First PROMETIDOS-CM Summer School

Semantics of Concurrent Processes: Unification and new Directions

David de Frutos Escrig

In collaboration with
Carlos Gregorio Rodríguez, Miguel Palomino, Ignacio Fábregas,
David Romero, Luca Aceto, Anna Ingólfsdóttir

Departamento de Sistemas Informáticos y Programación
Universidad Complutense de Madrid
Madrid, Septiembre 2011

Outline

1. Introduction and Motivation
2. Classic Semantics of Processes
3. Unification of the Observational Semantics
4. Unification of the Equational Semantics
5. Unification of the Logical Semantics
6. Covariant-contravariant Semantics
7. Several New Semantics
   - Possible Worlds Logic
   - Meet and Join Semantics
   - Partial Offers Semantics

Outline

1. Introduction and Motivation
2. Classic Semantics of Processes
3. Unification of the Observational Semantics
4. Unification of the Equational Semantics
5. Unification of the Logical Semantics
6. Covariant-contravariant Semantics
7. Several New Semantics
   - Possible Worlds Logic
   - Meet and Join Semantics
   - Partial Offers Semantics

Subject of study

"agents that act and interact continuously with other similar agents and with their common environment. The agents may be real-world objects (even people), or they may be artefacts, embodied perhaps in computer hardware or software systems"

C.A.R. Hoare

Subject of study

“agents that act and interact continuously with other similar agents and with their common environment. The agents may be real-world objects (even people), or they may be artefacts, embodied perhaps in computer hardware or software systems”

C.A.R. Hoare

Subject of study

"agents that act and interact continuously with other similar agents and with their common environment. The agents may be real-world objects (even people), or they may be artefacts, embodied perhaps in computer hardware or software systems"

C.A.R. Hoare

Subject of study

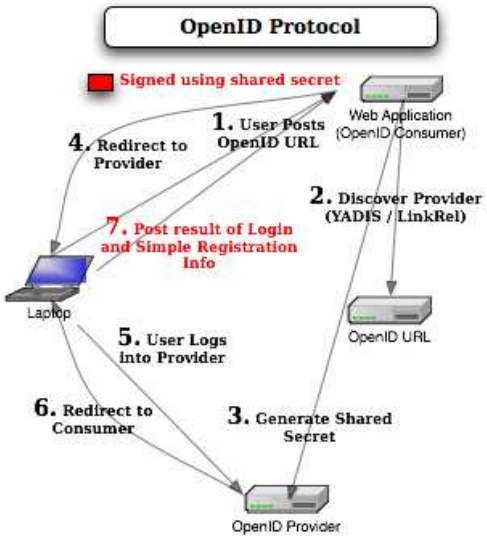
"agents that act and interact continuously with other similar agents and with their common environment. The agents may be real-world objects (even people), or they may be artefacts, embodied perhaps in computer hardware or software systems"

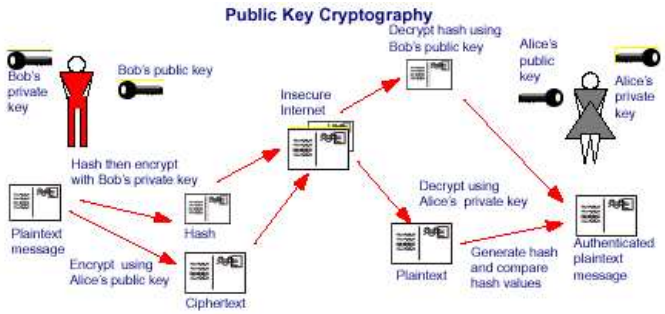
C.A.R. Hoare

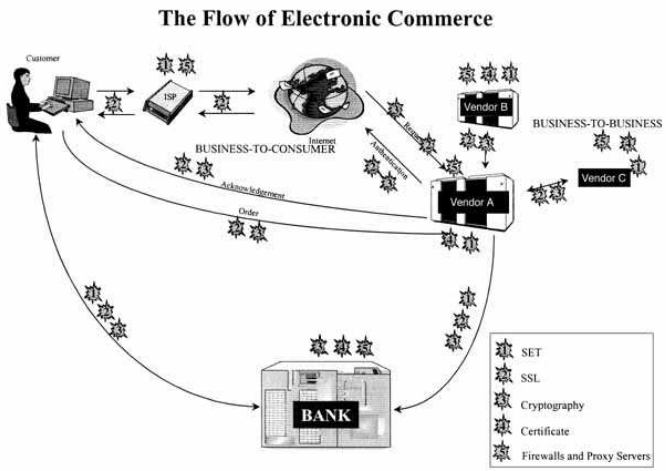
Subject of study

"agents that act and interact continuously with other similar agents and with their common environment. The agents may be real-world objects (even people), or they may be artefacts, embodied perhaps in computer hardware or software systems"

C.A.R. Hoare







Sequential Programs vs. Concurrent Processes

Sequential Programs

input —- $P \longrightarrow$ *output*

{*precondition*} P {*postcondition*}

Concurrent Processes

Possible infinite computations.

Open, reactive systems.

Assynchronus computations: non-determinism.

