DIRECTED ALGEBRAIC TOPOLOGY

AND

CONCURRENCY

Emmanuel Haucourt

emmanuel.haucourt@polytechnique.edu

MPRI : Concurrency (2.3.1)
– Lecture 3 –

2020 – 2021

GEOMETRIC MODELS

Cartesian product

Cartesian product in $\mathcal{S}et$

Cartesian product in $\mathcal{Set}$

$$A \times B := \{(a, b) \mid a \in A \text{ and } b \in B\}$$

Cartesian product in $\mathcal{Set}$

$$A \times B := \{(a, b) \mid a \in A \text{ and } b \in B\}$$

There exist two mappings π_A and π_B

$$\begin{aligned} \pi_A :\ A \times B &\longrightarrow A \\ (a, b) &\longmapsto a \end{aligned} \qquad \begin{aligned} \pi_B :\ A \times B &\longrightarrow B \\ (a, b) &\longmapsto b \end{aligned}$$

Cartesian product in $\mathcal{S}et$

$$A \times B := \{(a, b) \mid a \in A \text{ and } b \in B\}$$

There exist two mappings π_A and π_B

$$\pi_A : A \times B \longrightarrow A \qquad (a, b) \longmapsto a$$

$$\pi_B : A \times B \longrightarrow B \qquad (a, b) \longmapsto b$$

such that for all sets X the following map is a bijection

$$\mathcal{S}et[X, A \times B] \longrightarrow \mathcal{S}et[X, A] \times \mathcal{S}et[X, B]$$

$$h \longmapsto (\pi_A \circ h, \pi_B \circ h)$$

Cartesian product in a category $\mathcal{C}$

The object c is the Cartesian product (in $\mathcal{C}$) of a and b when there exist two morphisms $\pi_a : c \to a$ and $\pi_b : c \to b$ such that for all objects x of $\mathcal{C}$ the following map is a bijection

$$\mathcal{C}[x, c] \longrightarrow \mathcal{C}[x, a] \times \mathcal{C}[x, b]$$

$$h \longmapsto (\ \pi_a \circ h\ ,\ \pi_b \circ h\ )$$

When such an object c exists we write $c = a \times b$

Cartesian product in the category of graphs (*Grph*)

Cartesian product in the category of graphs ($\mathcal{Grph}$)

$$\left(\begin{array}{c} A \\ \mathsf{t} \downarrow\downarrow \mathsf{s} \\ V \end{array} \right) \times \left(\begin{array}{c} A' \\ \mathsf{t}' \downarrow\downarrow \mathsf{s}' \\ V' \end{array} \right) \cong$$

Cartesian product in the category of graphs ($\mathcal{Grph}$)

$$\left(\begin{array}{c} A \\ {\scriptstyle t}\big\downarrow\big\downarrow{\scriptstyle s} \\ V \end{array}\right) \times \left(\begin{array}{c} A' \\ {\scriptstyle t'}\big\downarrow\big\downarrow{\scriptstyle s'} \\ V' \end{array}\right) \cong \left(\begin{array}{c} A \times A' \\ {\scriptstyle t\times t'}\big\downarrow\big\downarrow{\scriptstyle s\times s'} \\ V \times V' \end{array}\right)$$

The Cartesian product in $\mathcal{Grph}$ is deduced form the Cartesian product in $\mathcal{Set}$

Examples of Cartesian products

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{T}op$

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$.

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{T}\!op$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$.

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{PoSp}$

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{PoSp}$ is given by $X \times Y$ and the product order $\sqsubseteq_X \times \sqsubseteq_Y$.

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{PoSp}$ is given by $X \times Y$ and the product order $\sqsubseteq_X \times \sqsubseteq_Y$.
- The product of $(X, [\mathcal{U}]_\sim)$ and $(Y, [\mathcal{V}]_\sim)$ in $\mathcal{Lpo}$

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{PoSp}$ is given by $X \times Y$ and the product order $\sqsubseteq_X \times \sqsubseteq_Y$.
- The product of $(X, [\mathcal{U}]_\sim)$ and $(Y, [\mathcal{V}]_\sim)$ in $\mathcal{Lpo}$ is given by $X \times Y$ together with the collection of ordered charts $U \times V$ with $U \in \mathcal{U}$ and $V \in \mathcal{V}$.

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{PoSp}$ is given by $X \times Y$ and the product order $\sqsubseteq_X \times \sqsubseteq_Y$.
- The product of $(X, [\mathcal{U}]_\sim)$ and $(Y, [\mathcal{V}]_\sim)$ in $\mathcal{Lpo}$ is given by $X \times Y$ together with the collection of ordered charts $U \times V$ with $U \in \mathcal{U}$ and $V \in \mathcal{V}$.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{emb}$

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{PoSp}$ is given by $X \times Y$ and the product order $\sqsubseteq_X \times \sqsubseteq_Y$.
- The product of $(X, [\mathcal{U}]_\sim)$ and $(Y, [\mathcal{V}]_\sim)$ in $\mathcal{Lpo}$ is given by $X \times Y$ together with the collection of ordered charts $U \times V$ with $U \in \mathcal{U}$ and $V \in \mathcal{V}$.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{emb}$ does not exist.

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{PoSp}$ is given by $X \times Y$ and the product order $\sqsubseteq_X \times \sqsubseteq_Y$.
- The product of $(X, [\mathcal{U}]_\sim)$ and $(Y, [\mathcal{V}]_\sim)$ in $\mathcal{Lpo}$ is given by $X \times Y$ together with the collection of ordered charts $U \times V$ with $U \in \mathcal{U}$ and $V \in \mathcal{V}$.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{emb}$ does not exist.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{ctr}$

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathscr{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathscr{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathscr{PoSp}$ is given by $X \times Y$ and the product order $\sqsubseteq_X \times \sqsubseteq_Y$.
- The product of $(X, [\mathcal{U}]_\sim)$ and $(Y, [\mathcal{V}]_\sim)$ in $\mathscr{Lpo}$ is given by $X \times Y$ together with the collection of ordered charts $U \times V$ with $U \in \mathcal{U}$ and $V \in \mathcal{V}$.
- The product of (X, d_X) and (Y, d_Y) in $\mathscr{Met}_{emb}$ does not exist.
- The product of (X, d_X) and (Y, d_Y) in $\mathscr{Met}_{ctr}$ is given by $X \times Y$ together with $d\big((x, y), (x', y')\big) = \max\{d_X(x, x'), d_Y(y, y')\}$.

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{PoSp}$ is given by $X \times Y$ and the product order $\sqsubseteq_X \times \sqsubseteq_Y$.
- The product of $(X, [\mathcal{U}]_\sim)$ and $(Y, [\mathcal{V}]_\sim)$ in $\mathcal{Lpo}$ is given by $X \times Y$ together with the collection of ordered charts $U \times V$ with $U \in \mathcal{U}$ and $V \in \mathcal{V}$.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{emb}$ does not exist.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{ctr}$ is given by $X \times Y$ together with $d\big((x, y), (x', y')\big) = \max\{d_X(x, x'), d_Y(y, y')\}$.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{top}$

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{PoSp}$ is given by $X \times Y$ and the product order $\sqsubseteq_X \times \sqsubseteq_Y$.
- The product of $(X, [\mathcal{U}]_\sim)$ and $(Y, [\mathcal{V}]_\sim)$ in $\mathcal{Lpo}$ is given by $X \times Y$ together with the collection of ordered charts $U \times V$ with $U \in \mathcal{U}$ and $V \in \mathcal{V}$.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{emb}$ does not exist.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{ctr}$ is given by $X \times Y$ together with $d\big((x, y), (x', y')\big) = \max\{d_X(x, x'), d_Y(y, y')\}$.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{top}$ can also be given by $X \times Y$ together with the Euclidean product

$$d\big((x, y), (x', y')\big) = \sqrt{d_X^2(x, x') + d_Y^2(y, y')}$$

Examples of Cartesian products

- The product of (X, Ω_X) and (Y, Ω_Y) in $\mathcal{Top}$ is given by $X \times Y$ together with unions of subsets of the form $U \times V$ with $U \in \Omega_X$ and $V \in \Omega_Y$. It is the least topology making the projections continuous.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{Pos}$ is given by $X \times Y$ and the partial order $\sqsubseteq$ defined by $(x, y) \sqsubseteq (x', y')$ when $x \sqsubseteq_X x'$ and $y \sqsubseteq_Y y'$. It is the greatest partial order such that the projection are poset morphisms.
- The product of $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ in $\mathcal{PoSp}$ is given by $X \times Y$ and the product order $\sqsubseteq_X \times \sqsubseteq_Y$.
- The product of $(X, [\mathcal{U}]_\sim)$ and $(Y, [\mathcal{V}]_\sim)$ in $\mathcal{Lpo}$ is given by $X \times Y$ together with the collection of ordered charts $U \times V$ with $U \in \mathcal{U}$ and $V \in \mathcal{V}$.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{emb}$ does not exist.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{ctr}$ is given by $X \times Y$ together with $d\big((x, y), (x', y')\big) = \max\{d_X(x, x'), d_Y(y, y')\}$.
- The product of (X, d_X) and (Y, d_Y) in $\mathcal{Met}_{top}$ can also be given by $X \times Y$ together with the Euclidean product

$$d\big((x, y), (x', y')\big) = \sqrt{d_X^2(x, x') + d_Y^2(y, y')}$$

- Categories of models of algebraic theories.

Infinite Cartesian product

Infinite Cartesian product

The product of a family $(A_i)_{i\in\mathcal{I}}$ of objects of a category $\mathcal{C}$, when it exists, is an object

$$\prod_i A_i$$

Infinite Cartesian product

The product of a family $(A_i)_{i\in\mathcal{I}}$ of objects of a category $\mathcal{C}$, when it exists, is an object

$$\prod_i A_i$$

together with projections

$$\pi_{A_j} : \prod_i A_i \longrightarrow A_j$$

Infinite Cartesian product

The product of a family $(A_i)_{i \in \mathcal{I}}$ of objects of a category $\mathcal{C}$, when it exists, is an object

$$\prod_i A_i$$

together with projections

$$\pi_{A_j} : \prod_i A_i \longrightarrow A_j$$

such that the next mapping is a bijection.

$$\begin{array}{ccc} \mathcal{C}(X, \prod_i A_i) & \longrightarrow & \prod_i \mathcal{C}(X, A_i) \\ h & \longmapsto & (\, \pi_{A_i} \circ h \,) \end{array}$$

Infinite Cartesian product

The product of a family $(A_i)_{i\in\mathcal{I}}$ of objects of a category $\mathcal{C}$, when it exists, is an object

$$\prod_i A_i$$

together with projections

$$\pi_{A_j} : \prod_i A_i \longrightarrow A_j$$

such that the next mapping is a bijection.

$$\begin{array}{ccc} \mathcal{C}(X, \prod_i A_i) & \longrightarrow & \prod_i \mathcal{C}(X, A_i) \\ h & \longmapsto & (\ \pi_{A_i} \circ h\ ) \end{array}$$

Infinite products of directed circle does not exist in $\mathcal{L}po$.

Turning discrete models into geometric ones

Canonical partition

Canonical partition

$$G : A \overset{\partial^{+}}{\underset{\partial^{-}}{\rightrightarrows}} V$$

Canonical partition

$$G : A \underset{\partial^-}{\overset{\partial^+}{\rightrightarrows}} V \qquad\qquad \upharpoonleft G \downharpoonright \;=\; V \;\sqcup\; A \times \,]0,1[$$

Canonical partition

$$G : A \underset{\partial^-}{\overset{\partial^+}{\rightrightarrows}} V \qquad\qquad \lceil G \rfloor \;=\; V \;\sqcup\; A \times \,]0,1[$$

$$\lceil G_1 \rfloor \times \cdots \times \lceil G_n \rfloor \;=\; (\; V_1 \sqcup A_1 \times]0,1[\;) \;\times\; \cdots \;\times\; (\; V_n \sqcup A_n \times]0,1[\;)$$

Canonical partition

$$G : A \underset{\partial^-}{\overset{\partial^+}{\rightrightarrows}} V \qquad\qquad \rceil G \lfloor \;=\; V \;\sqcup\; A \times \,]0,1[$$

$$\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor \;=\; (\; V_1 \sqcup A_1 \times]0,1[\;) \;\times\; \cdots \;\times\; (\; V_n \sqcup A_n \times]0,1[\;)$$

$$\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor \;=\; \bigsqcup_{\substack{\text{points } p \text{ of} \\ G_1, \ldots, G_n}} \{p\} \times]0,1[^{\,\dim(p_1,\ldots,p_n)}$$

where $p = (p_1, \ldots, p_n)$, $p_i \in V_i \sqcup A_i$, and $\dim p = \#\big\{ i \in \{1, \ldots, n\} \mid p_i \in A_i \big\}$

Canonical partition

$$G : A \overset{\partial^+}{\underset{\partial^-}{\rightrightarrows}} V \qquad \lceil G \rfloor \; = \; V \; \sqcup \; A \times \,]0,1[$$

$$\lceil G_1 \rfloor \times \cdots \times \lceil G_n \rfloor \;=\; (\, V_1 \sqcup A_1 \times]0,1[\,) \;\times\; \cdots \;\times\; (\, V_n \sqcup A_n \times]0,1[\,)$$

$$\lceil G_1 \rfloor \times \cdots \times \lceil G_n \rfloor \;=\; \bigsqcup_{\substack{\text{points } p \text{ of} \\ G_1, \ldots, G_n}} \{p\} \times]0,1[^{\,\dim(p_1,\ldots,p_n)}$$

where $p = (p_1, \ldots, p_n)$, $p_i \in V_i \sqcup A_i$, and $\dim p = \#\big\{ i \in \{1, \ldots, n\} \mid p_i \in A_i \big\}$

$B_p = \{p\} \times]0,1[^{\,\dim(p_1,\ldots,p_n)}$ is called a canonical block

Canonical partition

$$G : A \overset{\partial^+}{\underset{\partial^-}{\rightrightarrows}} V \qquad\qquad \lceil G \rfloor \;=\; V \;\sqcup\; A \times \,]0,1[$$

$$\lceil G_1 \rfloor \times \cdots \times \lceil G_n \rfloor \;=\; (\, V_1 \sqcup A_1 \times]0,1[\,) \;\times\; \cdots \;\times\; (\, V_n \sqcup A_n \times]0,1[\,)$$

$$\lceil G_1 \rfloor \times \cdots \times \lceil G_n \rfloor \;=\; \bigsqcup_{\substack{\text{points } p \text{ of} \\ G_1,\ldots,G_n}} \{p\} \times]0,1[^{\,\dim(p_1,\ldots,p_n)}$$

where $p = (p_1, \ldots, p_n)$, $p_i \in V_i \sqcup A_i$, and $\dim p = \#\big\{ i \in \{1, \ldots, n\} \mid p_i \in A_i \big\}$

$B_p = \{p\} \times]0,1[^{\,\dim(p_1,\ldots,p_n)}$ is called a canonical block

The collection of canonical blocks forms the canonical partition of $\lceil G_1 \rfloor \times \cdots \times \lceil G_n \rfloor$.

