Control Flow Structures of Concurrent Programs are Higher Dimensional Mathematical Objects

Emmanuel Haucourt

Wednesday 13th April 2016
A Toy Language
from Dijkstra's "Cooperating Sequential Processes" paper

Program
A Toy Language
from Dijkstra's "Cooperating Sequential Processes" paper

Resource Declarations
A Toy Language
from Dijkstra's “Cooperating Sequential Processes” paper

Resource Declarations

Process Declarations
A Toy Language
from Dijkstra's “Cooperating Sequential Processes” paper
A Toy Language
Resource Declaration
A Toy Language
Resource Declaration

- `sem: <int> <set of identifiers>`
A Toy Language

Resource Declaration

- sem: <int> <set of identifiers>
- sync: <int> <set of identifiers>
A Toy Language
Resource Declaration

- **sem:** `<int>` `<set of identifiers>`
- **sync:** `<int>` `<set of identifiers>`
- **var:** `<identifier>` = `<constant>`
A Toy Language
The Hasse / Syracuse algorithm

var: x = 7

proc:
p = ()+[x=1]+C(q)

proc:
q = (x:=x/2 ; C(p))+[x \% 2 = 0]+(x:=3*x+1; C(p))

init: p
Building the Control Flow Graph
of the Hasse-Syracuse algorithm
Building the Control Flow Graph
of the Hasse-Syracuse algorithm

entry point of the
basic block of $p$

$x=1$

$C(q)$
Building the Control Flow Graph
of the Hasse-Syracuse algorithm

$entry point of the basic block of p$

$x := 1$

$C(q)$

$entry point of the basic block of q$

$x := 3x + 1$

$x := x/2$

$x \% 2 = 0$

$C(p)$

$x := 3x + 1$
Building the Control Flow Graph
of the Hasse-Syracuse algorithm

entry point of the basic block of $p$

entry point of the basic block of $q$

$x = 1$

$x = 3x + 1$

$x = x/2$

$x \% 2 = 0$

$x = 3x + 1$
Building the Control Flow Graph of the Hasse-Syracuse algorithm

entry point of the basic block of $p$

$x = 1$

entry point of the basic block of $q$

$x \equiv 0 \pmod{2}$

$x := x / 2$

$x := 3 \times x + 1$

$C(p)$

$C(q)$
Building the Control Flow Graph of the Hasse-Syracuse algorithm

entry point of the basic block of $p$  

$\mathcal{C}(q)$

entry point of the basic block of $q$

$x := 3x + 1$

$x := x / 2$

$x \% 2 = 0$

entry point of the basic block of $p$
Building the Control Flow Graph
of the Hasse-Syracuse algorithm

entry point of the basic block of $p$

$x := 1$

entry point of the basic block of $q$

$x \equiv x \mod 2 = 0$

$x := x/2$

$x := 3 \times x + 1$
Reducing the Control Flow Graph of the Hasse-Syracuse algorithm

The current value of $x$ is 7.
Reducing the Control Flow Graph
of the Hasse-Syracuse algorithm

\[ x := \begin{cases} x/2 & \text{if } x \equiv 0 \pmod{2} \\ 3x + 1 & \text{otherwise} \end{cases} \]
Reducing the Control Flow Graph
of the Hasse-Syracuse algorithm

entry point

\[ x := x / 2 \]

\[ x := 3 \times x + 1 \]

\[ x \mod 2 = 0 \]

\[ x = 1 \]

the current value of \( x \) is 7
An Execution Trace

on a control flow graph

\[ x := \frac{x}{2} \]

\[ x := 3x + 1 \]

the current value of \( x \) is 7
An Execution Trace
on a control flow graph

the current value of $x$ is 7
An Execution Trace

on a control flow graph

the current value of x is 7
An Execution Trace
on a control flow graph

the current value of $x$ is 22
An Execution Trace
on a control flow graph

- \( x := x/2 \)
- \( x := 3x + 1 \)
- \( x \mod 2 = 0 \)
- \( x = 1 \)

The current value of \( x \) is 22.
An Execution Trace
on a control flow graph

The current value of $x$ is 22
An Execution Trace

on a control flow graph

entry point

\[ x := x / 2 \]
\[ x := 3 \times x + 1 \]
\[ x \% 2 = 0 \]
\[ x = 1 \]

the current value of \( x \) is 7

\[ x = 1 \]

the current value of \( x \) is 22
An Execution Trace

on a control flow graph

\[
x := x/2
\]

\[
x := 3 \cdot x + 1
\]

the current value of \( x \) is 11
An Execution Trace

on a control flow graph

An Execution Trace on a control flow graph

the current value of \( x \) is 11
An Execution Trace
on a control flow graph

The current value of $x$ is 11
An Execution Trace

on a control flow graph

entry point

\[ x = 1 \]