Subject of study

Processes

Mathematical abstraction of agent behaviour

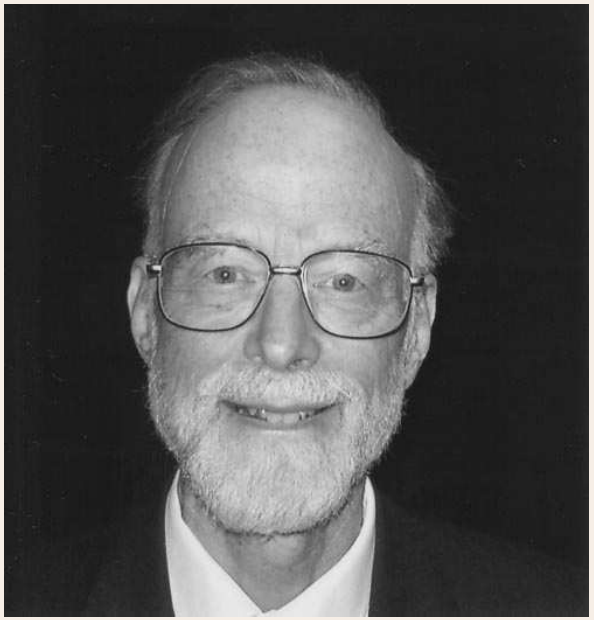
Subject of study

Interleaving Semantics

Processes

Sequences of executed actions

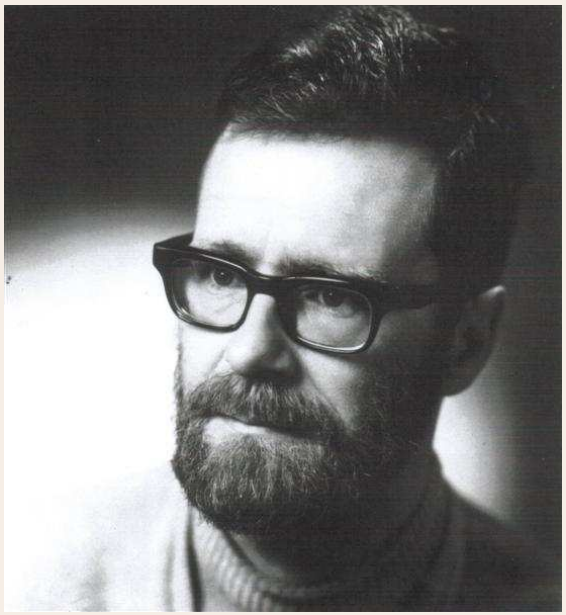
Hoare



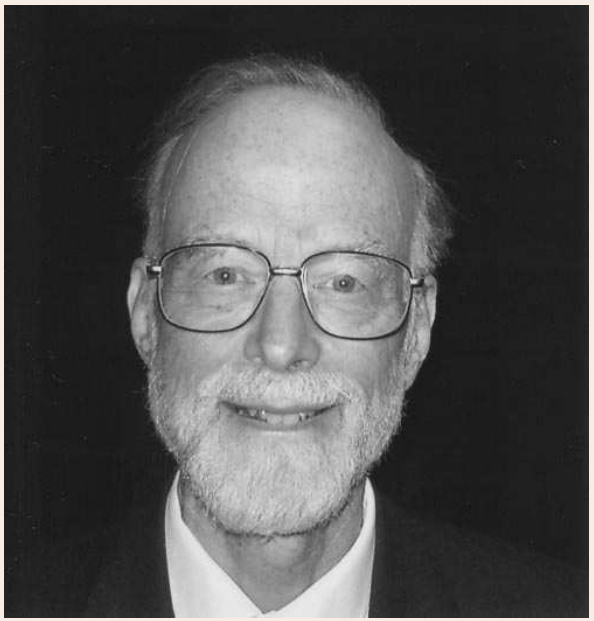
Subject of study

Processes

Dijkstra



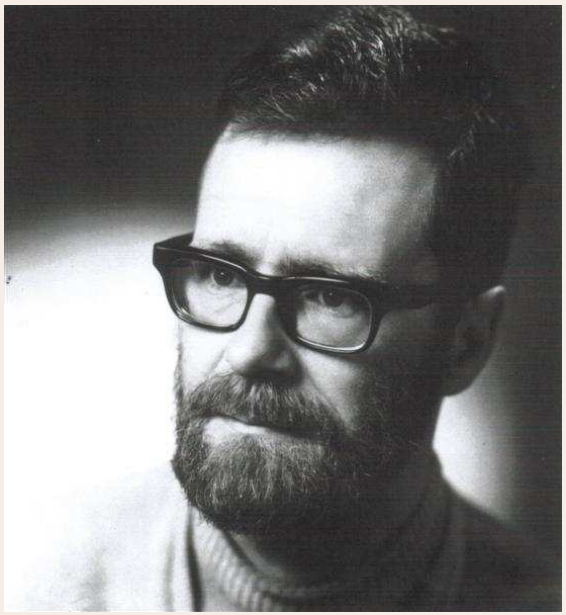
Hoare



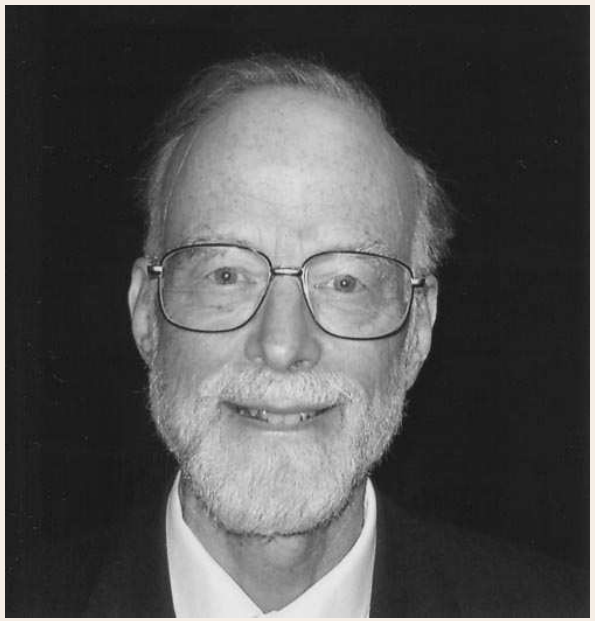
Subject of study

Processes

Dijkstra



Hoare

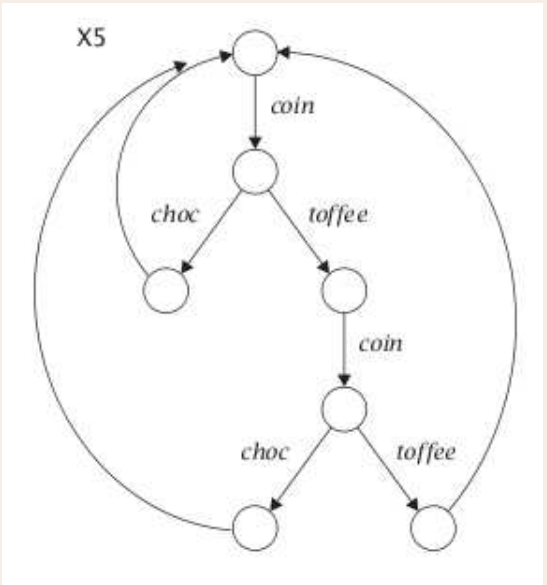


Milner

Subject of study

Processes

Operational



Subject of study

Processes

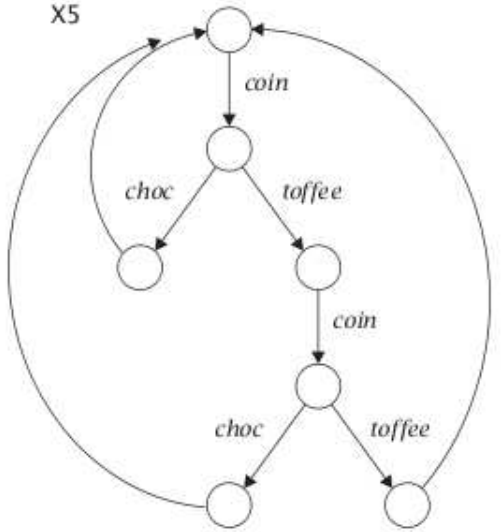
Denotational

$$failures(P \parallel Q) =$$
$$\{ s, (X \cup Y) \mid s \in (\alpha P \cup \alpha Q)^* \wedge (s \upharpoonright \alpha P, X) \in failures(P) \wedge (s \upharpoonright \alpha Q, Y) \in failures(Q) \} \cup$$
$$\{ s, X \mid s \in divergences(P \parallel Q) \}$$

$$failures(f(P)) = \{ f^*(s), f(X) \mid (s, X) \in failures(P) \}$$

$$failures(P \,\Box\, Q) =$$
$$\{ s, X \mid (s, X) \in failures(P) \cap failures(Q) \vee (s \neq \langle\rangle \wedge (s, X) \in failures(P) \cup failures(Q)) \} \cup$$
$$\{ s, X \mid s \in divergences(P \,\Box\, Q) \}$$