The geometric model of a conservative program

The geometric model of a conservative program

The forbidden region of a conservative program $\Pi = (G_1, \ldots, G_n)$ is the disjoint union of canonical blocks

$$\bigsqcup_{\substack{\text{forbidden points } p \\ \text{of } (G_1, \ldots, G_n)}} B_p$$

The geometric model of a conservative program

The forbidden region of a conservative program $\Pi = (G_1, \ldots, G_n)$ is the disjoint union of canonical blocks

$$\bigsqcup_{\substack{\text{forbidden points } p \\ \text{of } (G_1, \ldots, G_n)}} B_p$$

The geometric model of Π is the locally ordered metric space

$$\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright \setminus \{\text{forbidden region}\}$$

The geometric model of a conservative program

The forbidden region of a conservative program $\Pi = (G_1, \ldots, G_n)$ is the disjoint union of canonical blocks

$$\bigsqcup_{\substack{\text{forbidden points } p \\ \text{of } (G_1, \ldots, G_n)}} B_p$$

The geometric model of Π is the locally ordered metric space

$$\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright \setminus \{\text{forbidden region}\}$$

the distance being given by

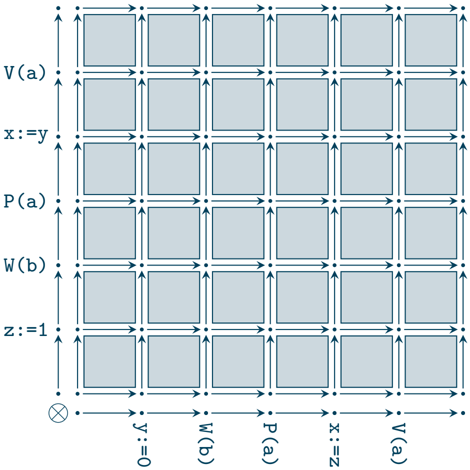
$$d(p, p') \quad = \quad \max \left\{ d_{\upharpoonleft G_i \downharpoonright}(p_i, p'_i) \mid i \in \{1, \ldots, n\} \right\}$$

in accordance with the fact that the execution time of the simultaneous execution of many processes is the longest execution time among that of the processes considered individually.

Gallery of examples

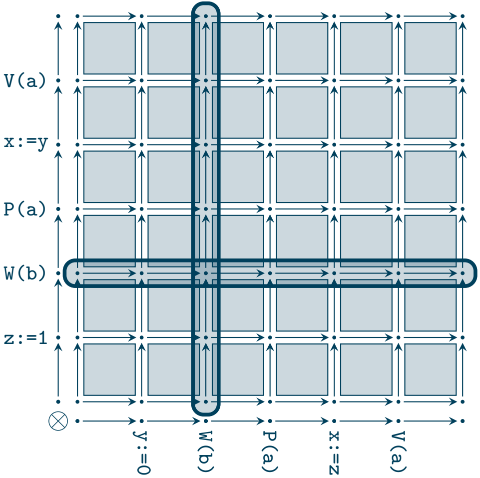
From discrete to continuous

`sem: 1 a` `sync: 1 b`



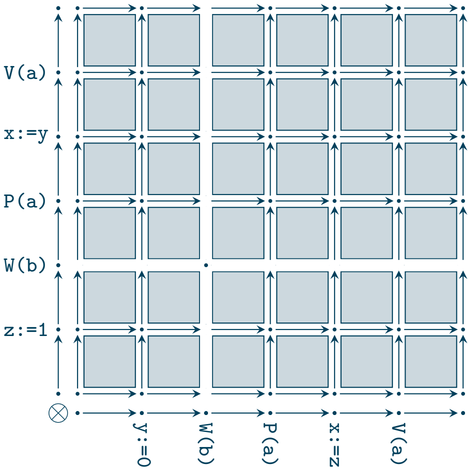
From discrete to continuous

`sem: 1 a` `sync: 1 b`



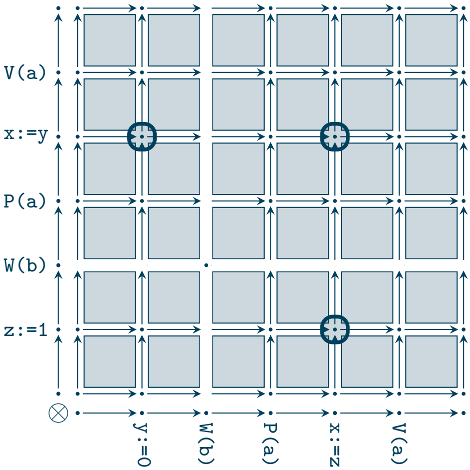
From discrete to continuous

sem: 1 a sync: 1 b



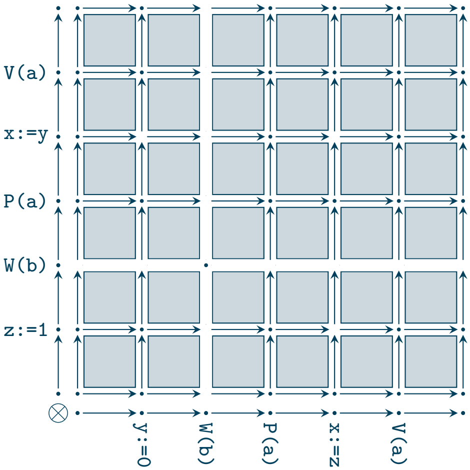
From discrete to continuous

`sem: 1 a` `sync: 1 b`



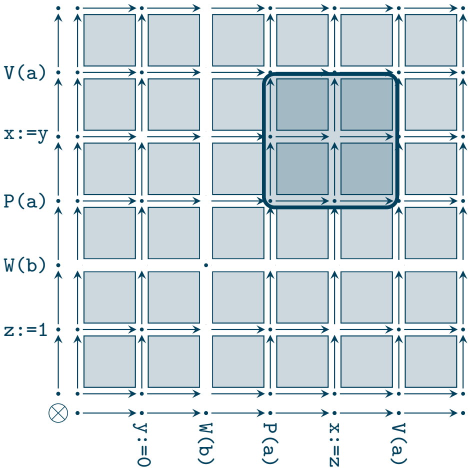
From discrete to continuous

`sem:  1 a        sync:  1 b`



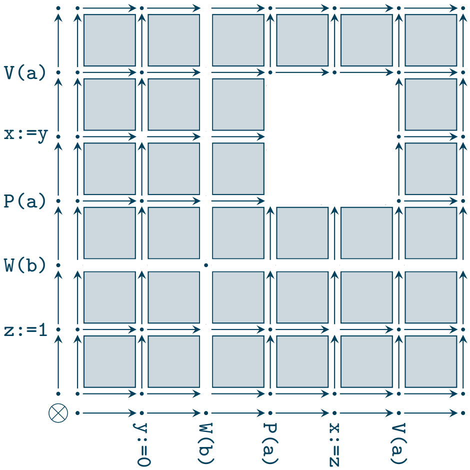
From discrete to continuous

`sem: 1 a` `sync: 1 b`



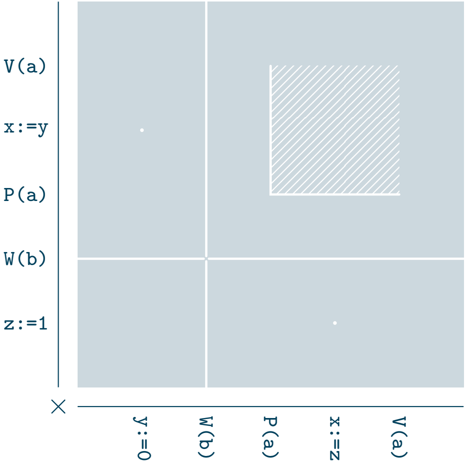
From discrete to continuous

`sem: 1 a        sync: 1 b`



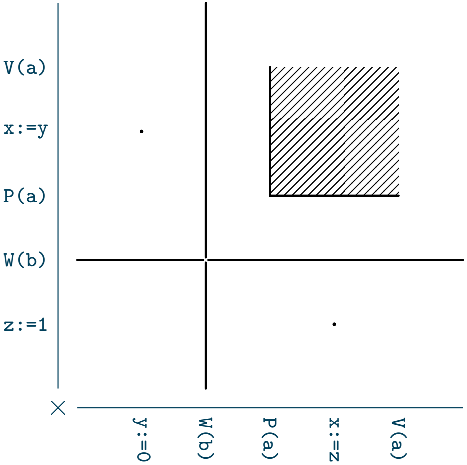
From discrete to continuous

`sem: 1 a` `sync: 1 b`



From discrete to continuous

sem: 1 a sync: 1 b



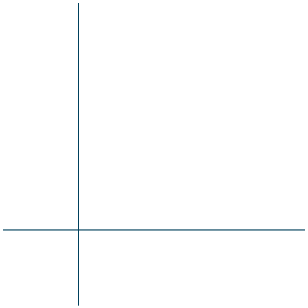
Square

Square

```
sem 1 a
proc:  p = P(a);V(a)
init:  2p
```

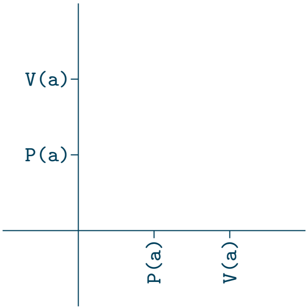
Square

```
sem 1 a
proc:  p = P(a);V(a)
init:  2p
```



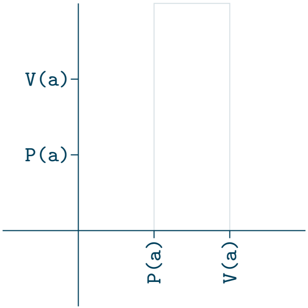
Square

```
sem 1 a
proc:  p = P(a);V(a)
init:  2p
```



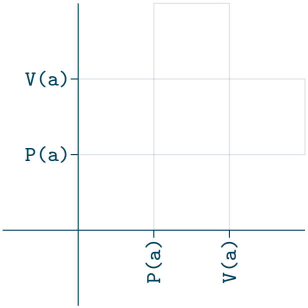
Square

```
sem 1 a
proc:  p = P(a);V(a)
init:  2p
```



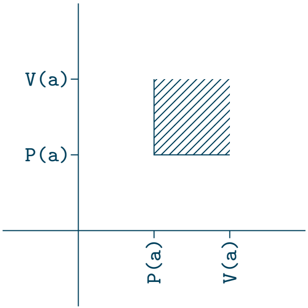
Square

```
sem 1 a
proc:  p = P(a);V(a)
init:  2p
```



Square

```
sem 1 a
proc:  p = P(a);V(a)
init:  2p
```



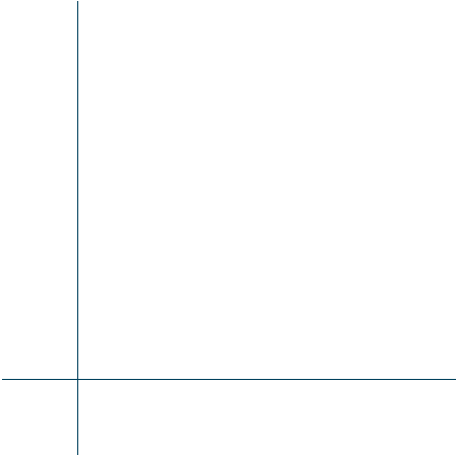
Swiss Cross

Swiss Cross

```
sem 1 a b
proc:
p = P(a);P(b);V(b);V(a)
q = P(b);P(a);V(a);V(b)
init:  p q
```

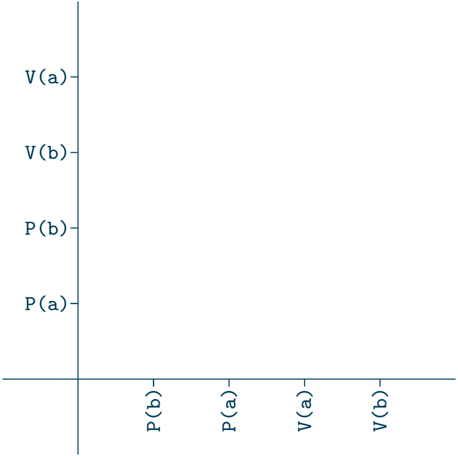
Swiss Cross

```
sem 1 a b
proc:
p = P(a);P(b);V(b);V(a)
q = P(b);P(a);V(a);V(b)
init:  p q
```