\[ x = \frac{x}{2} \]

\[ x \mod 2 = 0 \]

x%2=0

\[ x = 3x + 1 \]

\[ x = \frac{x}{2} \]

the current value of \( x \) is 11
An Execution Trace
on a control flow graph

the current value of \( x \) is 34
An Execution Trace
on a control flow graph

the current value of $x$ is 34
An Execution Trace

on a control flow graph

x := x/2
x := 3x + 1
x \% 2 = 0

the current value of x is 7

The current value of x is 34
An Execution Trace
on a control flow graph

x := x / 2
x := 3 * x + 1
x ≡ 0
x = 1

the current value of x is 7

x := x / 2

the current value of x is 34
An Execution Trace

on a control flow graph

the current value of $x$ is 17
An Execution Trace
on a control flow graph

x := x/2
x := 3*x + 1
x%2 = 0
x = 1
the current value of x is 7

x := x/2

the current value of x is 17
An Execution Trace
on a control flow graph

the current value of $x$ is 17
An Execution Trace
on a control flow graph

entry point

x := x/2

x%2 = 0

x := 3*x + 1

the current value of x is 17
An Execution Trace
on a control flow graph

x := x/2
x := 3x + 1
x % 2 = 0
x = 1

the current value of x is 7

the current value of x is 52
An Execution Trace
on a control flow graph

x := x/2
x := 3*x + 1
x \% 2 = 0
x = 1

the current value of x is 7

x := x/2

x := 3*x + 1

the current value of x is 52
An Execution Trace on a control flow graph

\[
x := x/2
\]
\[
x := 3x + 1
\]
\[
x \% 2 = 0
\]
\[
x = 1
\]

the current value of \( x \) is 52
An Execution Trace

on a control flow graph

entry point

x := x/2

x := 3*x + 1

x \mod 2 = 0

x = 1

the current value of x is 7

x = 1

the current value of x is 52
An Execution Trace
on a control flow graph

x := x/2

x := 3*x + 1

x % 2 = 0

x = 1

the current value of x is 7

the current value of x is 26

the current value of x is 26
An Execution Trace

on a control flow graph

the current value of \( x \) is 26
An Execution Trace

on a control flow graph

\[ x := x / 2 \]
\[ x := 3 \times x + 1 \]
\[ x \mod 2 = 0 \]
\[ x = 1 \]

the current value of \( x \) is 7

the current value of \( x \) is 26
An Execution Trace

on a control flow graph

the current value of \( x \) is 26
An Execution Trace
on a control flow graph

the current value of $x$ is 13
An Execution Trace
on a control flow graph

the current value of \( x \) is 13
An Execution Trace
on a control flow graph

the current value of x is 13
An Execution Trace
on a control flow graph

x := x/2
x := 3*x + 1
x % 2 = 0
x = 1

the current value of x is 7

the current value of x is 13
An Execution Trace
on a control flow graph

\( \text{x} := \text{x} / 2 \)
\( \text{x} := 3 \times \text{x} + 1 \)
\( \text{x} \mod 2 = 0 \)
\( \text{x} = 1 \)

the current value of \( \text{x} \) is 7

the current value of \( \text{x} \) is 40
An Execution Trace
on a control flow graph

The current value of $x$ is 40
An Execution Trace
on a control flow graph

the current value of \( x \) is 40
An Execution Trace
on a control flow graph

x := x/2
x := 3*x + 1
x % 2 = 0
x = 1

the current value of x is 7

x := x/2

x := 3*x + 1

the current value of x is 40
An Execution Trace
on a control flow graph

the current value of $x$ is 20
An Execution Trace
on a control flow graph

x := x/2

x := 3*x + 1

x % 2 = 0

x := 1

the current value of x is 7

the current value of x is 20

the current value of x is 20
An Execution Trace
on a control flow graph

x:=x/2
x:=3*x+1
x%2=0
x=1
the current value of x is 7
the current value of x is 20

the current value of x is 20
An Execution Trace

on a control flow graph

entry point

x:=x/2

x:=3*x+1

x%2=0

x=1

the current value of x is 20
An Execution Trace

on a control flow graph

entry point

x := x/2

x%2=0

x := 3*x+1

the current value of x is 10
An Execution Trace

on a control flow graph

entry point

\[
x = 1
\]

\[
x := x/2
\]

\[
x \mod 2 = 0
\]

\[
x := 3 \times x + 1
\]

the current value of \( x \) is 10
An Execution Trace

on a control flow graph

\[ x := x / 2 \]

\[ x := 3 \times x + 1 \]

\[ x \mod 2 = 0 \]

\[ x = 1 \]