Operational



Subject of study

Processes

Algebraic

$$(x + y) : z = x : z + y : z$$
$$(x : y) : z = x : (y : z)$$
$$(x : y) \cdot z = x : (y \cdot z)$$
$$(x \cdot y) : z = x \cdot (y : z)$$
$$\delta : x = \delta$$

$$(a : x) \between y = a : (x \between y + x \mid y)$$
$$(a : x) \mid (b : y) = (a \mid b) : (x \mid y)$$
$$(a : x) \mid (by) = (a \mid b) : (x \between y + x \mid y)$$
$$(a : x) \mid b = (a \mid b) : x$$

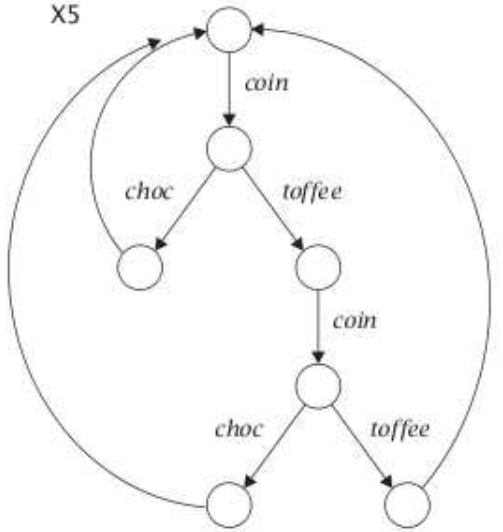
Denotational

$$\begin{aligned}
&failures(P \parallel Q) = \\
&\quad \{ s, (X \cup Y) \mid s \in (\alpha P \cup \alpha Q)^* \wedge (s \upharpoonright \alpha P, X) \in failures(P) \wedge \\
&\qquad\qquad (s \upharpoonright \alpha Q, Y) \in failures(Q) \} \cup \\
&\quad \{ s, X \mid s \in divergences(P \parallel Q) \}
\end{aligned}$$

$$failures(f(P)) = \{ f^*(s), f(X) \mid (s, X) \in failures(P) \}$$

$$\begin{aligned}
&failures(P \Box Q) = \\
&\quad \{ s, X \mid (s, X) \in failures(P) \cap failures(Q) \vee \\
&\qquad\qquad (s \neq \langle\rangle \wedge (s, X) \in failures(P) \cup failures(Q)) \} \cup \\
&\quad \{ s, X \mid s \in divergences(P \Box Q) \}
\end{aligned}$$

Operational



Subject of study

$$p \stackrel{?}{=} q$$

Processes

Algebraic

$$(x + y) : z = x : z + y : z$$
$$(x : y) : z = x : (y : z)$$
$$(x : y) \cdot z = x : (y \cdot z)$$
$$(x \cdot y) : z = x \cdot (y : z)$$
$$\delta : x = \delta$$

$$(a : x) \between y = a : (x \between y + x \mid y)$$
$$(a : x) \mid (b : y) = (a \mid b) : (x \mid y)$$
$$(a : x) \mid (by) = (a \mid b) : (x \between y + x \mid y)$$
$$(a : x) \mid b = (a \mid b) : x$$

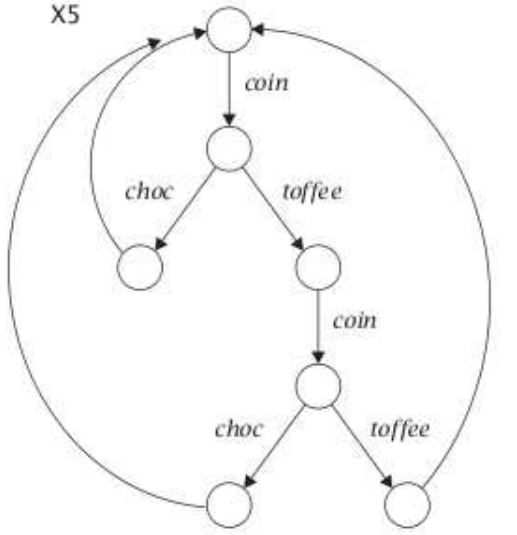
Denotational

$$failures(P \parallel Q) =$$
$$\{ s, (X \cup Y) \mid s \in (\alpha P \cup \alpha Q)^* \wedge (s \upharpoonright \alpha P, X) \in failures(P) \wedge (s \upharpoonright \alpha Q, Y) \in failures(Q) \} \cup$$
$$\{ s, X \mid s \in divergences(P \parallel Q) \}$$

$$failures(f(P)) = \{ f^*(s), f(X) \mid (s, X) \in failures(P) \}$$

$$failures(P \Box Q) =$$
$$\{ s, X \mid (s, X) \in failures(P) \cap failures(Q) \vee (s \neq \langle\rangle \wedge (s, X) \in failures(P) \cup failures(Q)) \} \cup$$
$$\{ s, X \mid s \in divergences(P \Box Q) \}$$

Operational



Subject of study

$$p \stackrel{?}{=} q \qquad p \stackrel{?}{\leq} q$$

Processes

Algebraic

$$(x + y) : z = x : z + y : z$$
$$(x : y) : z = x : (y : z)$$
$$(x : y) \cdot z = x : (y \cdot z)$$
$$(x \cdot y) : z = x \cdot (y : z)$$
$$\delta : x = \delta$$

$$(a : x) \between y = a : (x \between y + x \mid y)$$
$$(a : x) \mid (b : y) = (a \mid b) : (x \mid y)$$
$$(a : x) \mid (by) = (a \mid b) : (x \between y + x \mid y)$$
$$(a : x) \mid b = (a \mid b) : x$$

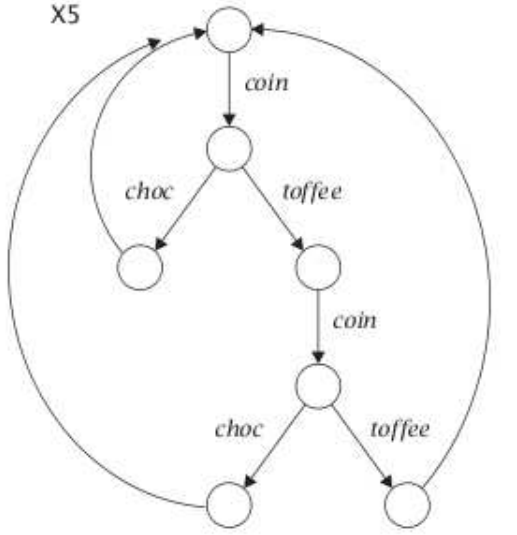
Denotational

$$\begin{aligned}
&failures(P \parallel Q) = \\
&\quad \{ s, (X \cup Y) \mid s \in (\alpha P \cup \alpha Q)^* \wedge (s \upharpoonright \alpha P, X) \in failures(P) \wedge \\
&\qquad\qquad (s \upharpoonright \alpha Q, Y) \in failures(Q) \} \cup \\
&\quad \{ s, X \mid s \in divergences(P \parallel Q) \}
\end{aligned}$$