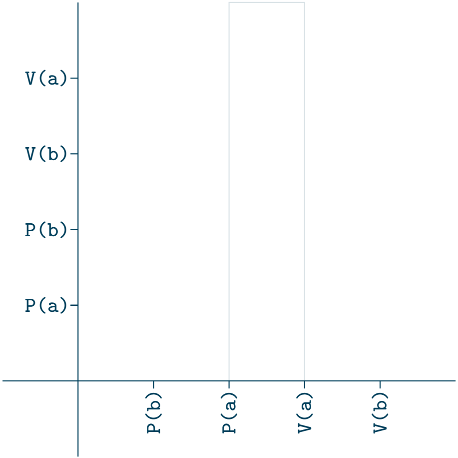
Swiss Cross

```
sem 1 a b
proc:
p = P(a);P(b);V(b);V(a)
q = P(b);P(a);V(a);V(b)
init:  p q
```



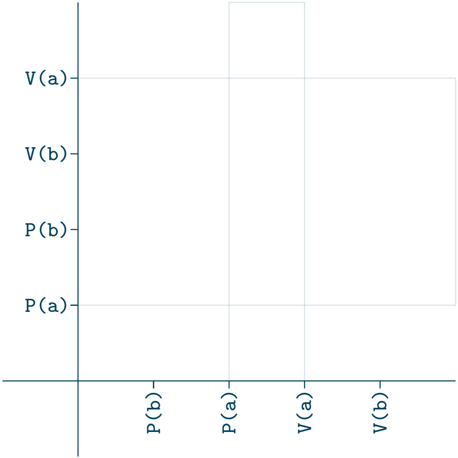
Swiss Cross

```
sem 1 a b
proc:
p = P(a);P(b);V(b);V(a)
q = P(b);P(a);V(a);V(b)
init:  p q
```



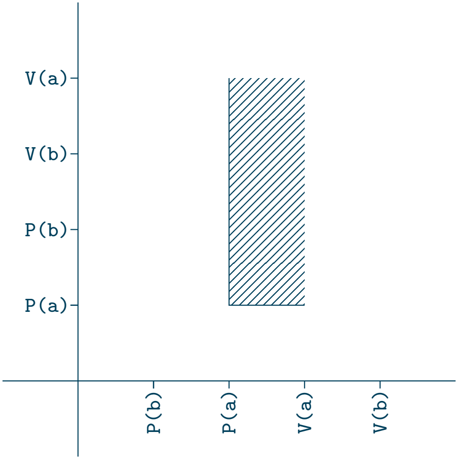
Swiss Cross

```
sem 1 a b
proc:
p = P(a);P(b);V(b);V(a)
q = P(b);P(a);V(a);V(b)
init:  p q
```



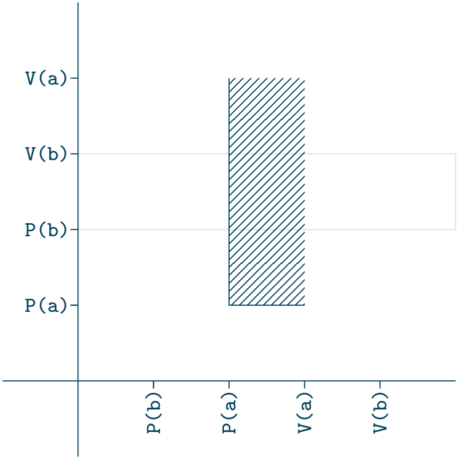
Swiss Cross

```
sem 1 a b
proc:
p = P(a);P(b);V(b);V(a)
q = P(b);P(a);V(a);V(b)
init:  p q
```



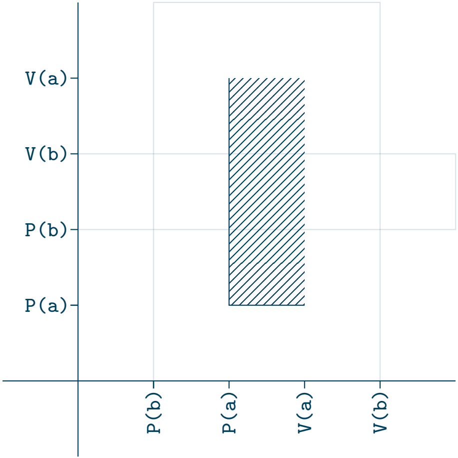
Swiss Cross

```
sem 1 a b
proc:
p = P(a);P(b);V(b);V(a)
q = P(b);P(a);V(a);V(b)
init:  p q
```



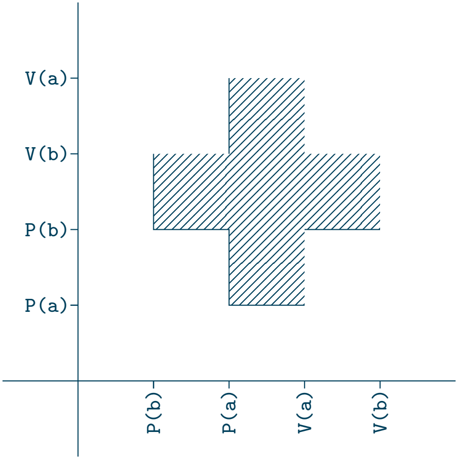
Swiss Cross

```
sem 1 a b
proc:
p = P(a);P(b);V(b);V(a)
q = P(b);P(a);V(a);V(b)
init:  p q
```



Swiss Cross

```
sem 1 a b
proc:
p = P(a);P(b);V(b);V(a)
q = P(b);P(a);V(a);V(b)
init:  p q
```



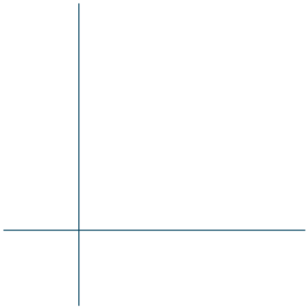
Binary synchronization

Binary synchronization

```
sync 1 a
proc:  p = W(a)
init:  2p
```

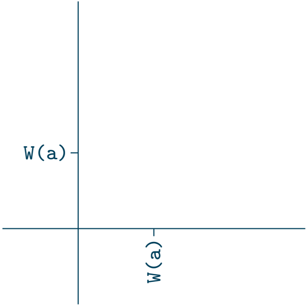
Binary synchronization

```
sync 1 a
proc:  p = W(a)
init:  2p
```



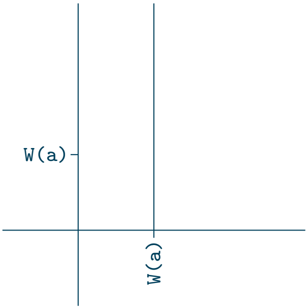
Binary synchronization

```
sync 1 a
proc:  p = W(a)
init:  2p
```



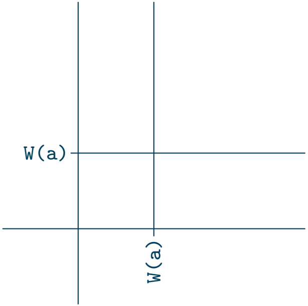
Binary synchronization

```
sync 1 a
proc:  p = W(a)
init:  2p
```



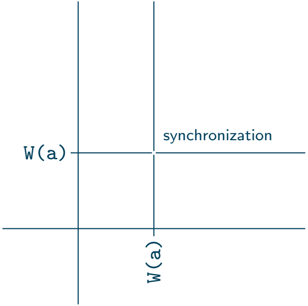
Binary synchronization

```
sync 1 a
proc:  p = W(a)
init:  2p
```



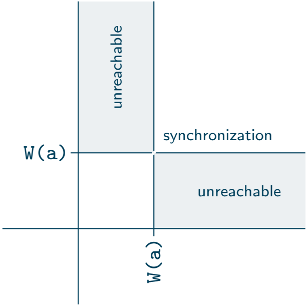
Binary synchronization

```
sync 1 a
proc:  p = W(a)
init:  2p
```



Binary synchronization

```
sync 1 a
proc:  p = W(a)
init:  2p
```



Producer/Consumer

nonlooping

Producer/Consumer

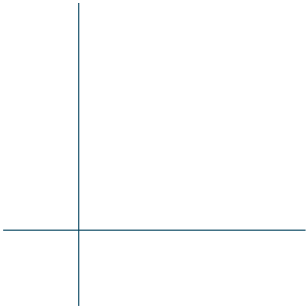
nonlooping

```
sync 1 a
proc:
  p = x:=x+1 ; W(a)
  c = W(a)   ; x:=x-1
init:  p c
```

Producer/Consumer

nonlooping

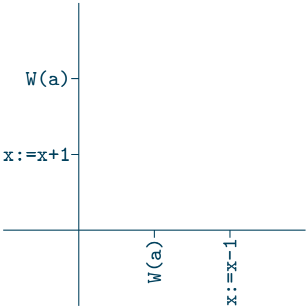
```
sync 1 a
proc:
  p = x:=x+1 ; W(a)
  c = W(a)   ; x:=x-1
init:  p c
```



Producer/Consumer

nonlooping

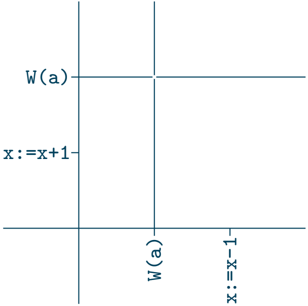
```
sync 1 a
proc:
  p = x:=x+1 ; W(a)
  c = W(a)   ; x:=x-1
init:  p c
```



Producer/Consumer

nonlooping

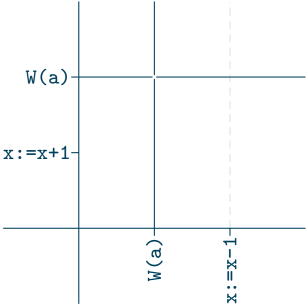
```
sync 1 a
proc:
  p = x:=x+1 ; W(a)
  c = W(a)   ; x:=x-1
init:  p c
```



Producer/Consumer

nonlooping

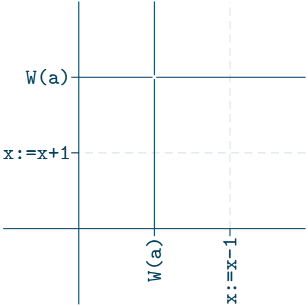
```
sync 1 a
proc:
  p = x:=x+1 ; W(a)
  c = W(a)   ; x:=x-1
init:  p c
```



Producer/Consumer

nonlooping

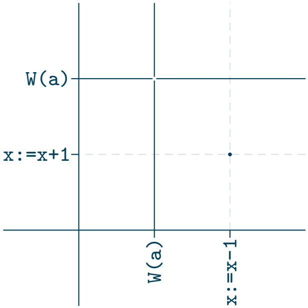
```
sync 1 a
proc:
  p = x:=x+1 ; W(a)
  c = W(a)   ; x:=x-1
init:  p c
```



Producer/Consumer

nonlooping

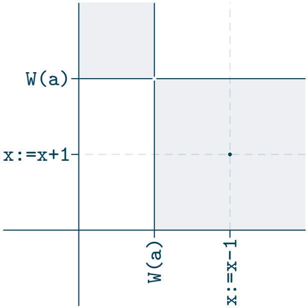
```
sync 1 a
proc:
  p = x:=x+1 ; W(a)
  c = W(a)   ; x:=x-1
init:  p c
```



Producer/Consumer

nonlooping

```
sync 1 a
proc:
  p = x:=x+1 ; W(a)
  c = W(a)   ; x:=x-1
init:  p c
```



Producer/Consumer

looping

Producer/Consumer

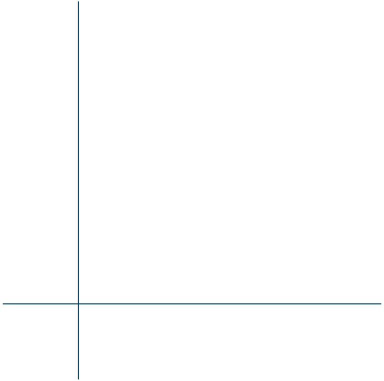
looping

```
sync 1 a b
proc:
  p = x:=x+1 ; W(a)   ; W(b) ; J(p)
  c = W(a)   ; x:=x-1 ; W(b) ; J(c)
init:  p c
```

Producer/Consumer

looping

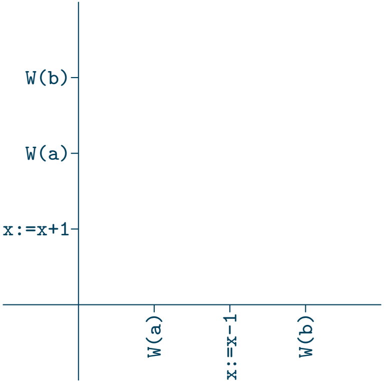
```
sync 1 a b
proc:
  p = x:=x+1 ; W(a)   ; W(b) ; J(p)
  c = W(a)   ; x:=x-1 ; W(b) ; J(c)
init:  p c
```



Producer/Consumer

looping

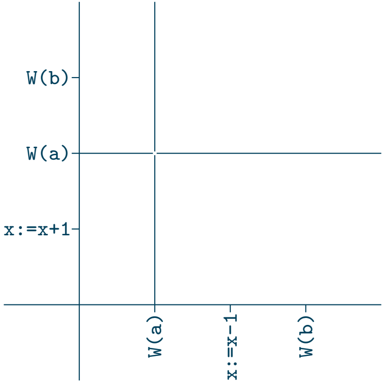
```
sync 1 a b
proc:
  p = x:=x+1 ; W(a)   ; W(b) ; J(p)
  c = W(a)   ; x:=x-1 ; W(b) ; J(c)
init:  p c
```