the current value of \( x \) is 10
An Execution Trace
on a control flow graph

the current value of $x$ is 10
An Execution Trace
on a control flow graph

entry point

the current value of $x$ is 7

the current value of $x$ is 5

the current value of $x$ is 5
An Execution Trace
on a control flow graph

the current value of $x$ is 5
An Execution Trace
on a control flow graph

x := x/2
x := 3*x + 1
x%2 = 0
x = 1

the current value of $x$ is 7

the current value of $x$ is 5

the current value of $x$ is 5
An Execution Trace
on a control flow graph

the current value of $x$ is 5
An Execution Trace
on a control flow graph

the current value of $x$ is 16
An Execution Trace
on a control flow graph

the current value of $x$ is 16
An Execution Trace
on a control flow graph

entry point

the current value of x is 16
An Execution Trace
on a control flow graph

x := x/2
x := 3*x + 1
x%2 = 0
x = 1

the current value of x is 7
the current value of x is 16
An Execution Trace
on a control flow graph

the current value of \(x\) is 8
An Execution Trace

on a control flow graph

x := x/2

x := 3*x + 1

x % 2 = 0

x = 1

the current value of x is 7

entry point

x = 1

x := 3*x + 1

x % 2 = 0

the current value of x is 8
An Execution Trace
on a control flow graph

x := x/2

x := 3*x + 1

x \% 2 = 0

x = 1

the current value of x is 7

the current value of x is 8

the current value of x is 8
An Execution Trace

on a control flow graph

entry point

\[ x := x/2 \]

\[ x := 3x + 1 \]

\[ x \mod 2 = 0 \]

\[ x = 1 \]

the current value of \( x \) is 7

the current value of \( x \) is 8

the current value of \( x \) is 8

the current value of \( x \) is 8
An Execution Trace

on a control flow graph

entry point

\begin{align*}
x &= x/2 \\
x &= 3x + 1 \\
x \% 2 &= 0 \\
x &= 1 \\
\end{align*}

the current value of $x$ is 4
An Execution Trace
on a control flow graph

The current value of $x$ is 4
An Execution Trace
on a control flow graph

the current value of \( x \) is 4
An Execution Trace

on a control flow graph

entry point

\( x = 1 \)

\( x \equiv 2 \equiv 0 \)

\( x = 3 \times x + 1 \)

the current value of \( x \) is 4
An Execution Trace
on a control flow graph

entry point

\[ x := x/2 \]

\[ x := 3x + 1 \]

\[ x \% 2 = 0 \]

\[ x = 1 \]

the current value of \( x \) is 7

the current value of \( x \) is 2

the current value of \( x \) is 2
An Execution Trace
on a control flow graph

the current value of \( x \) is 2
An Execution Trace

on a control flow graph

entry point

\[ x = x / 2 \]

\[ x = 3 \times x + 1 \]

\[ x \% 2 = 0 \]

the current value of \( x \) is 2

the current value of \( x \) is 7
An Execution Trace
on a control flow graph

- $x := x/2$
- $x := 3x + 1$
- $x \mod 2 = 0$
- $x = 1$
- the current value of $x$ is 7
- the current value of $x$ is 2

the current value of $x$ is 2
An Execution Trace
on a control flow graph

the current value of $x$ is 1
An Execution Trace
on a control flow graph

1

\[ x := \frac{x}{2} \]

\[ x := 3x + 1 \]

\[ x \mod 2 = 0 \]

\[ x = 1 \]

The current value of \( x \) is 1
An Execution Trace
on a control flow graph

the current value of \( x \) is 1
An Execution Trace
on a control flow graph

x := x/2
x := 3*x + 1
x % 2 == 0
x = 1

the current value of x is 1
An Execution Trace

on a control flow graph

the current value of $x$ is 1
Precubical sets

higher dimensional graphs
Precubical sets
higher dimensional graphs
Precubical sets
higher dimensional graphs
Precubical sets
higher dimensional graphs
Precubical sets
higher dimensional graphs
Precubical sets
higher dimensional graphs

$K_0$
Precubical sets
higher dimensional graphs
Precubical sets
higher dimensional graphs
Precubical sets

higher dimensional graphs
Precubical sets
higher dimensional graphs
Tensor product
of precubical sets