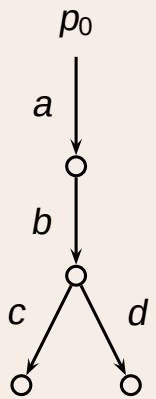
$$failures(f(P)) = \{ f^*(s), f(X) \mid (s, X) \in failures(P) \}$$

$$\begin{aligned}
&failures(P \,\Box\, Q) = \\
&\quad \{ s, X \mid (s, X) \in failures(P) \cap failures(Q) \vee \\
&\qquad\qquad (s \neq \langle\rangle \wedge (s, X) \in failures(P) \cup failures(Q)) \} \cup \\
&\quad \{ s, X \mid s \in divergences(P \,\Box\, Q) \}
\end{aligned}$$

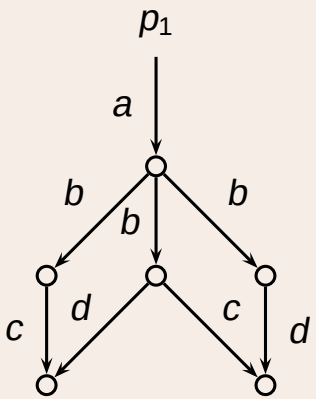
Operational



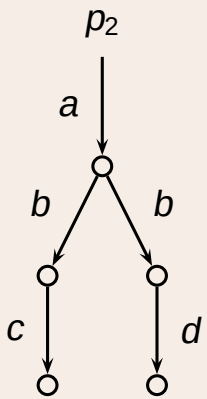
Some Simple Processes



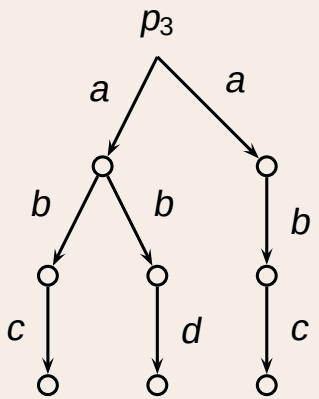
$ab(c + d)$



$a(bc + bd + b(d + c))$



$a(bc + bd)$



$a(bc + bd) + abc$

Trace Semantics

$$\textbf{traces}(p) = \{\sigma \mid \exists\, p' \quad p \xrightarrow{\sigma} p', \sigma = a_1 \ldots a_n\}$$

$$p =_T q \Leftrightarrow \textbf{traces}(p) = \textbf{traces}(q)$$

Bisimulation Semantics

A relation S is a bisimulation if

- $p \xrightarrow{a} p'$ then $\exists q'$ such that $q \xrightarrow{a} q'$ and $p' S q'$
- $q \xrightarrow{a} q'$ then $\exists p'$ such that $p \xrightarrow{a} p'$ and $q' S p'$

We write $p =_B q$ if there is a bisimulation S such that pSq

BCCSP processes

Definition

The set BCCSP(*Act*) of processes is defined by:

$$p ::= \mathbf{0} \mid ap \mid p + q$$

where $a \in Act$; $\mathbf{0}$ represents the process that performs no action; for every action in *Act*, there is a prefix operator; and + is a choice operator.

Definition

The operational semantics for BCCSP terms is defined by

$$ap \xrightarrow{a} p \qquad \frac{p \xrightarrow{a} p'}{p + q \xrightarrow{a} p'} \qquad \frac{q \xrightarrow{a} q'}{p + q \xrightarrow{a} q'}$$

BCCSP processes

Definition

The set BCCSP(*Act*) of processes is defined by:

$$p ::= \mathbf{0} \mid ap \mid p + q$$

where $a \in Act$; $\mathbf{0}$ represents the process that performs no action; for every action in *Act*, there is a prefix operator; and + is a choice operator.

Definition

The operational semantics for BCCSP terms is defined by

$$ap \xrightarrow{a} p \qquad \frac{p \xrightarrow{a} p'}{p + q \xrightarrow{a} p'} \qquad \frac{q \xrightarrow{a} q'}{p + q \xrightarrow{a} q'}$$

Trace Semantics: Denotational Definition

$$\textbf{traces}(p) = \{\sigma \mid \exists\, p' \quad p \xrightarrow{\sigma} p', \sigma = a_1 \ldots a_n\}$$

$$p =_T q \Leftrightarrow \textbf{traces}(p) = \textbf{traces}(q)$$

$$\begin{aligned}
\textbf{traces}(\mathbf{0}) &= \{\langle\rangle\} \\
\textbf{traces}(ap) &= \{\langle\rangle\} \cup \{\langle a\rangle^\frown t \mid t \in \textbf{traces}(p)\} \\
\textbf{traces}(p+q) &= \textbf{traces}(p) \cup \textbf{traces}(q)
\end{aligned}$$

$$p =_T q \Leftrightarrow \textbf{traces}(p) = \textbf{traces}(q)$$

Simulations

(Plain) Simulation

Relation S is a *simulation* if whenever pSq then

- If $p \xrightarrow{a} p'$ then $\exists q'$ such that $q \xrightarrow{a} q'$ and $p' S q'$

We write $p \sqsubseteq_S q$ if there is a simulation S such that pSq

Ready simulation

Relation S is a *ready simulation* if whenever pSq, then

- If $p \xrightarrow{a} p'$ then $\exists q'$ such that $q \xrightarrow{a} q'$ and $p' S q'$
- $p \xrightarrow{a} \iff q \xrightarrow{a} \quad (I(p) = I(q))$

We write $p \sqsubseteq_{RS} q$ if there is a ready simulation S such that pSq

Constrained Simulations

Ready simulation

Relation S is a *ready simulation* if whenever pSq, then

- If $p \xrightarrow{a} p'$ then $\exists q'$ such that $q \xrightarrow{a} q'$ and $p' S q'$
- $p \xrightarrow{a} \iff q \xrightarrow{a} \quad (I(p) = I(q))$