Producer/Consumer

looping

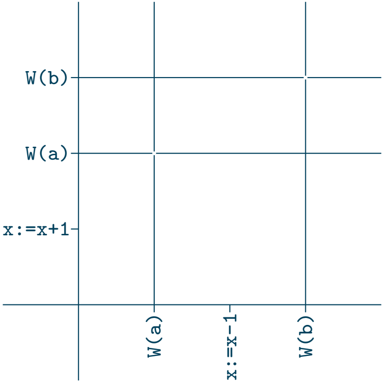
```
sync 1 a b
proc:
  p = x:=x+1 ; W(a)   ; W(b) ; J(p)
  c = W(a)   ; x:=x-1 ; W(b) ; J(c)
init:  p c
```



Producer/Consumer

looping

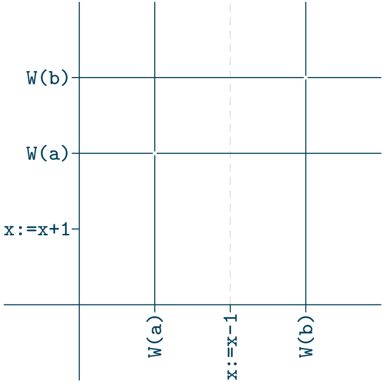
```
sync 1 a b
proc:
  p = x:=x+1 ; W(a)   ; W(b) ; J(p)
  c = W(a)   ; x:=x-1 ; W(b) ; J(c)
init:  p c
```



Producer/Consumer

looping

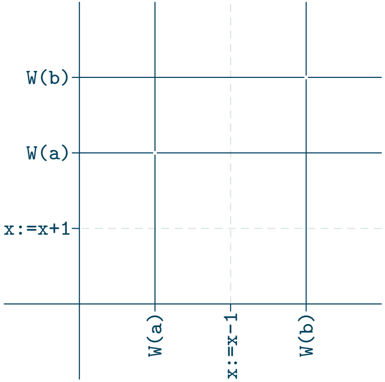
```
sync 1 a b
proc:
  p = x:=x+1 ; W(a)   ; W(b) ; J(p)
  c = W(a)   ; x:=x-1 ; W(b) ; J(c)
init:  p c
```



Producer/Consumer

looping

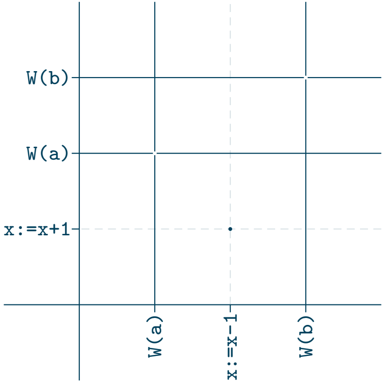
```
sync 1 a b
proc:
  p = x:=x+1 ; W(a)    ; W(b) ; J(p)
  c = W(a)   ; x:=x-1 ; W(b) ; J(c)
init:  p c
```



Producer/Consumer

looping

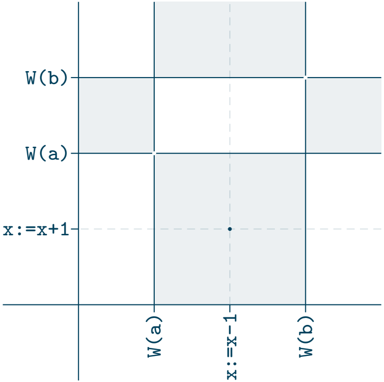
```
sync 1 a b
proc:
  p = x:=x+1 ; W(a)   ; W(b) ; J(p)
  c = W(a)   ; x:=x-1 ; W(b) ; J(c)
init:  p c
```



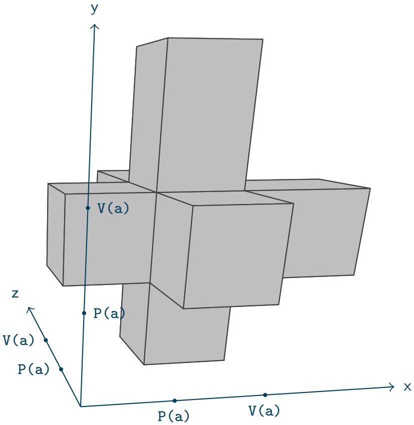
Producer/Consumer

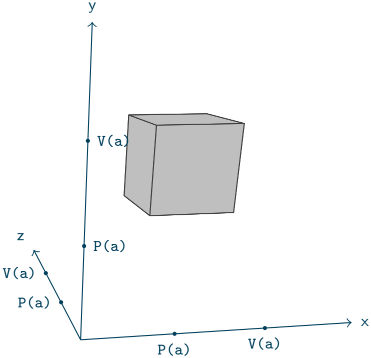
looping

```
sync 1 a b
proc:
  p = x:=x+1 ; W(a)   ; W(b) ; J(p)
  c = W(a)   ; x:=x-1 ; W(b) ; J(c)
init:  p c
```

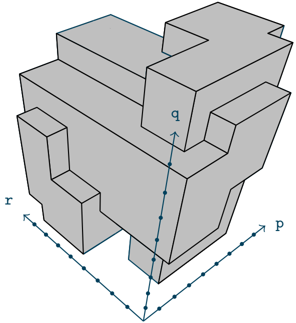


3D Swiss Cross (tetrahemihexacron) and floating cube





The Lipski algorithm




```
sem 1:  u v w x y z
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
```

Geometric vs Discrete

Justifying de definition of discrete directed paths

Justifying de definition of discrete directed paths

Let B_p and $B_{p'}$ be canonical blocks.

Justifying de definition of discrete directed paths

Let B_p and $B_{p'}$ be canonical blocks.

If there exists a directed path starting in B_p, ending in $B_{p'}$, and whose image is contained in $B_p \cup B_{p'}$ then one of the following facts is satisfied:

Justifying de definition of discrete directed paths

Let B_p and $B_{p'}$ be canonical blocks.

If there exists a directed path starting in B_p, ending in $B_{p'}$, and whose image is contained in $B_p \cup B_{p'}$ then one of the following facts is satisfied:

- for all $i \in \{1, \ldots, n\}$, $p_i = p'_i$ or p_i is the source of the arrow p'_i,

Justifying de definition of discrete directed paths

Let B_p and $B_{p'}$ be canonical blocks.

If there exists a directed path starting in B_p, ending in $B_{p'}$, and whose image is contained in $B_p \cup B_{p'}$ then one of the following facts is satisfied:

- for all $i \in \{1, \ldots, n\}$, $p_i = p'_i$ or p_i is the source of the arrow p'_i, or
- for all $i \in \{1, \ldots, n\}$, $p_i = p'_i$ or p'_i is the target of the arrow p_i.

Discretization and lifting

Discretization and lifting

- Given a directed path γ on the local pospace $\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright$ we have a finite partition $I_0 < \cdots < I_N$ of $\mathrm{dom}(\gamma)$ such that for all $k \in \{0, \ldots, N\}$, there exists a (necessarily unique) point p^k such that $\gamma(I_k) \subseteq B_{p^k}$.

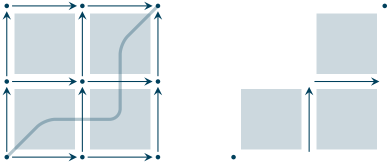
Discretization and lifting

- Given a directed path γ on the local pospace $\lceil G_1 \rfloor \times \cdots \times \lceil G_n \rfloor$ we have a finite partition $I_0 < \cdots < I_N$ of $\mathrm{dom}(\gamma)$ such that for all $k \in \{0, \ldots, N\}$, there exists a (necessarily unique) point p^k such that $\gamma(I_k) \subseteq B_{p^k}$.
- The sequence $p^0, \ldots, p^N$ is a directed path on $(G_1, \ldots, G_n)$, it is called the discretization of γ and denoted by $D(\gamma)$.

Discretization and lifting

- Given a directed path γ on the local pospace $\lceil G_1 \rfloor \times \cdots \times \lceil G_n \rfloor$ we have a finite partition $I_0 < \cdots < I_N$ of $\mathrm{dom}(\gamma)$ such that for all $k \in \{0, \ldots, N\}$, there exists a (necessarily unique) point p^k such that $\gamma(I_k) \subseteq B_{p^k}$.
- The sequence $p^0, \ldots, p^N$ is a directed path on $(G_1, \ldots, G_n)$, it is called the discretization of γ and denoted by $D(\gamma)$.
- Given a directed path δ on $(G_1, \ldots, G_n)$ there exists a directed path γ on $\lceil G_1 \rfloor \times \cdots \times \lceil G_n \rfloor$ whose discretization is δ, such a directed path γ is said to be a lifting of δ.

Example of discretization



Admissible directed paths and execution traces

on $\lceil G_1 \rfloor \times \cdots \times \lceil G_n \rfloor$

Admissible directed paths and execution traces

on $\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft$

The sequence of multi-instructions of a directed path γ on $\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft$ is that of its discretization of $D(\gamma)$.

Admissible directed paths and execution traces

on $\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft$

The sequence of multi-instructions of a directed path γ on $\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft$ is that of its discretization of $D(\gamma)$.

A directed path on $\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft$ is admissible (resp. an execution trace) iff so is its discretization.

Admissible directed paths and execution traces

on $\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft$

The sequence of multi-instructions of a directed path γ on $\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft$ is that of its discretization of $D(\gamma)$.

A directed path on $\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft$ is admissible (resp. an execution trace) iff so is its discretization.

The action of a directed path γ on $\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft$ on the right of a state σ is that of its discretization of $D(\gamma)$.

Example

```
var x = 0
var y = 0
var z = 0
sync 1 b
sem 1 a
```

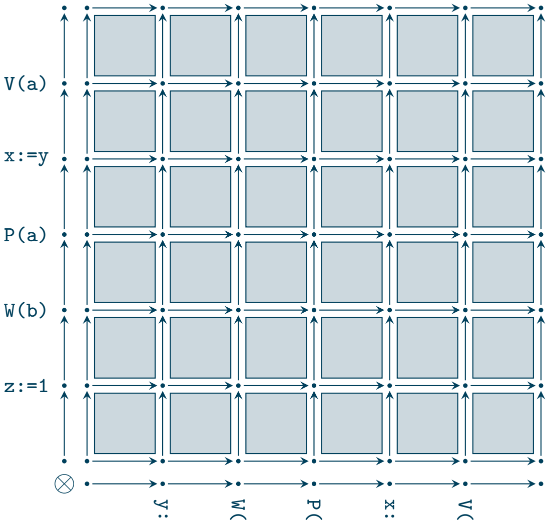
```
proc p = y:=0 ; W(b) ; P(a) ; x:=z ; V(a)

proc q = z:=1 ; W(b) ; P(a) ; x:=y ; V(a)
```

```
init p q
```

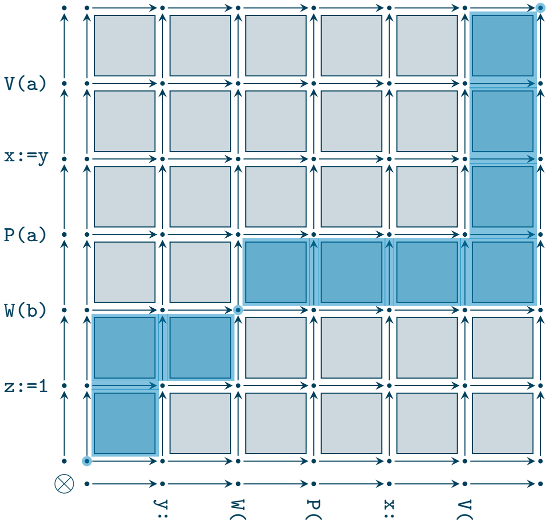
Discretization of an execution trace

`sem: 1 a` `sync: 1 b`



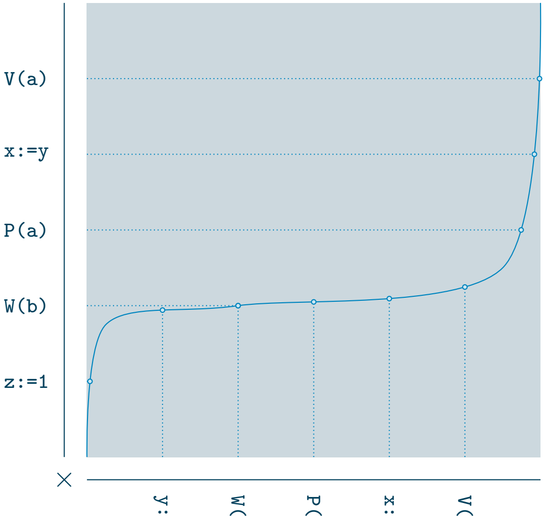
Discretization of an execution trace

sem: 1 a   sync: 1 b



Discretization of an execution trace

sem: 1 a sync: 1 b



Potential function on $\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright$

Potential function on $\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft$

If the program under consideration is conservative, then we have the potential function

$$F : \upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft \times \mathcal{S} \to \{\text{multisets over } \{1, \ldots, n\}\}$$

Potential function on $\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright$

If the program under consideration is conservative, then we have the potential function

$$F : \upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright \times \mathcal{S} \to \{\text{multisets over } \{1, \ldots, n\}\}$$

The function F is constant on each canonical block B_p,

Potential function on $\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright$

If the program under consideration is conservative, then we have the potential function

$$F : \upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright \times \mathcal{S} \to \{\text{multisets over } \{1, \ldots, n\}\}$$

The function F is constant on each canonical block B_p, its value is given by $\tilde{F}(p)$ where $\tilde{F}$ denotes the “discrete” potential function.

