates)

$$X = X_1 \times \ldots \times X_k$$

If $X = X_1 \times X_2 \times X_3$ there exists many factorizations by regrouping terms

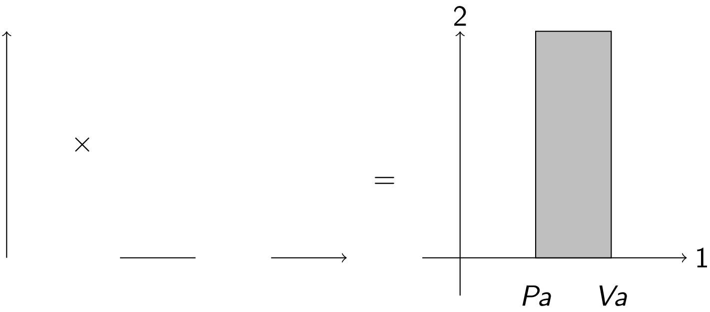
$$X = (X_1 \times X_2) \times X_3 = X_1 \times (X_2 \times X_3) = \underbrace{(X_1 \times X_2 \times X_3)}_{\textit{trivial factorization}}$$

Theorem (Balabonski, Haucourt)

A cubical area admits a unique factorization in irreducibles elements

Factorization

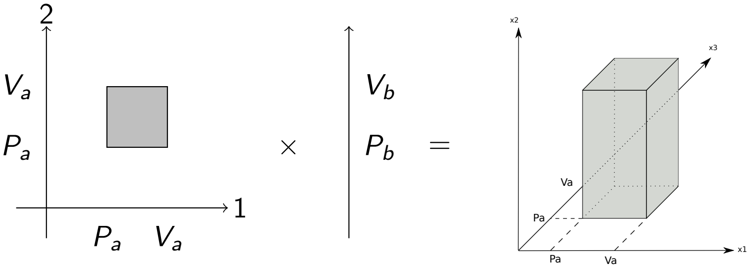
The 2-dimensional "pillar"



$$X = \mathbb{R} \times (\mathbb{R} \setminus [1,2])$$

Factorization

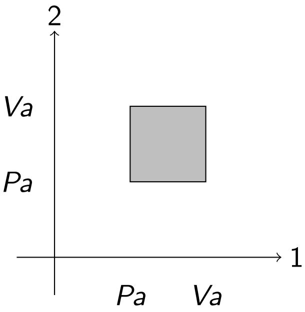
The 3-dimensional "pillar"



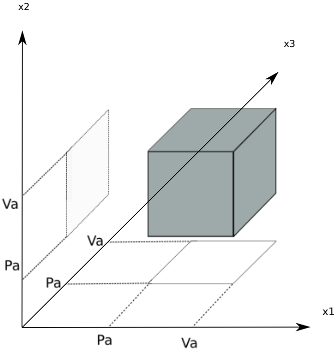
$$X = (\mathbb{R}^2 \setminus ([1,2]^2)) \times \mathbb{R}$$

Factorization

The "floating cube" examples



Has no factorization except the trivial one.



Has no factorization except the trivial one.

Link with the decomposition of programs

Key concept

Factorizing the geometric model $[P]$ of a concurrent programs is equivalent to decomposing P into groups of independent process.

goal

Our goal is thus to give an algorithm who factorizes a cubical area as much as possible.

Syntactic algorithm

a naive algorithm

An equivalence relation

From a *PV* program P we say that the process P_i and P_j are related if they both share a ressource. We'll write $P_i \sim P_j$.

The syntactic algorithm

Take the transitive closure of $\sim$ over all the process. The equivalence classes found are classes of **syntactic independent** programs.