We write $p \sqsubseteq_{RS} q$ if there is a ready simulation S such that pSq

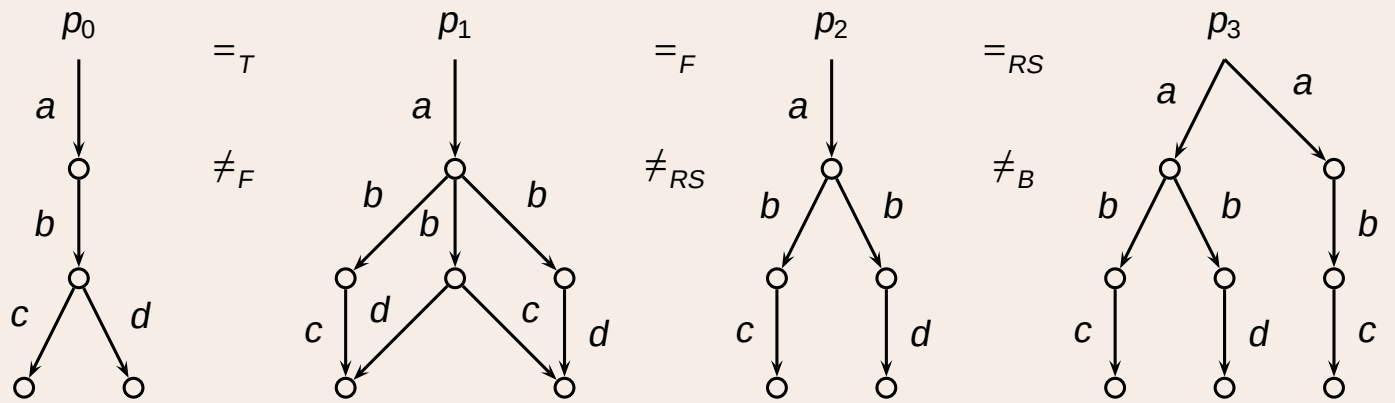
C-constrained Simulation

Relation S is a *C-constrained simulation* if whenever pSq, then

- If $p \xrightarrow{a} p'$ then $\exists q'$ such that $q \xrightarrow{a} q'$ and $p' S q'$
- pCq

We write $p \sqsubseteq_{CS} q$ if there is a C-constrained simulation S such that pSq

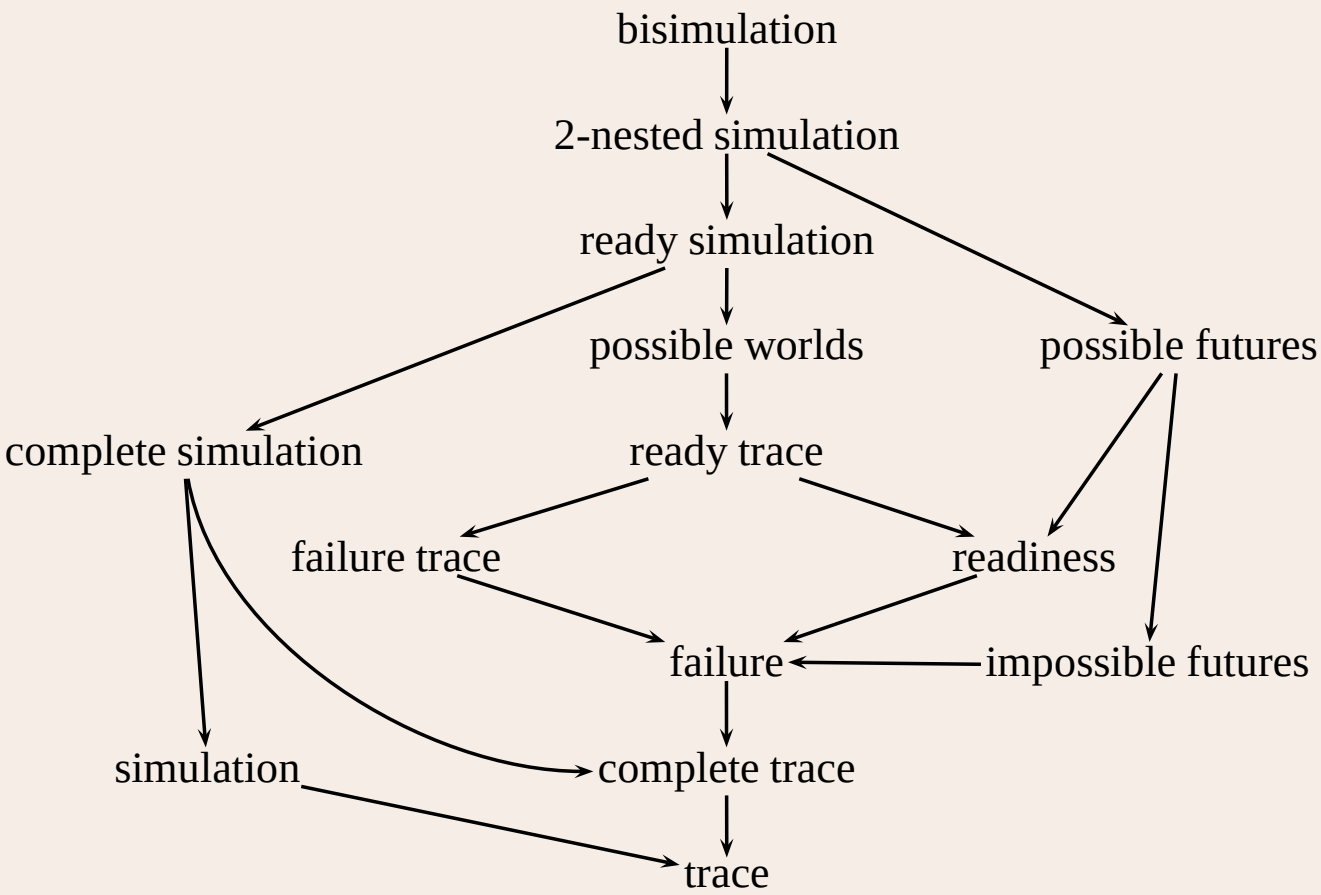
Comparing the Semantics



$ab(c + d)$ $a(bc + bd + b(d + c))$ $a(bc + bd)$ $a(bc + bd) + abc$

$=_T$ Traces	$=_F$ Failures	$=_{RS}$ Ready Similarity	$=_B$ Strong Bisimulation

Linear Time-Branching Time spectrum



BCCSP axiomatisations for the equivalences

	B	RS	PW	RT	FT	R	F	CS	CT	S	T
$(x + y) + z = x + (y + z)$	+	+	+	+	+	+	+	+	+	+	+
$x + y = y + x$	+	+	+	+	+	+	+	+	+	+	+
$x + 0 = x$	+	+	+	+	+	+	+	+	+	+	+
$x + x = x$	+	+	+	+	+	+	+	+	+	+	+
$I(x) = I(y) \Rightarrow a(x + y) = a(x + y) + ay$		+	v	v	v	v	v	v	v	v	v
$a(bx + by + z) = a(bx + z) + a(by + z)$			+	v	v	v	v		v		v
$I(x) = I(y) \Rightarrow ax + ay = a(x + y)$				+	+	v	v		v		v
$ax + ay = ax + ay + a(x + y)$					+		v		v		v
$a(bx + u) + a(by + v) = a(bx + by + u) + a(by + v)$						+	+		v		v
$ax + a(y + z) = ax + a(x + y) + a(y + z)$							+		v		v
$a(x + by + z) = a(x + by + z) + a(by + z)$								+	v	v	v
$a(bx + u) + a(cy + v) = a(bx + cy + u + v)$									+		v
$a(x + y) = a(x + y) + ay$										+	v
$ax + ay = a(x + y)$											+