Geometric models are sound and complete

Geometric models are sound and complete

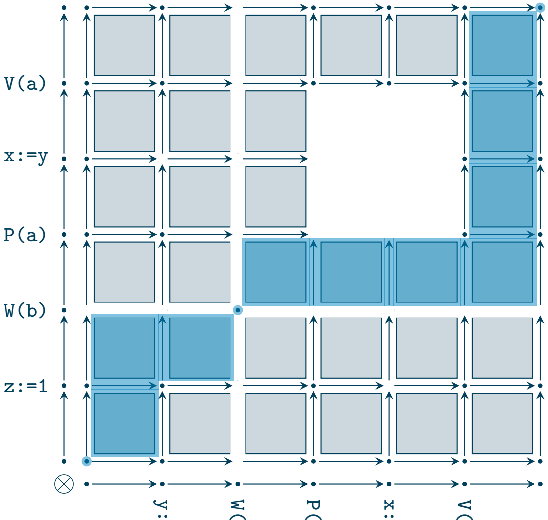
- Any directed path on a continuous model is admissible.

Geometric models are sound and complete

- Any directed path on a continuous model is admissible.
- Conversely, for each admissible path on a continuous model which meets a forbidden point, there exists a directed path which avoids them and such that both directed paths induce the same sequence of multi-instructions.

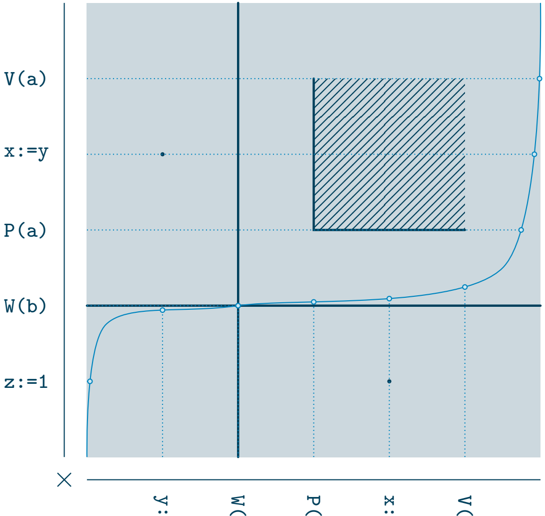
Directed paths on the geometric model are admissible

`sem:  1 a        sync:  1 b`



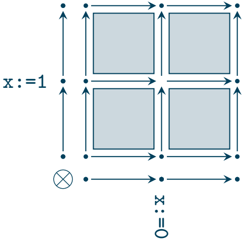
Directed paths on the geometric model are admissible

sem: 1 a   sync: 1 b



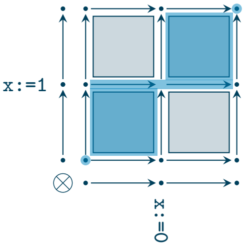
Continuous replacement

sem: 1 a sync: 1 b



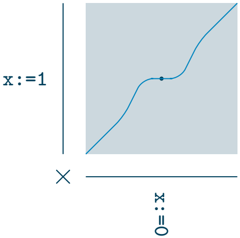
Continuous replacement

`sem:  1 a        sync:  1 b`



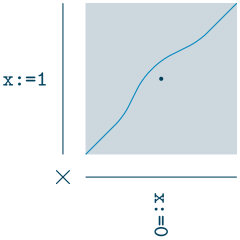
Continuous replacement

sem: 1 a sync: 1 b



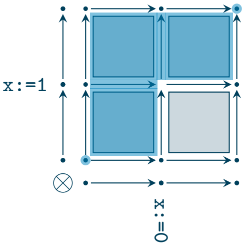
Continuous replacement

`sem: 1 a` `sync: 1 b`



Continuous replacement

`sem: 1 a        sync: 1 b`



The motivating theorem

Trade off

More mathematics for more properties?

Trade off

More mathematics for more properties?

- Both discrete and geometric models are sound and complete.

Trade off

More mathematics for more properties?

- Both discrete and geometric models are sound and complete.
- The continuous models satisfy extra properties that are “naturally” expressed in terms of metrics.

Uniform distance between directed paths

Uniform distance between directed paths

Given a compact Hausdorff space K and a metric space (X, d_X), the set of continuous maps from K to X can be equipped with the uniform distance

$$d(f,g) \quad = \quad \max\{d_X(f(k), g(k)) \mid k \in K\} \ .$$

Uniform distance between directed paths

Given a compact Hausdorff space K and a metric space (X, d_X), the set of continuous maps from K to X can be equipped with the uniform distance

$$d(f, g) \quad = \quad \max\{d_X(f(k), g(k)) \mid k \in K\} \ .$$

We consider the case where $K = [0, r]$ is the domain of definition of a directed path and (X, d_X) is the geometric model of a conservative program.

The main theorem

The main theorem

Let B_p and $B_{p'}$ be canonical blocks of the geometric model X of a conservative program.

The main theorem

Let B_p and $B_{p'}$ be canonical blocks of the geometric model X of a conservative program.

Let $dX^{[0,r]}(B_p, B_{p'})$ be the set of directed paths on X whose sources and targets lie in B_p and $B_{p'}$ respectively.

The main theorem

Let B_p and $B_{p'}$ be canonical blocks of the geometric model X of a conservative program.

Let $dX^{[0,r]}(B_p, B_{p'})$ be the set of directed paths on X whose sources and targets lie in B_p and $B_{p'}$ respectively.

Let γ be an element of $dX^{[0,r]}(B_p, B_{p'})$.

The main theorem

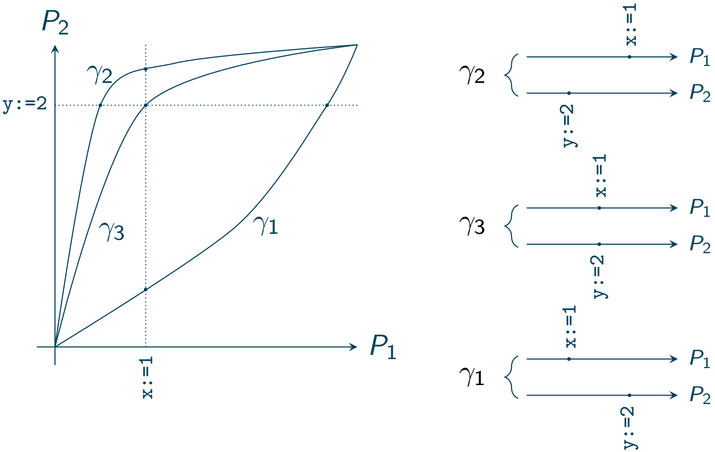
Let B_p and $B_{p'}$ be canonical blocks of the geometric model X of a conservative program.

Let $dX^{[0,r]}(B_p, B_{p'})$ be the set of directed paths on X whose sources and targets lie in B_p and $B_{p'}$ respectively.

Let γ be an element of $dX^{[0,r]}(B_p, B_{p'})$.

There exists an open ball Ω of $dX^{[0,r]}(B_p, B_{p'})$, centred in γ, such that all the elements of Ω induce the same action on valuations. Moreover, if γ is an execution trace, then so are all the elements of Ω.

Illustration



HOMOTOPY OF PATHS

The undirected case

Homotopy of paths

Homotopy of paths

Let γ and δ be two paths on X defined over the segment $[0, r]$

Homotopy of paths

Let γ and δ be two paths on X defined over the segment $[0, r]$

A homotopy from γ to δ is a continuous map h from $[0, r] \times [0, q]$ to X such that

Homotopy of paths

Let γ and δ be two paths on X defined over the segment $[0, r]$

A homotopy from γ to δ is a continuous map h from $[0, r] \times [0, q]$ to X such that

- The mappings $h(0, -) : s \in [0, q] \mapsto h(0, s)$ and $h(r, -) : s \in [0, q] \mapsto h(r, s)$ are constant

Homotopy of paths

Let γ and δ be two paths on X defined over the segment $[0, r]$

A homotopy from γ to δ is a continuous map h from $[0, r] \times [0, q]$ to X such that

- The mappings $h(0, -) : s \in [0, q] \mapsto h(0, s)$ and $h(r, -) : s \in [0, q] \mapsto h(r, s)$ are constant
- The mappings $h(-, 0) : t \in [0, r] \mapsto h(t, 0)$ and $h(-, q) : s \in [0, r] \mapsto h(t, q)$ are γ and δ

Homotopy of paths

Let γ and δ be two paths on X defined over the segment $[0, r]$

A homotopy from γ to δ is a continuous map h from $[0, r] \times [0, q]$ to X such that

- The mappings $h(0, -) : s \in [0, q] \mapsto h(0, s)$ and $h(r, -) : s \in [0, q] \mapsto h(r, s)$ are constant
- The mappings $h(-, 0) : t \in [0, r] \mapsto h(t, 0)$ and $h(-, q) : s \in [0, r] \mapsto h(t, q)$ are γ and δ

As a consequence we have $\gamma(0) = \delta(0)$ and $\gamma(r) = \delta(r)$.

Uniform distance and Curryfication

Uniform distance and Curryfication

Suppose that X is a metric space.

Uniform distance and Curryfication

Suppose that X is a metric space.

For all compact Hausdorff space K, the homset $\mathcal{Top}(K, X)$ with the (topology induced by the) uniform distance is denoted by X^K

Uniform distance and Curryfication

Suppose that X is a metric space.

For all compact Hausdorff space K, the homset $\mathcal{T}op(K, X)$ with the (topology induced by the) uniform distance is denoted by X^K

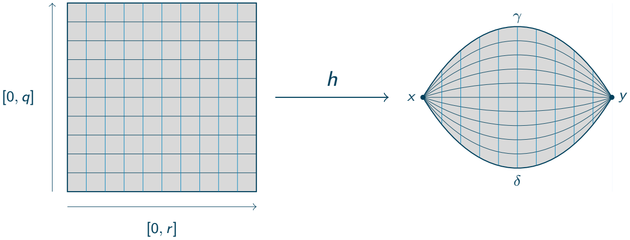
The Curryfication $(\hat{\_})$ induces a homeomorphism from $X^{[0,r]\times[0,q]}$ to $\left(X^{[0,r]}\right)^{[0,q]}$

$$(h : [0, r] \times [0, q] \to X) \to (\hat{h} : [0, q] \to X^{[0,r]})$$

The two faces of homotopies

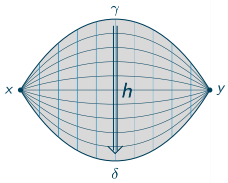
The two faces of homotopies

h is a continuous map from $[0, r] \times [0, q]$ to X i.e. $h \in \mathcal{T}op\big[[0, r] \times [0, q], X\big]$



The two faces of homotopies

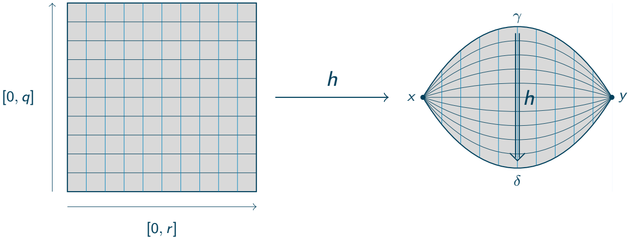
but is also a path from γ to δ in the space $X^{[0,r]}$ i.e. $h \in \mathcal{Top}\big[[0,q], X^{[0,r]}\big]$



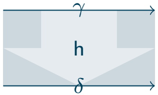
The two faces of homotopies

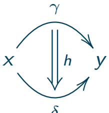
h is a continuous map from $[0, r] \times [0, q]$ to X i.e. $h \in \mathcal{Top}\big[[0, r] \times [0, q], X\big]$

but is also a path from γ to δ in the space $X^{[0,r]}$ i.e. $h \in \mathcal{Top}\big[[0, q], X^{[0,r]}\big]$



We introduce the following notation





Concatenation of homotopies

vertical composition

Concatenation of homotopies

vertical composition

Let $g : [0, r] \times [0, q'] \to X$ and $h : [0, r] \times [0, q] \to X$ be homotopies from γ to ξ and from ξ to δ.

Concatenation of homotopies

vertical composition

Let $g : [0, r] \times [0, q'] \to X$ and $h : [0, r] \times [0, q] \to X$ be homotopies from γ to ξ and from ξ to δ.

The mapping $h * g : [0, r] \times [0, q + q'] \to X$ defined by

$$h * g(t, s) = \begin{cases} g(t, s) & \text{if } 0 \leqslant s \leqslant q \\ h(t, s - q) & \text{if } q \leqslant s \leqslant q + q' \end{cases}$$

is a homotopy from γ to δ.

Concatenation of homotopies

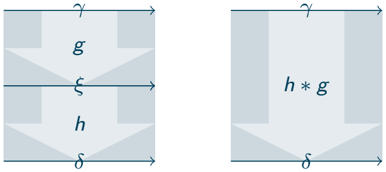
vertical composition

Let $g : [0,r] \times [0,q'] \to X$ and $h : [0,r] \times [0,q] \to X$ be homotopies from γ to ξ and from ξ to δ.

The mapping $h * g : [0,r] \times [0, q+q'] \to X$ defined by

$$h * g(t,s) \quad = \quad \begin{cases} g(t,s) & \text{if } 0 \leqslant s \leqslant q \\ h(t, s-q) & \text{if } q \leqslant s \leqslant q+q' \end{cases}$$

is a homotopy from γ to δ.