Given precubical sets $K$ and $K'$ of dimension $p$ and $q$, the set of $d$-cubes for $0 \leq d \leq p + q$

$$(K \otimes K')_d = \bigsqcup_{i+j=d} K_i \times K'_j$$

For $x \otimes y \in K_i \times K'_j$ with $i + j = d$ the $k^{th}$ face map, with $0 \leq k < d$, is given by

$$\partial_k^\pm (x \otimes y) = \begin{cases} 
\partial_k^\pm (x) \otimes y & \text{if } 0 \leq k < i \\
(x \otimes \partial_k^\pm y) & \text{if } i \leq k < d
\end{cases}$$
A Toy Language
Synchronization: the $W(\_)$ instruction

sync: 1 b

proc: p = W(b)

init: 2p
Tensor product of control flow graphs

\[ W(b) \otimes W(c) \]
Tensor product
of control flow graphs
Tensor product
of control flow graphs
Tensor product
of control flow graphs
Tensor product of control flow graphs
Tensor product
of control flow graphs
Discrete paths are “continuous”
Discrete paths are “continuous”
Discrete paths are “continuous”
Discrete paths are "continuous"
Discrete paths are “continuous”
Discrete paths are “continuous”
Discrete paths are “continuous”
Discrete paths
are "continuous"
Discrete paths are “continuous”
Discrete paths are “continuous”
Discrete paths
are “continuous”
Discrete paths are “continuous”
Discrete paths
are “continuous”
Discrete path on a model of dimension $N$

A sequence of points $p_0, \ldots, p_K$ s.t. for all $k \in \{1, \ldots, K\}$ one has

for all $n \in \{1, \ldots, N\}$ $\partial^+ p_n(k-1) = p_n(k)$ or $p_n(k) = p_n(k-1)$

or

for all $n \in \{1, \ldots, N\}$ $p_n(k-1) = \partial^- p_n(k)$ or $p_n(k) = p_n(k-1)$
Concurrent execution trace

\(\text{sync: } 1 b\)
Concurrent execution trace

\[ \text{sync: } \, 1 \, b \]
Concurrent execution trace

sync: 1 b
Concurrent execution trace

sync: \ 1 \ b
Concurrent execution trace

\[ \text{sync: } 1 b \]
Concurrent execution trace

sync: 1 b
Concurrent execution trace

\[ \text{sync: } 1 \ b \]
Concurrent execution trace

sync: 1 b
Not admissible concurrent execution trace

sync: 1 b
Not admissible concurrent execution trace

sync: 1 b
Not admissible concurrent execution trace

sync: 1 b
Not admissible concurrent execution trace

sync: 1 b
Forbidden points
due to synchronization

Each point \( p = (p_1, \ldots, p_d) \) such that

\[
0 < \text{card}\{ k \in \{1, \ldots, d\} \mid \text{label}(p_k) = W(b) \} \leq \text{arity}(b)
\]

is forbidden.
var: x = 0

proc: p = (x := 1)
proc: q = (x := 2)

init: p q
Not admissible execution trace
due to race condition

the value of $x$ is 0
Not admissible execution trace
due to race condition

the value of $x$ is 0
Not admissible execution trace
due to race condition

the value of $x$ is 0
Not admissible execution trace
due to race condition

the value of $x$ is ?
Admissible execution trace

that however meets a forbidden point

\[ x := 1 \]
\[ x := 2 \]
\[ \otimes \]

the value of \( x \) is 0
Admissible execution trace
that however meets a forbidden point

the value of $x$ is 0
Admissible execution trace
that however meets a forbidden point

the value of $x$ is 0
Admissible execution trace
that however meets a forbidden point

the value of $x$ is 1
Admissible execution trace
that however meets a forbidden point

\[
x := 1
\]
\[
x := 2
\]

the value of \( x \) is 2
Admissible execution trace
that however meets a forbidden point

the value of $x$ is 2
Admissible execution trace
that however meets a forbidden point

the value of $x$ is 2
Admissible execution trace
that however meets a forbidden point

the value of $x$ is 2
Admissible execution trace
avoiding forbidden points

the value of $x$ is 0
Admissible execution trace

avoiding forbidden points

\[ x := 1 \]
\[ x := 2 \]
\[ \otimes \]

the value of \( x \) is 0
Admissible execution trace
avoiding forbidden points

the value of $x$ is 0
Admissible execution trace
avoiding forbidden points

the value of $x$ is 1
Admissible execution trace
avoiding forbidden points

the value of $x$ is 1
Admissible execution trace
avoiding forbidden points

the value of $x$ is 2
Admissible execution trace
avoiding forbidden points

the value of $x$ is 2
Admissible execution trace
avoiding forbidden points

the value of $x$ is 2
Admissible execution trace
avoiding forbidden points

the value of $x$ is 2
Forbidden points
due to race conditions

A point \( p = (p_1, \ldots, p_d) \) is a race condition when there exist \( i \neq j \) such that
- both \( \lambda_i(p_i) \) and \( \lambda_j(p_j) \) are assignments trying to alter the same variable or
- \( \lambda_i(p_i) \) tries to alter a free variable of \( \lambda_j(p_j) \) or \( \lambda_j(\alpha) \) for some arrow \( \alpha \) such that \( \partial \alpha = p_j \).