BCCSP axiomatisations for the preorders

	B	RS	PW	RT	FT	R	F	CS	CT	S	T
$(x + y) + z = x + (y + z)$	+	+	+	+	+	+	+	+	+	+	+
$x + y = y + x$	+	+	+	+	+	+	+	+	+	+	+
$x + 0 = x$	+	+	+	+	+	+	+	+	+	+	+
$x + x = x$	+	+	+	+	+	+	+	+	+	+	+
$ax \sqsubseteq ax + ay$		+	+	+	+	+	+	v	v	v	v
$a(bx + by + z) = a(bx + z) + a(by + z)$			+	v	v	v	v		v		v
$I(x) = I(y) \Rightarrow ax + ay = a(x + y)$				+	v	v	v		v		v
$ax + ay \sqsupseteq a(x + y)$					+		v		v		v
$a(bx + u) + a(by + v) \sqsupseteq a(bx + by + u)$						+	v		v		v
$ax + a(y + z) \sqsupseteq a(x + y)$							+		v		v
$ax \sqsubseteq ax + y$								+	+	v	v
$a(bx + u) + a(cy + v) = a(bx + cy + u + v)$									+		v
$x \sqsubseteq x + y$										+	+
$ax + ay = a(x + y)$											+

BCCSP axiomatisations for the preorders

	B	RS	PW	RT	FT	R	F	CS	CT	S	T
$(x + y) + z = x + (y + z)$	+	+	+	+	+	+	+	+	+	+	+
$x + y = y + x$	+	+	+	+	+	+	+	+	+	+	+
$x + 0 = x$	+	+	+	+	+	+	+	+	+	+	+
$x + x = x$	+	+	+	+	+	+	+	+	+	+	+
$ax \sqsubseteq ax + ay$		+	+	+	+	+	+	v	v	v	v
$a(bx + by + z) = a(bx + z) + a(by + z)$			+	v	v	v	v		v		v
$I(x) = I(y) \Rightarrow ax + ay = a(x + y)$				+	v	v	v		v		v
$ax + ay \sqsupseteq a(x + y)$					+		v		v		v
$a(bx + u) + a(by + v) \sqsupseteq a(bx + by + u)$						+	v		v		v
$ax + a(y + z) \sqsupseteq a(x + y)$							+		v		v
$ax \sqsubseteq ax + y$								+	+	v	v
$a(bx + u) + a(cy + v) = a(bx + cy + u + v)$									+		v
$x \sqsubseteq x + y$										+	+
$ax + ay = a(x + y)$											+

Some questions...

- Is there any relation between the axioms of preorders and equivalences? it should be!
- Cannot we derive both of them together?

Observational framework

Observations

- Branching observations for simulation semantics
- Linear observations for extensional semantics (degenerated branching)
- Deterministic branching observations for rare semantics

Branching observations

- A Local observation function L_N observing the states.
- Branching general observation (bgo) a partial aggregated view of a collection of computations of a process observing L_N at their states

Observational framework

Observations

- Branching observations for simulation semantics
- Linear observations for extensional semantics (degenerated branching)
- Deterministic branching observations for rare semantics

Branching observations

- A Local observation function L_N observing the states.
- Branching general observation (bgo) a partial aggregated view of a collection of computations of a process observing L_N at their states

Observational framework

$BGO_N(p)$

$$\{\langle L_N(p), S\rangle \mid S \subseteq \{(a, bgo) \mid bgo \in BGO_N(p'), p \xrightarrow{a} p'\}\}$$

Example

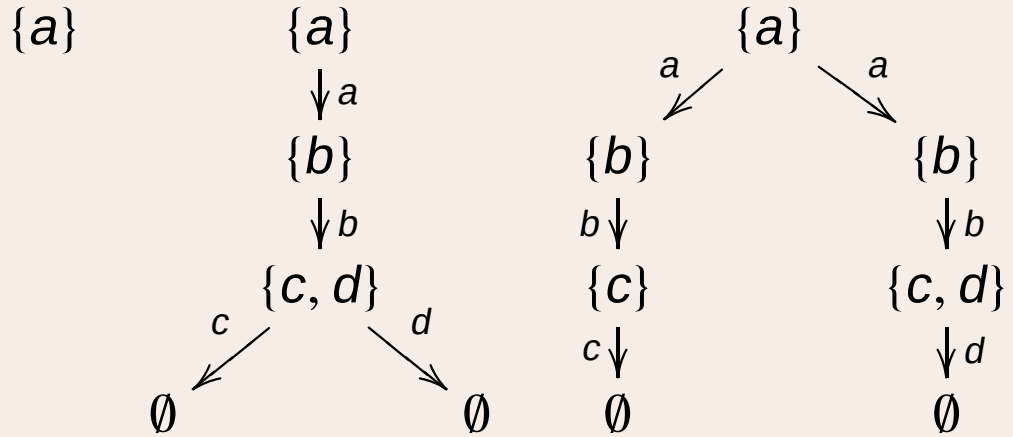
Some bgo's in $BGO_I(p)$ for $p = a(b(c + d) + bc + bd)$

Observational framework

$BGO_N(p)$

$$\{\langle L_N(p), S\rangle \mid S \subseteq \{(a, bgo) \mid bgo \in BGO_N(p'), p \xrightarrow{a} p'\}\}$$

Example



Some bgo's in $BGO_I(p)$ for $p = a(b(c + d) + bc + bd)$

Observational framework

Characterisation

$$\begin{array}{ccc} p & \sqsubseteq & q \\ \downarrow & & \downarrow \\ BGO(p) & \leq & BGO(q) \end{array}$$

C-Simulations

$$\begin{array}{ccc} p & \overset{\sqsubseteq^{C}}{\rightarrow} & q \\ \downarrow & & \downarrow \\ BGO_{C}(p) & \subseteq & BGO_{C}(q) \end{array}$$

Observational framework

Characterisation

$$\begin{array}{ccc} p & \sqsubseteq & q \\ \Downarrow & & \Downarrow \\ BGO(p) & \leq & BGO(q) \end{array}$$

C-Simulations

$$\begin{array}{ccc} p & \overset{\sqsubseteq^{C}}{\rightarrow} & q \\ \Downarrow & & \Downarrow \\ BGO_{C}(p) & \subseteq & BGO_{C}(q) \end{array}$$