The directed case

Directed homotopy on a locally ordered space

Directed homotopy on a locally ordered space

Let $\gamma, \delta \in \mathcal{L}po([0, r], X)$ such that $\partial^{-}\gamma = \partial^{-}\delta$ and $\partial^{+}\gamma = \partial^{+}\delta$.

Directed homotopy on a locally ordered space

Let $\gamma, \delta \in \mathcal{L}po([0,r], X)$ such that $\partial^- \gamma = \partial^- \delta$ and $\partial^+ \gamma = \partial^+ \delta$.

- A directed homotopy from γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ that induces a local pospace morphism.

Directed homotopy on a locally ordered space

Let $\gamma, \delta \in \mathcal{L}po([0, r], X)$ such that $\partial^- \gamma = \partial^- \delta$ and $\partial^+ \gamma = \partial^+ \delta$.

- A directed homotopy from γ to δ is a homotopy of paths $h : [0, r] \times [0, q] \to X$ that induces a local pospace morphism.
- An anti-directed homotopy from γ to δ is a homotopy of paths $h : [0, r] \times [0, q] \to X$ such that $(t, s) \mapsto h(t, q - s)$ is a directed homotopy from δ to γ.

Directed homotopy on a locally ordered space

Let $\gamma, \delta \in \mathcal{L}po([0,r], X)$ such that $\partial^- \gamma = \partial^- \delta$ and $\partial^+ \gamma = \partial^+ \delta$.

- A directed homotopy from γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ that induces a local pospace morphism.
- An anti-directed homotopy from γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ such that $(t,s) \mapsto h(t, q-s)$ is a directed homotopy from δ to γ.
- An elementary homotopy between γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ obtained as a finite concatenation of directed homotopies and anti-directed homotopies.

Directed homotopy on a locally ordered space

Let $\gamma, \delta \in \mathcal{Lpo}([0,r], X)$ such that $\partial^{-}\gamma = \partial^{-}\delta$ and $\partial^{+}\gamma = \partial^{+}\delta$.

- A directed homotopy from γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ that induces a local pospace morphism.
- An anti-directed homotopy from γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ such that $(t,s) \mapsto h(t, q-s)$ is a directed homotopy from δ to γ.
- An elementary homotopy between γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ obtained as a finite concatenation of directed homotopies and anti-directed homotopies.
- A weakly directed homotopy from γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ whose intermediate paths $h(\_, s)$, for $s \in [0,q]$, are directed.

Directed homotopy on a locally ordered space

Let $\gamma, \delta \in \mathcal{L}po([0,r], X)$ such that $\partial^- \gamma = \partial^- \delta$ and $\partial^+ \gamma = \partial^+ \delta$.

- A directed homotopy from γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ that induces a local pospace morphism.
- An anti-directed homotopy from γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ such that $(t,s) \mapsto h(t, q-s)$ is a directed homotopy from δ to γ.
- An elementary homotopy between γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ obtained as a finite concatenation of directed homotopies and anti-directed homotopies.
- A weakly directed homotopy from γ to δ is a homotopy of paths $h : [0,r] \times [0,q] \to X$ whose intermediate paths $h(\_, s)$, for $s \in [0,q]$, are directed.
- Any elementary homotopy is a weakly directed homotopy. The converse is false.

Directed homotopy on a locally ordered space

Let $\gamma, \delta \in \mathcal{L}po([0, r], X)$ such that $\partial^- \gamma = \partial^- \delta$ and $\partial^+ \gamma = \partial^+ \delta$.

- A directed homotopy from γ to δ is a homotopy of paths $h : [0, r] \times [0, q] \to X$ that induces a local pospace morphism.
- An anti-directed homotopy from γ to δ is a homotopy of paths $h : [0, r] \times [0, q] \to X$ such that $(t, s) \mapsto h(t, q - s)$ is a directed homotopy from δ to γ.
- An elementary homotopy between γ to δ is a homotopy of paths $h : [0, r] \times [0, q] \to X$ obtained as a finite concatenation of directed homotopies and anti-directed homotopies.
- A weakly directed homotopy from γ to δ is a homotopy of paths $h : [0, r] \times [0, q] \to X$ whose intermediate paths $h(\_, s)$, for $s \in [0, q]$, are directed.
- Any elementary homotopy is a weakly directed homotopy. The converse is false.
- Each of the preceding class of homotopies is stable under concatenation.

Homotopy and dihomotopy relations

Homotopy and dihomotopy relations

Two paths γ and γ' are said to be homotopic when there exists a homotopy between them.
We have the equivalence relation $\sim_h$ between paths on a topological space.

Homotopy and dihomotopy relations

Two paths γ and γ' are said to be homotopic when there exists a homotopy between them.
We have the equivalence relation $\sim_h$ between paths on a topological space.

They are said to be dihomotopic when there exists an elementary homotopy between them.
We have the equivalence relation $\sim_d$ between directed paths on a locally ordered space.

Homotopy and dihomotopy relations

Two paths γ and γ' are said to be homotopic when there exists a homotopy between them.
We have the equivalence relation $\sim_h$ between paths on a topological space.

They are said to be dihomotopic when there exists an elementary homotopy between them.
We have the equivalence relation $\sim_d$ between directed paths on a locally ordered space.

They are said to be weakly dihomotopic when there exists a weakly directed homotopy between them. We have the equivalence relation $\sim_w$ between directed paths on a locally ordered space.

Reparametrization

Reparametrization

An increasing and surjective map $\theta : [0, r] \to [0, r]$ is called a reparametrization.

Reparametrization

An increasing and surjective map $\theta : [0, r] \to [0, r]$ is called a reparametrization.
The mapping

$$h : (t, s) \in [0, r] \times [0, 1] \mapsto \theta(t) + s \cdot (\max(t, \theta(t)) - \theta(t)) \in [0, r]$$

is a directed homotopy from θ to $\max(\mathrm{id}_{[0,r]}, \theta)$.

Reparametrization

An increasing and surjective map $\theta : [0, r] \to [0, r]$ is called a reparametrization.
The mapping

$$h : (t, s) \in [0, r] \times [0, 1] \mapsto \theta(t) + s \cdot (\max(t, \theta(t)) - \theta(t)) \in [0, r]$$

is a directed homotopy from θ to $\max(\mathrm{id}_{[0,r]}, \theta)$.

If $\gamma : [0, r] \to X$ is a directed path on the local pospace X, then $\gamma \circ h$ is a directed homotopy from $\gamma \circ \theta$ to $\gamma \circ \max(\mathrm{id}_{[0,r]}, \theta)$

Reparametrization

An increasing and surjective map $\theta : [0, r] \to [0, r]$ is called a reparametrization.
The mapping

$$h : (t, s) \in [0, r] \times [0, 1] \mapsto \theta(t) + s \cdot (\max(t, \theta(t)) - \theta(t)) \in [0, r]$$

is a directed homotopy from θ to $\max(\mathrm{id}_{[0,r]}, \theta)$.

If $\gamma : [0, r] \to X$ is a directed path on the local pospace X, then $\gamma \circ h$ is a directed homotopy from $\gamma \circ \theta$ to $\gamma \circ \max(\mathrm{id}_{[0,r]}, \theta)$

Therefore γ and $\gamma \circ \theta$ are dihomotopic.

Images of directed paths on a pospace

Images of directed paths on a pospace

Theorem

The image of a nonconstant directed path on a pospace is isomorphic to $[0, 1]$.

Images of directed paths on a pospace

Theorem

The image of a nonconstant directed path on a pospace is isomorphic to $[0, 1]$.

Corollary

Two directed paths on a posapce having the same image are dihomotopic.

Images of directed paths on a pospace

Theorem

The image of a nonconstant directed path on a pospace is isomorphic to $[0, 1]$.

Corollary

Two directed paths on a posapce having the same image are dihomotopic.

proof:

Images of directed paths on a pospace

Theorem

The image of a nonconstant directed path on a pospace is isomorphic to $[0, 1]$.

Corollary

Two directed paths on a posapce having the same image are dihomotopic.

proof:
Suppose that $\mathrm{im}(\gamma) = \mathrm{im}(\gamma')$.

Images of directed paths on a pospace

Theorem

The image of a nonconstant directed path on a pospace is isomorphic to $[0, 1]$.

Corollary

Two directed paths on a posapce having the same image are dihomotopic.

proof:
Suppose that $\mathrm{im}(\gamma) = \mathrm{im}(\gamma')$.
$\phi : [0, r] \to \mathrm{im}(\gamma)$ a pospace isomorphism.

Images of directed paths on a pospace

Theorem

The image of a nonconstant directed path on a pospace is isomorphic to $[0,1]$.

Corollary

Two directed paths on a posapce having the same image are dihomotopic.

proof:
Suppose that $\mathrm{im}(\gamma) = \mathrm{im}(\gamma')$.
$\phi : [0,r] \to \mathrm{im}(\gamma)$ a pospace isomorphism.
$\phi^{-1} \circ \gamma$ and $\phi^{-1} \circ \gamma'$ are reparametrization.

Images of directed paths on a pospace

Theorem

The image of a nonconstant directed path on a pospace is isomorphic to $[0, 1]$.

Corollary

Two directed paths on a posapce having the same image are dihomotopic.

proof:
Suppose that $\mathrm{im}(\gamma) = \mathrm{im}(\gamma')$.
$\phi : [0, r] \to \mathrm{im}(\gamma)$ a pospace isomorphism.
$\phi^{-1} \circ \gamma$ and $\phi^{-1} \circ \gamma'$ are reparametrization.
We have h an elementary homotopy from $\phi^{-1} \circ \gamma$ to $\phi^{-1} \circ \gamma'$.

Images of directed paths on a pospace

Theorem

The image of a nonconstant directed path on a pospace is isomorphic to $[0, 1]$.

Corollary

Two directed paths on a posapce having the same image are dihomotopic.

proof:
Suppose that $\mathrm{im}(\gamma) = \mathrm{im}(\gamma')$.
$\phi : [0, r] \to \mathrm{im}(\gamma)$ a pospace isomorphism.
$\phi^{-1} \circ \gamma$ and $\phi^{-1} \circ \gamma'$ are reparametrization.
We have h an elementary homotopy from $\phi^{-1} \circ \gamma$ to $\phi^{-1} \circ \gamma'$.
Hence $\phi \circ h$ is an elementary homotopy from γ and γ'.

Relation to geometric models

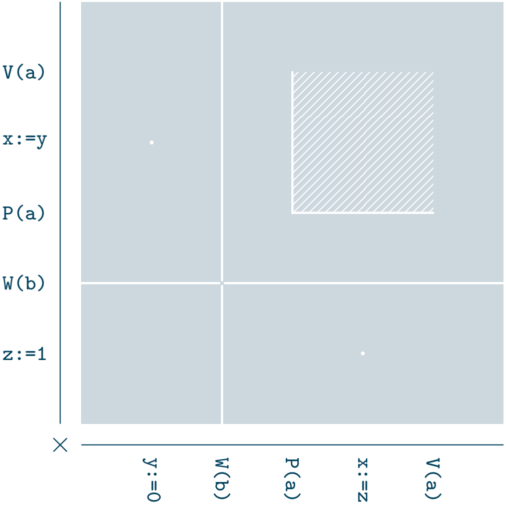
Main theorem

Main theorem

Two weakly dihomotopic paths on the geometric model of a conservative program induce the same action on valuations. Moreover, if one of them is an execution trace, then so is the other.