In that case the point \( p \) is forbidden.
The replacement property
for admissible execution traces

Replacement

Any admissible execution trace that meets a race condition is “equivalent” to an admissible execution trace which avoids all of them.
A Toy Language

Desynchronization: the \texttt{P(\_)} and \texttt{V(\_)} instructions

\begin{verbatim}
sem:  1 a

proc:  p = P(a);V(a)

init:  2p
\end{verbatim}
Admissible concurrent execution trace

sem: 1 a
Admissible concurrent execution trace

sem: 1 a
Admissible concurrent execution trace

sem: 1 a
Admissible concurrent execution trace

sem: 1 a
Admissible concurrent execution trace

sem: 1 a
Admissible concurrent execution trace

\( \text{sem: } 1 \ a \)
Admissible concurrent execution trace

\[ \text{sem: } 1 \ a \]
Admissible concurrent execution trace

sem: 1 a
Admissible concurrent execution trace

\[ \text{sem: } 1 \ a \]
Admissible concurrent execution trace

sem: 1 a
Admissible concurrent execution trace

sem: 1 a
Admissible concurrent execution trace

\(\text{sem: } 1 a\)
Not admissible concurrent execution trace

sem: 1 a
Not admissible concurrent execution trace

sem: 1 a
Not admissible concurrent execution trace

sem: $1 \ a$
Not admissible concurrent execution trace

\[
\text{sem: } 1 \ a
\]
The potential functions
of processes and programs

A process $\pi$ is conservative when for all paths and all semaphores $s$, the amount of tokens of type $s$ held by the process at the end of the execution trace only depends on its arrival point. In that case the process $\pi$ comes with a potential function $F_\pi$:

$\{\text{semaphores}\} \times \{\text{points}\} \rightarrow \mathbb{N}$

A program $\Pi$ is conservative when so are its processes $\pi_1, \ldots, \pi_d$ and its potential function is given by

$F_\Pi(s, (p_1, \ldots, p_d)) = \sum_{k=1}^{d} F_{\pi_k}(s, p_k)$

If $F_\Pi(s, p) > \text{arity}(s)$ for some semaphore $s$, then $p$ is forbidden.
The potential functions
of processes and programs

A process $\pi$ is conservative when for all paths and all semaphores $s$, the amount of tokens of type $s$ held by the process at the end of the execution trace only depends on its arrival point.
The potential functions
of processes and programs

A process $\pi$ is conservative when for all paths and all semaphores $s$, the amount of tokens of type $s$ held by the process at the end of the execution trace only depends on its arrival point. In that case the process $\pi$ comes with a potential function $F_\pi$

$$F_\pi : \{\text{semaphores}\} \times \{\text{points}\} \rightarrow \mathbb{N}$$
The potential functions
of processes and programs

A process $\pi$ is conservative when for all paths and all semaphores $s$, the amount of tokens of type $s$ held by the process at the end of the execution trace only depends on its arrival point. In that case the process $\pi$ comes with a potential function $F_\pi$

$$F_\pi : \{\text{semaphores}\} \times \{\text{points}\} \rightarrow \mathbb{N}$$

A program $\Pi$ is conservative when so are its processes $\pi_1, \ldots, \pi_d$. 
The potential functions
of processes and programs

A process $\pi$ is conservative when for all paths and all semaphores $s$, the amount of tokens of type $s$ held by the process at the end of the execution trace only depends on its arrival point. In that case the process $\pi$ comes with a potential function $F_\pi$

$$F_\pi : \{\text{semaphores}\} \times \{\text{points}\} \rightarrow \mathbb{N}$$

A program $\Pi$ is conservative when so are its processes $\pi_1, \ldots, \pi_d$ and its potential function is given by

$$F_\Pi(s, (p_1, \ldots, p_d)) = \sum_{k=1}^{d} F_{\pi_k}(s, p_k)$$
The potential functions
of processes and programs

A process $\pi$ is conservative when for all paths and all semaphores $s$, the amount of tokens of type $s$ held by the process at the end of the execution trace only depends on its arrival point. In that case the process $\pi$ comes with a potential function $F_\pi$

$$F_\pi : \{\text{semaphores}\} \times \{\text{points}\} \rightarrow \mathbb{N}$$