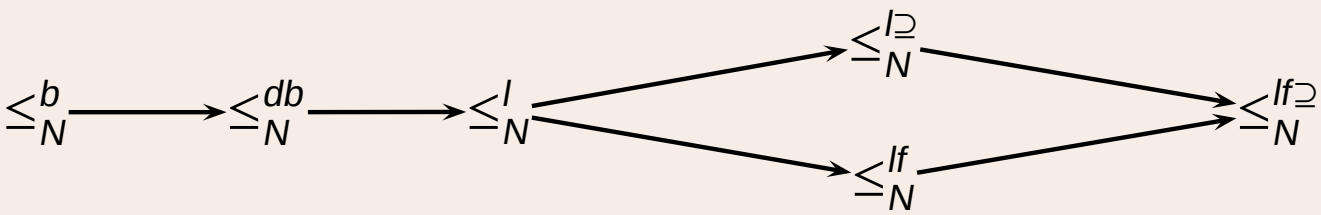
Observational framework

Characterisation

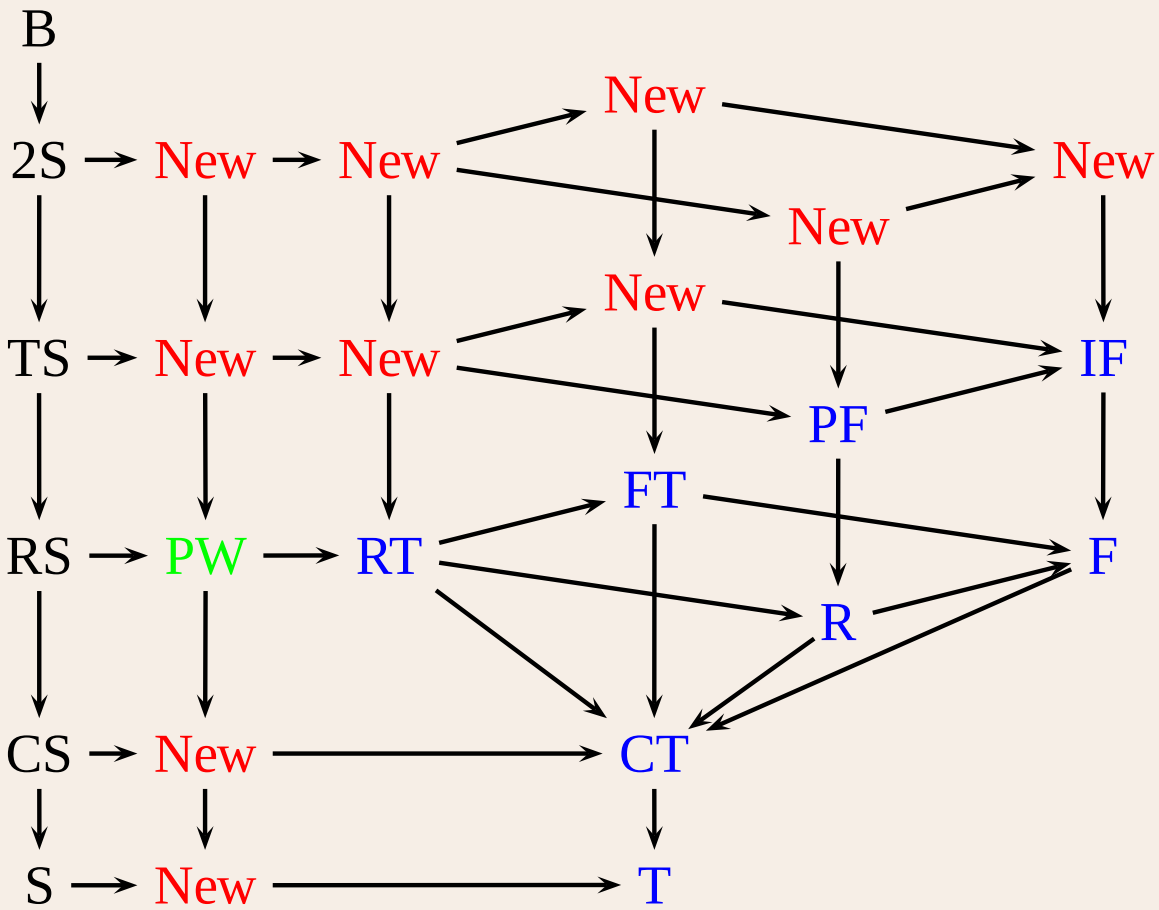
$$\begin{array}{ccc} p & \sqsubseteq & q \\ \Downarrow & & \Downarrow \\ BGO(p) & \leq & BGO(q) \end{array}$$

C-Simulations

$$\begin{array}{ccc} p & \overset{\sqsubseteq^{C}}{\rightarrow} & q \\ \Downarrow & & \Downarrow \\ BGO_{C}(p) & \subseteq & BGO_{C}(q) \end{array}$$



The new picture of the spectrum



Uniform axiomatisation

Axioms for the semantic preorders

	B	RS	PW	RT	FT	R	F	CS	CT	S	T
$(x + y) + z = x + (y + z)$	+	+	+	+	+	+	+	+	+	+	+
$x + y = y + x$	+	+	+	+	+	+	+	+	+	+	+
$x + 0 = x$	+	+	+	+	+	+	+	+	+	+	+
$x + x = x$	+	+	+	+	+	+	+	+	+	+	+
$ax \sqsubseteq ax + ay$		+	+	+	+	+	+	v	v	v	v
$a(bx + by + z) = a(bx + z) + a(by + z)$			+	v	v	v	v		v		v
$I(x) = I(y) \Rightarrow ax + ay = a(x + y)$				+	v	v	v		v		v
$ax + ay \sqsupseteq a(x + y)$					+		v		v		v
$a(bx + u) + a(by + v) \sqsupseteq a(bx + by + u)$						+	v		v		v
$ax + a(y + z) \sqsupseteq a(x + y)$							+		v		v
$ax \sqsubseteq ax + y$								+	+	v	v
$a(bx + u) + a(cy + v) = a(bx + cy + u + v)$									+		v
$x \sqsubseteq x + y$										+	+
$ax + ay = a(x + y)$											+

Uniform axiomatisation

Generic axioms

$$C(x, y) \Rightarrow x \sqsubseteq x + y$$
$$M(x, y, w) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

(and the bisimulation axioms)

Uniform axiomatisation

Generic axioms

$$C(x, y) \Rightarrow x \sqsubseteq x + y$$
$$M(x, y, w) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

Constraint I

$$I(x) = I(y) \Rightarrow x \sqsubseteq x + y$$

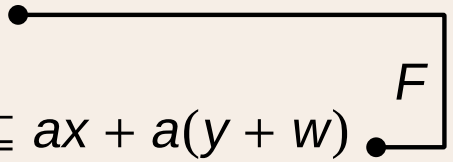
Uniform axiomatisation

Generic axioms

$$C(x, y) \Rightarrow x \sqsubseteq x + y$$
$$M(x, y, w) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

Constraint I

$$I(x) = I(y) \Rightarrow x \sqsubseteq x + y$$
$$(none) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$