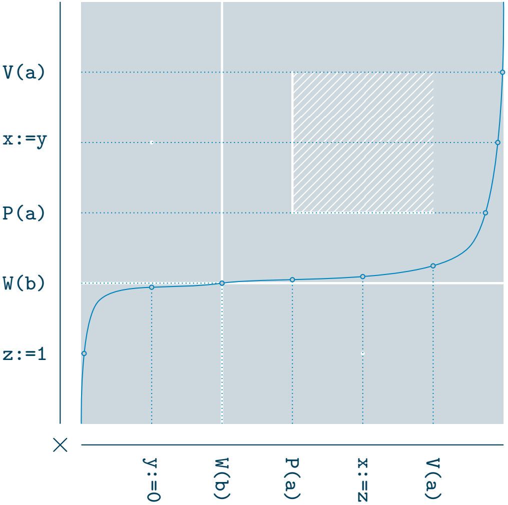
Weakly directed homotopy

`sem: 1 a` `sync: 1 b`



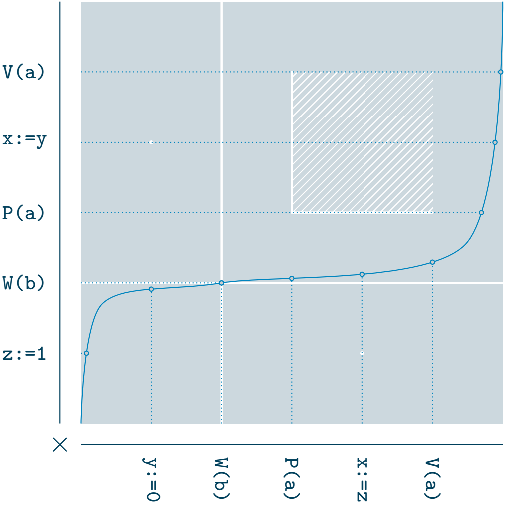
Weakly directed homotopy

`sem: 1 a` `sync: 1 b`



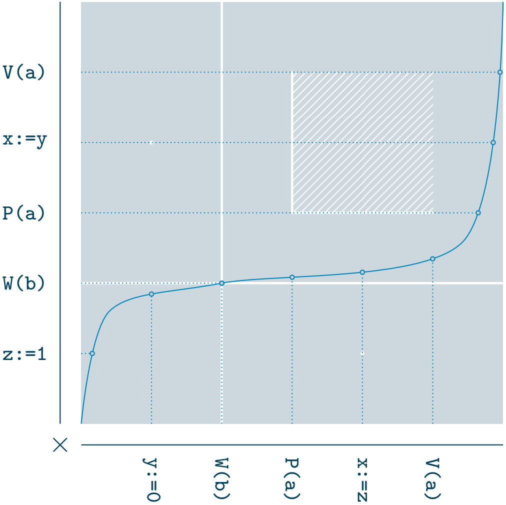
Weakly directed homotopy

`sem: 1 a` `sync: 1 b`



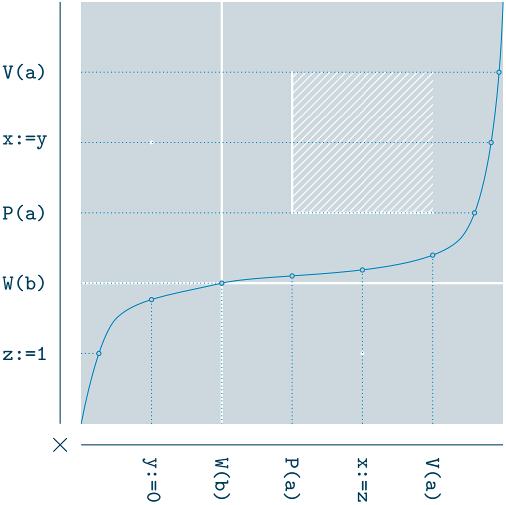
Weakly directed homotopy

`sem: 1 a        sync: 1 b`



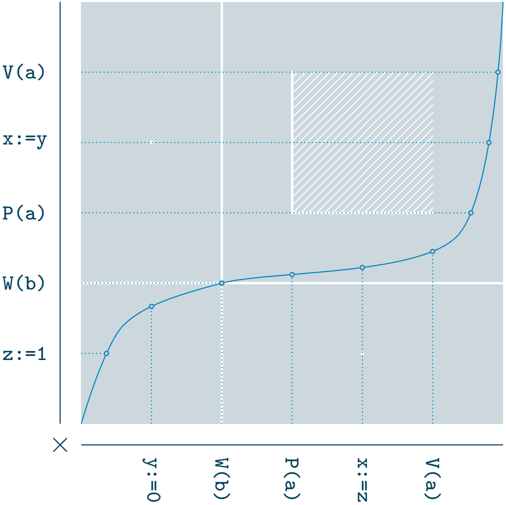
Weakly directed homotopy

`sem: 1 a` `sync: 1 b`



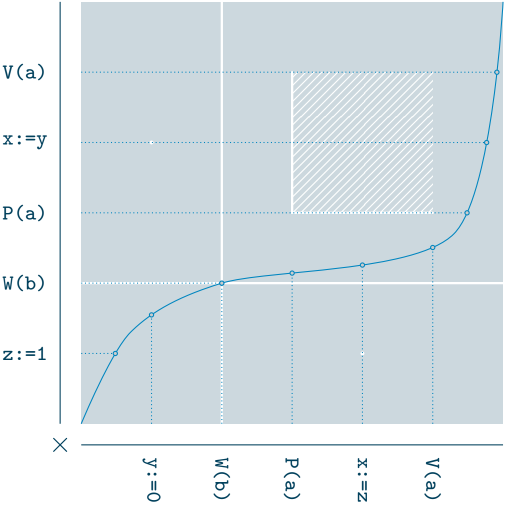
Weakly directed homotopy

`sem: 1 a        sync: 1 b`



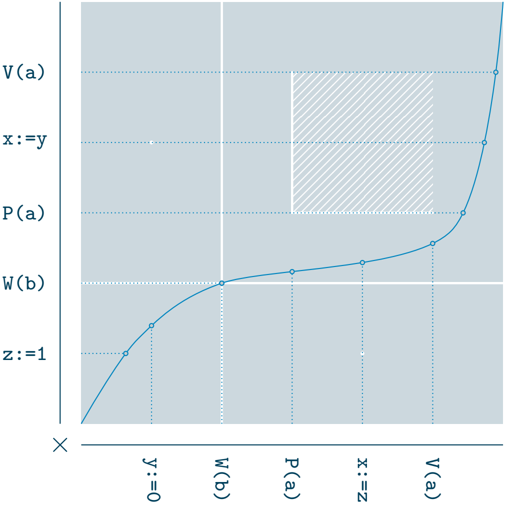
Weakly directed homotopy

`sem: 1 a` `sync: 1 b`



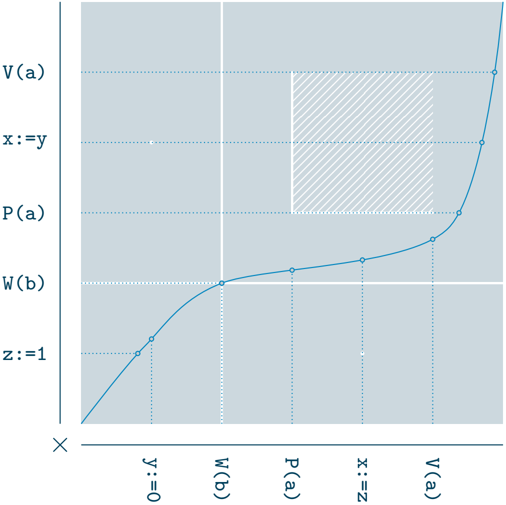
Weakly directed homotopy

`sem: 1 a` `sync: 1 b`



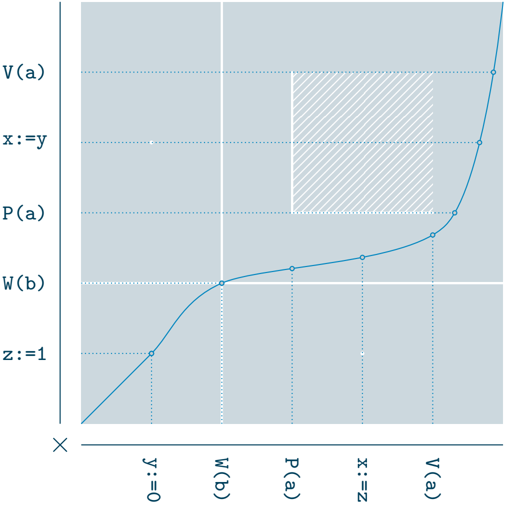
Weakly directed homotopy

`sem: 1 a` `sync: 1 b`



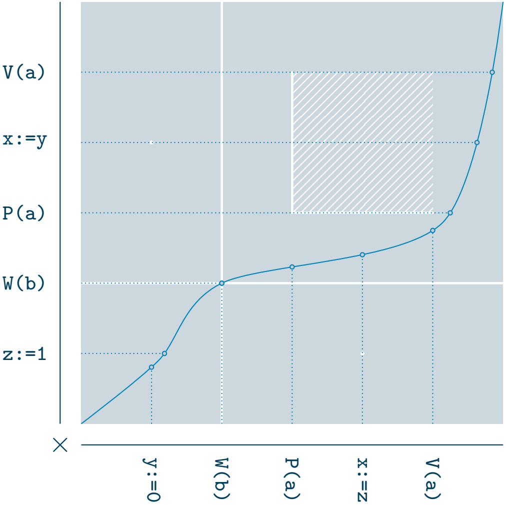
Weakly directed homotopy

sem: 1 a   sync: 1 b



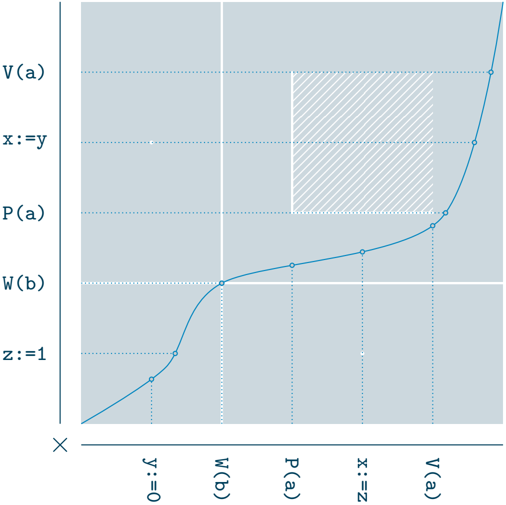
Weakly directed homotopy

`sem: 1 a` `sync: 1 b`



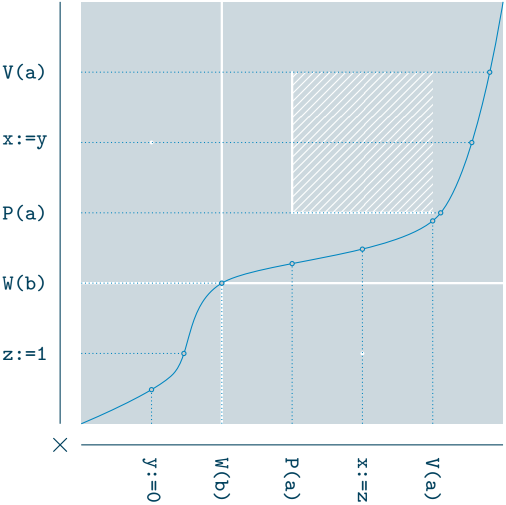
Weakly directed homotopy

```
sem:   1 a        sync:   1 b
```



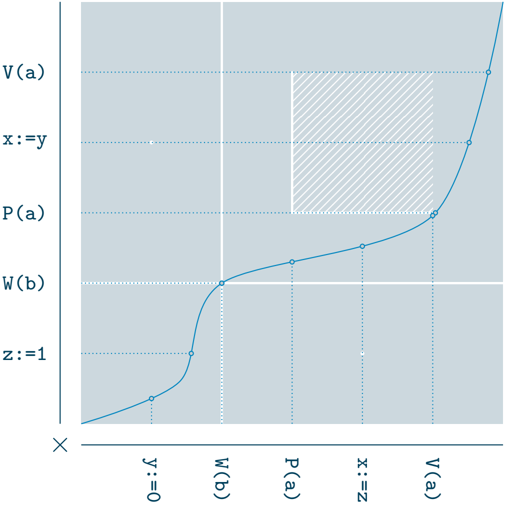
Weakly directed homotopy

`sem: 1 a` `sync: 1 b`



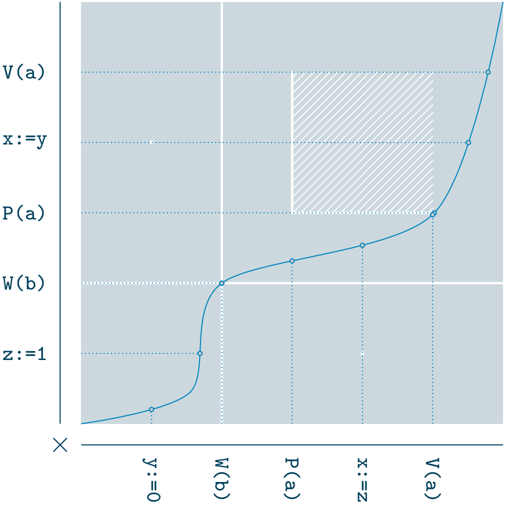
Weakly directed homotopy

sem: 1 a   sync: 1 b



Weakly directed homotopy

sem: 1 a sync: 1 b



Proof

Proof

By a standard result from general topology, the Curryfication of h

$$\hat{h} : s \in [0, q] \mapsto (t \in [0, r] \mapsto h(t, s) \in X)$$

is a continuous path on $dX^{[0,r]}(p, p')$.

Proof

By a standard result from general topology, the Curryfication of h

$$\hat{h} : s \in [0, q] \mapsto (t \in [0, r] \mapsto h(t, s) \in X)$$

is a continuous path on $dX^{[0,r]}(p, p')$.

The image of $\hat{h}$ is thus compact, so we cover it with open balls given by the main theorem of geometric models.

Proof

By a standard result from general topology, the Curryfication of h

$$\hat{h} : s \in [0, q] \mapsto (t \in [0, r] \mapsto h(t, s) \in X)$$

is a continuous path on $dX^{[0,r]}(p, p')$.

The image of $\hat{h}$ is thus compact, so we cover it with open balls given by the main theorem of geometric models.

By the Lebesgue number theorem there exists a real number $\varepsilon > 0$ such that $|s - s'| \leqslant \varepsilon$ implies that $\hat{h}(s)$ and $\hat{h}(s')$ belong to the same open ball from the covering.

Proof

By a standard result from general topology, the Curryfication of h

$$\hat{h} : s \in [0,q] \mapsto (t \in [0,r] \mapsto h(t,s) \in X)$$

is a continuous path on $dX^{[0,r]}(p,p')$.

The image of $\hat{h}$ is thus compact, so we cover it with open balls given by the main theorem of geometric models.

By the Lebesgue number theorem there exists a real number $\varepsilon > 0$ such that $|s - s'| \leqslant \varepsilon$ implies that $\hat{h}(s)$ and $\hat{h}(s')$ belong to the same open ball from the covering.

The conclusion follows considering the sequence

$$\hat{h}(0), \hat{h}(\varepsilon), \hat{h}(2\varepsilon), \hat{h}(3\varepsilon), \cdots, \hat{h}(n\varepsilon), \hat{h}(q)$$

where n is the greatest natural number such that $n\varepsilon \leqslant q$.

Programs with mutex only

a result by É. Goubault and S. Mimram

Programs with mutex only

a result by É. Goubault and S. Mimram

Let X be the geometric model of a conservative program whose semaphores have arity 1 (mutex), then two directed paths on X are dihomotopic if and only if they are homotopic.