A program $\Pi$ is conservative when so are its processes $\pi_1, \ldots, \pi_d$ and its potential function is given by

$$F_\Pi(s, (p_1, \ldots, p_d)) = \sum_{k=1}^{d} F_{\pi_k}(s, p_k)$$

If $F_\Pi(s, p) > \text{arity}(s)$ for some semaphore $s$, then $p$ is forbidden.
Conservative process

example
Conservative process

example
Conservative process

example
Conservative process
example
Conservative process

example
Conservative process

example
Conservative process

example
Conservative process

example
Conservative process

example
Conservative process

example
Conservative process

example
Conservative process

example
Conservative process

example
Conservative process

example
Conservative process

example
Conservative process
example
Conservative process

d Example

\[ P(s) \]

\[ V(s) \]
Conservative process

example
Not conservative process

example
Not conservative process

example
Not conservative process
example
Not conservative process

example
Not conservative process

example
Not conservative process

example
Not conservative process

example
Not conservative process

example
Not conservative process
example
Not conservative process

example
Not conservative process

example
Not conservative process

example
Not conservative process

$P(s)$

example
Not conservative process

example

$P(s)$
Not conservative process

equation

\[ P(s) \]
Not conservative process

example
Not conservative process

example
Not conservative process

example

conflict

$P(s)$
Discrete model

sem: 1 a
Discrete model

\( \text{sem: 1 a} \)
Discrete model

sem: 1 a
Discrete model

sem: 1 a

Diagram of discrete model with 1 and 0 states in the model.
Discrete model

sem: 1a
Discrete model

sem: 1 a
Discrete model

sem: 1 a
Discrete Model

sync: 1 b
Discrete Model

\[ \text{sync: } 1 \ b \]
Discrete Model

\[
\text{sync: } 1 \ b
\]
Discrete Model

sync: 1 b
Discrete Model

sync: 1 b

[Diagram of a grid with labeled nodes and arrows indicating connections. The grid is labeled with 1 W(b) and 0 W(a).]
Discrete Model

sync: \(1 \ b\)
Discrete Model

sync: 1 \ b
Locally ordered spaces

Directed atlas $\mathcal{U}$
Locally ordered spaces

Directed atlas $\mathcal{U}$

For all points $p$, 

$p$
Locally ordered spaces

Directed atlas $\mathcal{U}$

For all points $p$, for all directed neighborhoods $A$ and $B$ of $p$, there exists a directed neighborhood $C$ of $p$ such that $C \subseteq A \cap B$ and $\not\subseteq A \setminus C = \not\subseteq C \setminus B$. 
Locally ordered spaces

Directed atlas $\mathcal{U}$

For all points $p$, for all directed neighborhoods $A$ and $B$ of $p$, there exists a directed neighborhood $C$ of $p$ such that $C \subseteq A \cap B$ and $\leq_A |_C = \leq_C = \leq_B |_C$. 
The directed circle
as a local pospace
The directed circle
as a local pospace
The directed circle
as a local pospace
The directed circle
as a local pospace
The directed circle
as a local pospace
Directed geometric realization

Main property

\[ 1_\downarrow \downarrow : \{ \text{Precubical sets} \} \rightarrow \{ \text{Locally ordered spaces} \} \]
Directed geometric realization

Main property

\[\downarrow : \{\text{Precubical sets}\} \to \{\text{Locally ordered spaces}\}\]

\[U\downarrow(1K\downarrow) = \bigsqcup_{d \in \mathbb{N}} K_d \times ]0, 1[^d\]
Directed geometric realization

Main property

\[ U(K) = \bigsqcup_{d \in \mathbb{N}} K_d \times ]0, 1[^d \]

The main property

\[ \vert K^{(1)} \otimes \cdots \otimes K^{(n)} \vert \cong \vert K^{(1)} \vert \times \cdots \times \vert K^{(n)} \vert \]
Let $G^{(1)}, \ldots, G^{(n)}$ the control flow graphs of the program.
The continuous model
of a conservative program

Let $G^{(1)}, \ldots, G^{(n)}$ the control flow graphs of the program. For all $d \in \mathbb{N}$, the set $F_d$ of forbidden points of dimension $d$. 
Let \( G^{(1)}, \ldots, G^{(n)} \) the control flow graphs of the program. For all \( d \in \mathbb{N} \), the set \( F_d \) of forbidden points of dimension \( d \).

\[
K = G^{(1)} \otimes \ldots \otimes G^{(n)}
\]
The continuous model
of a conservative program

Let $G^{(1)}, \ldots, G^{(n)}$ the control flow graphs of the program. For all $d \in \mathbb{N}$, the set $F_d$ of forbidden points of dimension $d$.

$$ K = G^{(1)} \otimes \cdots \otimes G^{(n)} $$

\[
\bigsqcup_{d \in \mathbb{N}} (K_d \setminus F_d) \times ]0, 1[^d
\]
From discrete to continuous

sem: 1 a  sync: 1 b
From discrete to continuous

sem: 1 a    sync: 1 b

z := 1

W(b)

\[ \text{V}(a) \]

x := y

P(a)