Syntactic algorithm

An example

$$\begin{aligned} P \quad := \quad & P_1 = Pa.Va \\ \| \quad & P_2 = Pb.Vb \\ \| \quad & P_3 = Pa.Va \\ \| \quad & P_4 = Pb.Vb \end{aligned}$$

$P_1 \sim P_3$
$P_2 \sim P_4$
Syntactic factorization is

$$(1,3),(2,4)$$

$$\begin{aligned} P \quad := \quad & P_1 = Pa.Pb.Va.Vb \\ \| \quad & P_2 = Pb.Pc.Vb.Vc \\ \| \quad & P_3 = Pc.Pd.Vc.Vd \\ \| \quad & P_4 = Pd.Pa.Vc.Va \end{aligned}$$

$P_1 \sim P_2$
$P_2 \sim P_3$
$P_3 \sim P_4$
Syntactic factorization is trivial

$$(1,2,3,4)$$

Syntactic algorithm

A big limitation

$$
\begin{array}{rcrcl}
P & := & P_1 & = & Pa.Pc.Vc.Va \\
 & \| & P_2 & = & Pb.Pc.Vc.Vb \\
 & \| & P_3 & = & Pa.Pc.Vc.Va \\
 & \| & P_4 & = & Pb.Pc.Vc.Vb
\end{array}
$$

Here a, b are mutexes and c has **arity** 2. All the process are syntactically linked through c.
A careful examination shows that in fact P_1, P_3 are independent from P_2, P_4, the ressource c is useless.

Geometric algorithm

the context

From now on our input is a family $\mathcal{F} = \{C_1, \ldots, C_k\}$ of forbidden cubes

Goal

We want to factorize the cubical area

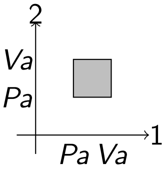
$$X = \mathbb{R}^n \setminus (C_1 \cup \ldots \cup C_k)$$

Into its unique factorization of irreducibles.

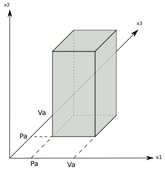
Back to geometry
syntactic link and cubes

Idea

If $P_1 \sim P_2$, then there is a forbidden cube C whose projections on coordinates 1 and 2 are different from $\mathbb{R}$ (finite).



$C = [1,2] \times [1,2]$, both 1 and 2 are not $\mathbb{R}$



$C = [1,2] \times [1,2] \times \mathbb{R}$ coordinates 1 and 2 of C are not $\mathbb{R}$ but not 3

The geometric algorithm

syntactic $\to$ geometric

geometric link

We say that two coordinates i and j are geometrically linked (through C) ($i \sim_g j$) if the projections of C on i and j are **not equal to** $\mathbb{R}$.

Geometric algorithm (naive version)

Let $X = \mathbb{R}^n \setminus (C_1 \cup ... \cup C_k)$, then the transitive closure of $\sim_g$ on all the C_i gives us a partition of the coordinate who is a factorization of X

The geometric algorihm

Example of the naive algorithm

$\mathcal{F} = \{C_1, C_2, C_3\}$ of $\mathbb{R}^4$

$$\begin{aligned} C_1 &= \mathbb{R} \times [0,1] \times [2,5] \times \mathbb{R} \\ C_2 &= [0,1] \times \mathbb{R} \times \mathbb{R} \times [0,1] \\ C_3 &= \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times [1,6] \end{aligned}$$

$2 \sim_g 3$ with C_1
$1 \sim_g 4$ with C_2
C_3 has only one projections not equal to $\mathbb{R}$.
Factorization is $(1,4),(2,3)$

Floating cube in dimension n

$\mathcal{F} = \{C_1\}$

$$C_1 = [1,2]^n$$

No projections equal to $\mathbb{R}$.
Thus $i \sim_g j$ for all coordinates.
We got the trivial factorization $(1, 2, ..., n)$

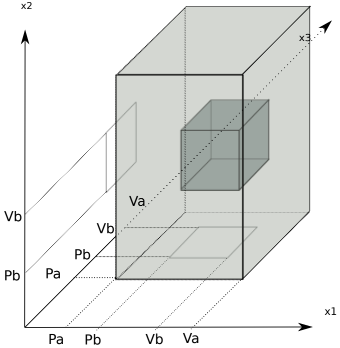
The problem
covering issue

$\mathcal{F} = \{C_1, C_2\}$

$$C_1 = [2,3] \times [2,3] \times [1,2]$$

$$C_2 = [1,4] \times [1,4] \times \mathbb{R}$$

$1 \sim_g 2 \sim_g 3$ with the floating cube C_1
But in fact $C_1 \subset C_2$ and X factorize as $(1,2),(3)$