INDEPENDENCE

Compatible programs

Compatible programs

Two programs P and Q are said to be compatible when their initial valuations and their arity maps coincide on the intersection of their domains of definition. In that case we define the parallel composition $P|Q$.

Compatible programs

Two programs P and Q are said to be compatible when their initial valuations and their arity maps coincide on the intersection of their domains of definition. In that case we define the parallel composition $P|Q$.

By extension we define the parallel composition of $P_1, \ldots, P_N$ when the programs are pairwise compatible.

Syntactical independence

Two programs are said to be syntactically independent when the set of resources they use are disjoint:

Two programs are said to be syntactically independent when the set of resources they use are disjoint:

- they have no variables in common,

Two programs are said to be syntactically independent when the set of resources they use are disjoint:

- they have no variables in common,
- they have no semaphores in common, and

Two programs are said to be syntactically independent when the set of resources they use are disjoint:

- they have no variables in common,
- they have no semaphores in common, and
- they have no barriers in common.

Two programs are said to be syntactically independent when the set of resources they use are disjoint:

- they have no variables in common,
- they have no semaphores in common, and
- they have no barriers in common.

Syntactically independent programs are compatible.

Two programs are said to be syntactically independent when the set of resources they use are disjoint:

- they have no variables in common,
- they have no semaphores in common, and
- they have no barriers in common.

Syntactically independent programs are compatible.

Syntactical independence can be decided statically,

Two programs are said to be syntactically independent when the set of resources they use are disjoint:

- they have no variables in common,
- they have no semaphores in common, and
- they have no barriers in common.

Syntactically independent programs are compatible.

Syntactical independence can be decided statically, it is compositional,

Two programs are said to be syntactically independent when the set of resources they use are disjoint:

- they have no variables in common,
- they have no semaphores in common, and
- they have no barriers in common.

Syntactically independent programs are compatible.

Syntactical independence can be decided statically, it is compositional, but it is too restrictive.

Model independence

Model Independence

Model Independence

Suppose the programs $P_1, \ldots, P_N$ are conservative.

Model Independence

Suppose the programs $P_1, \ldots, P_N$ are conservative.

The programs $P_1, \ldots, P_N$ are said to be model independent when

$$[\![P_1 | \cdots | P_N]\!] = [\![P_1]\!] \times \cdots \times [\![P_N]\!]$$

Model Independence

Suppose the programs $P_1, \ldots, P_N$ are conservative.

The programs $P_1, \ldots, P_N$ are said to be model independent when

$$[\![P_1 | \cdots | P_N]\!] = [\![P_1]\!] \times \cdots \times [\![P_N]\!]$$

Model independence can be decided statically.

Observational independence

Compatible permutations

Compatible permutations

Assume we have a partition

$$\{1, \ldots, n\} \quad = \quad S_1 \sqcup \cdots \sqcup S_N$$

Compatible permutations

Assume we have a partition

$$\{1, \ldots, n\} \quad = \quad S_1 \sqcup \cdots \sqcup S_N$$

Two multi-instructions μ and μ' ($\mathrm{dom}(\mu), \mathrm{dom}(\mu') \subseteq \{1, \ldots, n\}$) can be swapped when

$$\{j \in \{1, \ldots, N\} \mid S_j \cap \mathrm{dom}(\mu) \neq \emptyset\} \cap \{j \in \{1, \ldots, N\} \mid S_j \cap \mathrm{dom}(\mu') \neq \emptyset\} = \emptyset$$

Compatible permutations

Assume we have a partition

$$\{1, \ldots, n\} \quad = \quad S_1 \sqcup \cdots \sqcup S_N$$

Two multi-instructions μ and μ' ($\mathrm{dom}(\mu), \mathrm{dom}(\mu') \subseteq \{1, \ldots, n\}$) can be swapped when

$$\{j \in \{1, \ldots, N\} \mid S_j \cap \mathrm{dom}(\mu) \neq \emptyset\} \cap \{j \in \{1, \ldots, N\} \mid S_j \cap \mathrm{dom}(\mu') \neq \emptyset\} = \emptyset$$

A permutation π of the set $\{0, \ldots, q-1\}$ is said to be compatible with the sequence of multi-instructions $\mu_0, \ldots, \mu_{q-1}$ when it is order preserving on all pairs $\{k, k'\}$ such that μ_k and $\mu_{k'}$ cannot be swapped.

Compatible permutations

Assume we have a partition

$$\{1,\ldots,n\} \quad = \quad S_1 \sqcup \cdots \sqcup S_N$$

Two multi-instructions μ and μ' ($\mathrm{dom}(\mu), \mathrm{dom}(\mu') \subseteq \{1,\ldots,n\}$) can be swapped when

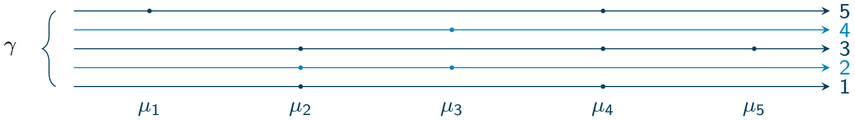
$$\{j \in \{1,\ldots,N\} \mid S_j \cap \mathrm{dom}(\mu) \neq \emptyset\} \cap \{j \in \{1,\ldots,N\} \mid S_j \cap \mathrm{dom}(\mu') \neq \emptyset\} = \emptyset$$

A permutation π of the set $\{0,\ldots,q-1\}$ is said to be compatible with the sequence of multi-instructions $\mu_0,\ldots,\mu_{q-1}$ when it is order preserving on all pairs $\{k,k'\}$ such that μ_k and $\mu_{k'}$ cannot be swapped.

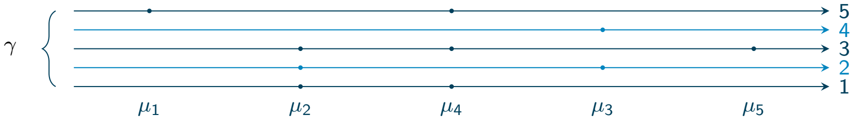
The permutation π is said to be compatible with the directed path γ when it is compatible with its associated sequence of multi-instructions.

Assume that $S_1 = \{1, 3, 5\}$ and $S_2 = \{2, 4\}$.

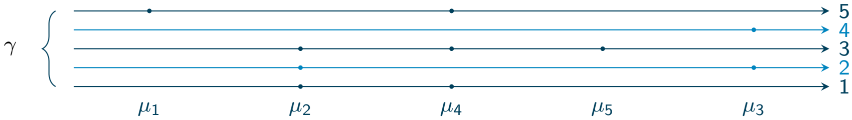
Assume that $S_1 = \{1, 3, 5\}$ and $S_2 = \{2, 4\}$.



Assume that $S_1 = \{1, 3, 5\}$ and $S_2 = \{2, 4\}$.



Assume that $S_1 = \{1, 3, 5\}$ and $S_2 = \{2, 4\}$.



Observational independence

related to partial order reduction (?)

Observational independence

related to partial order reduction (?)

Suppose that the programs $P_1, \ldots, P_N$ are compatible and that P_j has n_j running processes.

Observational independence

related to partial order reduction (?)

Suppose that the programs $P_1, \ldots, P_N$ are compatible and that P_j has n_j running processes.

The identifiers of the running processes of $P_1 | \cdots | P_N$ are the elements of $\{1, \ldots, n\}$ with

$$n = \sum_{j=1}^{N} n_j \;, \qquad \text{and for } j \in \{1, \ldots, N\} \qquad s_j = \sum_{k=1}^{j} n_k$$

Observational independence

related to partial order reduction (?)

Suppose that the programs $P_1, \dots, P_N$ are compatible and that P_j has n_j running processes.

The identifiers of the running processes of $P_1|\cdots|P_N$ are the elements of $\{1, \dots, n\}$ with

$$n \quad = \quad \sum_{j=1}^{N} n_j \ , \qquad \text{and for } j \in \{1, \dots, N\} \qquad s_j \quad = \quad \sum_{k=1}^{j} n_k$$

$$S_j \quad = \quad \{ \ i \in \{1, \dots, n\} \ \mid \ s_{j-1} \ < \ i \ \leqslant \ s_j \ \}$$

Observational independence

related to partial order reduction (?)

Suppose that the programs $P_1, \ldots, P_N$ are compatible and that P_j has n_j running processes.

The identifiers of the running processes of $P_1 | \cdots | P_N$ are the elements of $\{1, \ldots, n\}$ with

$$n = \sum_{j=1}^{N} n_j \ , \qquad \text{and for } j \in \{1, \ldots, N\} \qquad s_j = \sum_{k=1}^{j} n_k$$

$$S_j = \{\ i \in \{1, \ldots, n\}\ |\ s_{j-1} < i \leqslant s_j\ \}$$

The programs $P_1, \ldots, P_N$ are said to be observationally independent when:

- for all execution traces γ

Observational independence

related to partial order reduction (?)

Suppose that the programs $P_1, \ldots, P_N$ are compatible and that P_j has n_j running processes.

The identifiers of the running processes of $P_1|\cdots|P_N$ are the elements of $\{1, \ldots, n\}$ with

$$n \quad = \quad \sum_{j=1}^{N} n_j \;, \qquad \text{and for } j \in \{1, \ldots, N\} \qquad s_j \quad = \quad \sum_{k=1}^{j} n_k$$

$$S_j \quad = \quad \{\; i \in \{1, \ldots, n\} \;\mid\; s_{j-1} \;<\; i \;\leqslant\; s_j \;\}$$

The programs $P_1, \ldots, P_N$ are said to be observationally independent when:

- for all execution traces γ
- for all permutations π compatible with the sequence of multi-instructions $(\mu_0 \cdots \mu_{q-1})$ associated with γ,

Observational independence

related to partial order reduction (?)

Suppose that the programs $P_1, \ldots, P_N$ are compatible and that P_j has n_j running processes.

The identifiers of the running processes of $P_1|\cdots|P_N$ are the elements of $\{1, \ldots, n\}$ with

$$n \quad = \quad \sum_{j=1}^{N} n_j \ , \qquad \text{and for } j \in \{1, \ldots, N\} \qquad s_j \quad = \quad \sum_{k=1}^{j} n_k$$

$$S_j \quad = \quad \{\ i \in \{1, \ldots, n\} \ \mid \ s_{j-1} \ < \ i \ \leqslant \ s_j \ \}$$

The programs $P_1, \ldots, P_N$ are said to be observationally independent when:

- for all execution traces γ
- for all permutations π compatible with the sequence of multi-instructions $(\mu_0 \cdots \mu_{q-1})$ associated with γ,

there exists an execution trace γ' whose associated sequence of multi-instructions is $\pi \cdot (\mu_0 \cdots \mu_{q-1})$, which has the same action on the system state than γ, that is to say

$$\sigma \cdot (\mu_0 \cdots \mu_{q-1}) \quad = \quad \sigma \cdot (\mu_{\pi^{-1}(0)} \cdots \mu_{\pi^{-1}(q-1)}) \ .$$

Observational independence

related to partial order reduction (?)

Suppose that the programs $P_1, \ldots, P_N$ are compatible and that P_j has n_j running processes.

The identifiers of the running processes of $P_1 | \cdots | P_N$ are the elements of $\{1, \ldots, n\}$ with

$$n = \sum_{j=1}^{N} n_j , \qquad \text{and for } j \in \{1, \ldots, N\} \qquad s_j = \sum_{k=1}^{j} n_k$$

$$S_j = \{\ i \in \{1, \ldots, n\} \mid s_{j-1} < i \leqslant s_j\ \}$$

The programs $P_1, \ldots, P_N$ are said to be observationally independent when:

- for all execution traces γ
- for all permutations π compatible with the sequence of multi-instructions $(\mu_0 \cdots \mu_{q-1})$ associated with γ,

there exists an execution trace γ' whose associated sequence of multi-instructions is $\pi \cdot (\mu_0 \cdots \mu_{q-1})$, which has the same action on the system state than γ, that is to say

$$\sigma \cdot (\mu_0 \cdots \mu_{q-1}) = \sigma \cdot (\mu_{\pi^{-1}(0)} \cdots \mu_{\pi^{-1}(q-1)}) .$$

Observational independence cannot be decided statically,

Observational independence

related to partial order reduction (?)

Suppose that the programs $P_1, \ldots, P_N$ are compatible and that P_j has n_j running processes.

The identifiers of the running processes of $P_1 | \cdots | P_N$ are the elements of $\{1, \ldots, n\}$ with

$$n = \sum_{j=1}^{N} n_j \;, \qquad \text{and for } j \in \{1, \ldots, N\} \qquad s_j = \sum_{k=1}^{j} n_k$$

$$S_j = \{ \; i \in \{1, \ldots, n\} \mid s_{j-1} < i \leqslant s_j \; \}$$

The programs $P_1, \ldots, P_N$ are said to be observationally independent when:

- for all execution traces γ
- for all permutations π compatible with the sequence of multi-instructions $(\mu_0 \cdots \mu_{q-1})$ associated with γ,

there exists an execution trace γ' whose associated sequence of multi-instructions is $\pi \cdot (\mu_0 \cdots \mu_{q-1})$, which has the same action on the system state than γ, that is to say

$$\sigma \cdot (\mu_0 \cdots \mu_{q-1}) = \sigma \cdot (\mu_{\pi^{-1}(0)} \cdots \mu_{\pi^{-1}(q-1)}) \; .$$

Observational independence cannot be decided statically, moreover it is too loose.

Comparison

Main theorem

Main theorem

syntactic independence

$\Downarrow$

model independence

$\Downarrow$

observational independence