[\[ \text{y} := 0 \text{ W}(b) \text{ P}(a) \text{ x} := z \text{ V}(a) \]]

\[ \otimes \]
From discrete to continuous

sem: 1 a sync: 1 b
From discrete to continuous

sem: 1 a sync: 1 b
From discrete to continuous

sem: 1 a sync: 1 b
From discrete to continuous

sem: 1 a   sync: 1 b
From discrete to continuous

sem: 1 a    sync: 1 b

\[ \begin{align*}
&x := y \\
&P(a) \\
&W(b) \\
&z := 1
\end{align*} \]
From discrete to continuous

sem: 1 a  sync: 1 b
From discrete to continuous

sem: 1 a    sync: 1 b

V(a)

x:=y

P(a)

W(b)

z:=1
From discrete to continuous

sem: 1 a  sync: 1 b
From discrete to continuous

\[ y := 0 \]

\[ W(b) \]

\[ P(a) \]

\[ x := y \]

\[ V(a) \]

\[ z := 1 \]

\[ W(b) \]

\[ P(a) \]

\[ x := z \]

\[ V(a) \]
From discrete to continuous

sem: 1 a  sync: 1 b

\[ y := 0 \]
\[ W(b) \]
\[ P(a) \]
\[ x := z \]
\[ V(a) \]

\[ z := 1 \]
\[ W(b) \]
\[ P(a) \]
\[ x := z \]
\[ V(a) \]
From discrete to continuous

sem: 1 a   sync: 1 b

\[
\begin{align*}
  y &:= 0 \\
  W(b) & \quad P(a) & \quad x &:= z \\
  z &:= 1 \\
  W(b) & \quad P(a) & \quad x &:= y \\
  V(a) & \quad V(a)
\end{align*}
\]
From discrete to continuous

sem: 1 a  sync: 1 b
From discrete to continuous

sem: 1a sync: 1b

\[ y := 0 \]
\[ W(b) \]
\[ P(a) \]
\[ x := z \]
\[ V(a) \]
\[ z := 1 \]
\[ W(b) \]
\[ P(a) \]
\[ x := y \]
\[ V(a) \]
Weakly directed homotopy of directed paths
L. Fajstrup, É. Goubault, and M. Raussen (1998)

A weakly directed homotopy is a continuous map \( h : [0, r] \times [0, q] \to X \) such that
Weakly directed homotopy of directed paths
L. Fajstrup, É. Goubault, and M. Raussen (1998)

A weakly directed homotopy is a continuous map $h : [0, r] \times [0, q] \rightarrow X$ such that

1) the mappings $h(0, -)$ and $h(r, -)$ are constant
A weakly directed homotopy is a continuous map $h : [0, r] \times [0, q] \to X$ such that

1) the mappings $h(0, -)$ and $h(r, -)$ are constant

2) the mappings $h(-, s)$ are directed paths
Weakly directed homotopy of directed paths
L. Fajstrup, É. Goubault, and M. Raussen (1998)

A weakly directed homotopy is a continuous map \( h : [0, r] \times [0, q] \rightarrow X \) such that

1) the mappings \( h(0, -) \) and \( h(r, -) \) are constant

2) the mappings \( h(-, s) \) are directed paths
A weakly directed homotopy is a continuous map $h : [0, r] \times [0, q] \to X$ such that

1) the mappings $h(0, -)$ and $h(r, -)$ are constant

2) the mappings $h(-, s)$ are directed paths
Weakly directed homotopy of directed paths
L. Fajstrup, É. Goubault, and M. Raussen (1998)

A weakly directed homotopy is a continuous map $h : [0, r] \times [0, q] \to X$ such that

1) the mappings $h(0, -)$ and $h(r, -)$ are constant

2) the mappings $h(-, s)$ are directed paths
A weakly directed homotopy is a continuous map \( h : [0, r] \times [0, q] \to X \) such that

1) the mappings \( h(0, -) \) and \( h(r, -) \) are constant

2) the mappings \( h(-, s) \) are directed paths
Substantiating the continuous models

Main theorem

**Adequacy**

The “actions” of weakly dihomotopic directed paths are the same. A directed path is an execution trace iff it is weakly dihomotopic with an execution trace.
Tetrahemihexacron
a.k.a. 3D Swiss Cross

sem: 1 a

proc:
  p = P(a); V(a)

init: 3p
Floating cube

influence of arity

sem: 2 a

proc:
  p = P(a); V(a)

init: 3p
The dining philosophers
with its deadlock attractor

sem:  1 a b c

proc:
  x = P(a);P(b);V(a);V(b)
  y = P(b);P(c);V(b);V(c)
  z = P(c);P(a);V(c);V(a)

init:  x y z
The Lipski algorithm
has no deadlock

sem:  1 x y z u v w

proc:
  p = P(x);P(y);P(z);V(x);P(w);V(z);V(y);V(w)
  q = P(u);P(v);P(x);V(u);P(z);V(v);V(x);V(z)
  r = P(y);P(w);V(y);P(u);V(w);P(v);V(u);V(v)

init:  p q r
Regions
over $G_1, \ldots, G_d$

A one dimensional block over $G$ is a finite union of connected components of $|G|$.
Regions over $G_1, \ldots, G_d$

A one dimensional block over $G$ is a finite union of connected components of $|G|$.