A floating cube included in a pillar

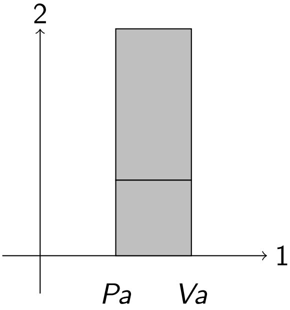
The problem
Another sort of covering

$\mathcal{F} = \{C_1, C_2\}$ in $\mathbb{R}^2$

$$C_1 =]-\infty, 1[\times[1,2]$$

$$C_2 = [1, \infty[\times[1,2]$$

$1 \sim_g 2$ with both cubes
But in fact
$C_1 \cup C_2 = \mathbb{R} \times [1,2]$ and X factorize as $(1),(2)$



The problem

The need for the maximal cubes

When a forbidden cube is included in the union of the other, you can lose some information on the factorisation in irreducibles
For every non empty family $\mathcal{F}$ of cubes, you can always add one cube who links together all the coordinates.

Definition

Let X be a cubical area and $\mathcal{F}$ a family of forbidden cube, then

$$MFC(X) = \gamma(\alpha(F))$$

Are the Maximal Forbidden Cubes of X

The geometric algorithm

The good one

You can apply the "naive" geometric algorithm to any covering family of cubes.

Theorem (Ninin)

The geometric algorithm applied to $MFC(X)$ yields the factorization in irreducibles.

Remark: Other family of forbidden cubes will give **a** factorization but not necesseraly the one in irreducibles.

The geometric algorithm

covering issues uncovered

$\mathcal{F} = \{C_1, C_2\}$

$C_1 = [2,3] \times [2,3] \times [1,2]$

$C_2 = [1,4] \times [1,4] \times \mathbb{R}$

$C_1 \subset C_2$ so $MFC(X) = C_2$

$\mathcal{F} = \{C_1, C_2\}$ in $\mathbb{R}^2$

$C_1 =]-\infty, 1[\times [1,2]$

$C_2 = [1, \infty[\times [1,2]$

$MFC(X) = \mathbb{R} \times [1,2]$

The geometric algorithm

an intuition of why this works

Suppose $X = X_1 \times X_2$ then

$$X^c = (X_1^c \times \mathbb{R}^{dim(X_2)} \cup \mathbb{R}^{dim(X_1)} \times X_2^c)$$

lemma that makes things works

$$MFC(X_1 \times X_2) = \{C_1 \times \mathbb{R}^{dim(X_2)}\} \cup \{\mathbb{R}^{dim(X_1)} \times C_2\}$$

with $C_i \in MFC(X_i)$

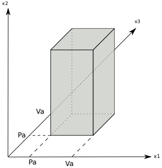
Finite coordinate, another way to see it

An example

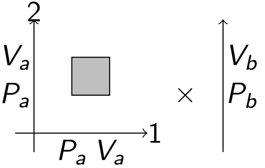
X (3d pillar)= $(2Dfloatingcube) \times \mathbb{R}$
$C_1 = [1,2]^2$ (cube of X_1^c)
$C_1 \times \mathbb{R}$ (cube of X^c)

Since $\mathbb{R}$ has no forbidden cube we cant find forbidden cube of X of the forms $\mathbb{R}^2 \times C_2$.

$X=$



$X_1 \times X_2 =$



Complexity of the algorithm

complexity

Given a family of k cubes in dimension n the complexity of the geometric algorithm is $O(n * k * log(k))$

This algorithm has been implemented in the static analyzer ALCOOL in ocaml.

A major problem

Computing the maximal cubes from a covering family is highly exponential.

Some solutions

We can find some alternatives to computing the maximal cubes

- Removing the cubes entirely contained in another one
- Finding cube included in the union of the others.
- We need the finite coordinate of the maximal cubes, they can be found without computing them explicitely

Perspectives

- Generalizing to programs with loops and branchings
- Proving that computing the maximal cubes is hard
- find good heuristics for relations of maximal cubes without computing them
- Define a notion of quasi-independance by modifying the semantic of a program hence losing some power while being more easily analyzed

Other works using thoses ideas

- It is possible to define categories on the cubical areas(similar to fondamental group)
- We have a theorem of unique factorization for some of them (no loops)
- We can show that the factorization are "equivalent"

THE END

Thank you for listening !