A block of dimension $d \in \mathbb{N}$ over $G_1, \ldots, G_d$ is a Cartesian product of one dimensional blocks $B_k$ over $G_k$ for $k \in \{1, \ldots, d\}$. 
Regions over $G_1, \ldots, G_d$

A one dimensional block over $G$ is a finite union of connected components of $|G|$.

A block of dimension $d \in \mathbb{N}$ over $G_1, \ldots, G_d$ is a Cartesian product of one dimensional blocks $B_k$ over $G_k$ for $k \in \{1, \ldots, d\}$.

A region of dimension $d \in \mathbb{N}$ over $G_1, \ldots, G_d$ is a finite union of $d$-blocks over $G_1, \ldots, G_d$. 
Regions
over \( G_1, \ldots, G_d \)

A one dimensional block over \( G \) is a finite union of connected components of \( |G| \).

A block of dimension \( d \in \mathbb{N} \) over \( G_1, \ldots, G_d \) is a Cartesian product of one dimensional blocks \( B_k \) over \( G_k \) for \( k \in \{1, \ldots, d\} \).

A region of dimension \( d \in \mathbb{N} \) over \( G_1, \ldots, G_d \) is a finite union of \( d \)-blocks over \( G_1, \ldots, G_d \).

If \( X \) and \( Y \) are regions over \( G_1, \ldots, G_d \) and \( G_1', \ldots, G_d' \), then \( X \times Y \) is a region over \( G_1, \ldots, G_d, G_1', \ldots, G_d' \).
Maximal blocks
Maximal blocks
Maximal blocks
Maximal blocks
Maximal blocks
Main results
Maximal subblocks and Boolean structure
Main results
Maximal subblocks and Boolean structure

**Maximal subblocks**

\[ X ⊆ 1G_1| \times \cdots \times 1G_d| \text{ is a region iff it has finitely many maximal subblocks.} \]
Main results
Maximal subblocks and Boolean structure

Maximal subblocks
$X \subseteq |G_1| \times \cdots \times |G_d|$ is a region iff it has finitely many maximal subblocks.

Boolean structure
The collection of regions over $G_1, \ldots, G_d$ form a Boolean subalgebra of the powerset of $|G_1| \times \cdots \times |G_d|$. 
Main results

Unique decomposition

Up to coordinates reordering, any region can be written as a Cartesian product of irreducible regions in a unique way. This is the prime decomposition of it.

Parallelization of code

The prime decomposition of the continuous model of some program provides a decomposition of the program as a parallel compound of "observationally independent" programs.
Main results
Unique decomposition

Prime decomposition
Up to coordinates reordering, any region can be written as a Cartesian product of irreducible regions in a unique way. This is the prime decomposition of it.
Main results

Unique decomposition

Prime decomposition

Up to coordinates reordering, any region can be written as a Cartesian product of irreducible regions in a unique way. This is the prime decomposition of it.

Parallelization of code

The prime decomposition of the continuous model of some program provides a decomposition of the program as a parallel compound of “observationally independent” programs.
Main result

Effectiveness

An algorithm (Nicolas Ninin)

Let $M_1, \ldots, M_b$ be the maximal subblocks of $X = \prod_{i=1}^{d} G_i \simeq \prod_{i=1}^{d} G_i \setminus X$. Let $\sim$ be the equivalence relation on $\{1, \ldots, d\}$ generated by $i \sim j$ when there exist $k \in \{1, \ldots, b\}$ such that

$$\text{proj}_i(M_k) \neq \prod_i G_i \quad \text{and} \quad \text{proj}_j(M_k) \neq \prod_i G_j$$

The prime decomposition of $X$ is given by the $\sim$-equivalence classes.
Parallelizing a program

sem: 1 a b
sem: 2 c

proc:
p = P(a);P(c);V(c);V(a)

proc:
q = P(b);P(c);V(c);V(b)

init: 2p 2q
Parallelizing a program

| sem: 1 a b | sem: 1 a b |
| sem: 2 c   | sem: 2 c   |
| proc:      | proc:      |
| p = P(a);P(c);V(c);V(a) | q = P(b);P(c);V(c);V(b) |
| init: 2p   | init: 2q   |
Directed Algebraic Topology and Concurrency