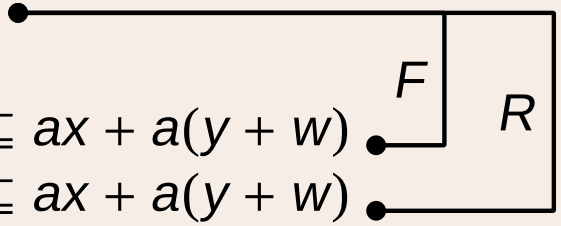
Uniform axiomatisation

Generic axioms

$$C(x, y) \Rightarrow x \sqsubseteq x + y$$
$$M(x, y, w) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

Constraint I

$$I(x) = I(y) \Rightarrow x \sqsubseteq x + y$$



$$(none) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$
$$I(x) \supseteq I(y) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

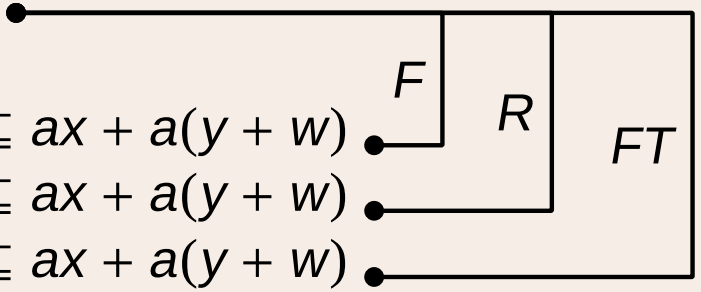
Uniform axiomatisation

Generic axioms

$$C(x, y) \Rightarrow x \sqsubseteq x + y$$
$$M(x, y, w) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

Constraint I

$$I(x) = I(y) \Rightarrow x \sqsubseteq x + y$$



$$(none) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$
$$I(x) \supseteq I(y) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$
$$I(w) \subseteq I(y) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

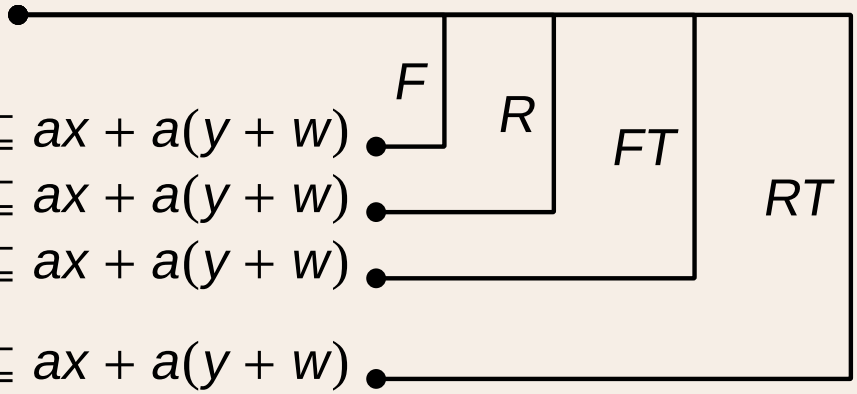
Uniform axiomatisation

Generic axioms

$$C(x, y) \Rightarrow x \sqsubseteq x + y$$
$$M(x, y, w) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

Constraint I

$$I(x) = I(y) \Rightarrow x \sqsubseteq x + y$$



$$(none) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$
$$I(x) \supseteq I(y) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$
$$I(w) \subseteq I(y) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$
$$\left.\begin{array}{l} I(x) = I(y) \\ I(w) \subseteq I(y) \end{array}\right\} \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

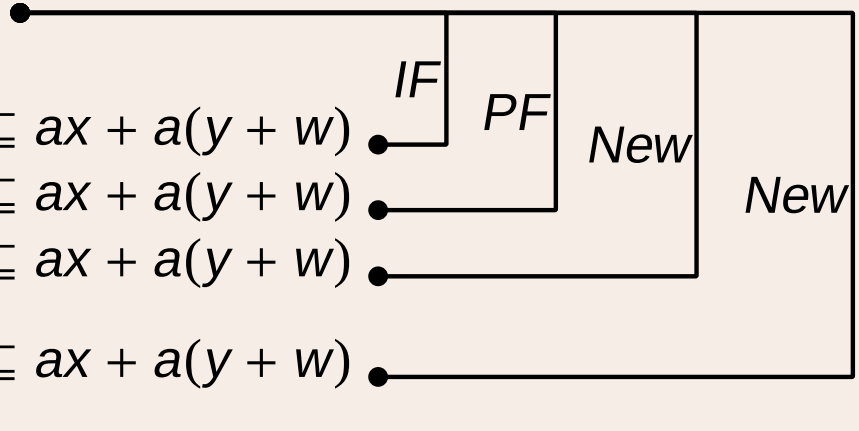
Uniform axiomatisation

Generic axioms

$$C(x,y) \Rightarrow x \sqsubseteq x+y$$
$$M(x,y,w) \Rightarrow a(x+y) \sqsubseteq ax + a(y+w)$$

Constraint T

$$T(x) = T(y) \Rightarrow x \sqsubseteq x+y$$



$$(none) \Rightarrow a(x+y) \sqsubseteq ax + a(y+w)$$
$$T(x) \supseteq T(y) \Rightarrow a(x+y) \sqsubseteq ax + a(y+w)$$
$$T(w) \subseteq T(y) \Rightarrow a(x+y) \sqsubseteq ax + a(y+w)$$
$$\left.\begin{array}{l} T(x) = T(y) \\ T(w) \subseteq T(y) \end{array}\right\} \Rightarrow a(x+y) \sqsubseteq ax + a(y+w)$$

Uniform axiomatisation

Generic axioms

$$C(x, y) \Rightarrow x \sqsubseteq x + y$$
$$M(x, y, w) \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

Uniform axiomatisation

Generic axioms

$$C(x, y) \Rightarrow x \sqsubseteq x + y$$
$$M(x, y, w) \Rightarrow ax + a(y + w) + a(x + y) = ax + a(y + w)$$

Hennessy-Milner logic

DEFINITION (Hennessy-Milner logic, HML)

The set $\mathcal{L}_{HM}$ of Hennessy-Milner logical formulas is defined by:

- If $\varphi_i \in \mathcal{L}_{HM}\ \forall i \in I$ then $\bigwedge_{i \in I} \varphi_i \in \mathcal{L}_{HM}$.
- If $a \in Act$ and $\varphi \in \mathcal{L}_{HM}$ then $a\varphi \in \mathcal{L}_{HM}$.
- If $\varphi \in \mathcal{L}_{HM}$ then $\neg\varphi \in \mathcal{L}_{HM}$.

DEFINITION (Satisfaction relation)

For each lts $\mathbb{P}$, the satisfaction relation $\models \subseteq \mathbb{P} \times \mathcal{L}_{HM}$ is defined by:

- $p \models a\varphi$ if there exists $q \in \mathbb{P} : p \xrightarrow{a} q$ and $q \models \varphi$;
- $p \models \bigwedge_{i \in I} \varphi_i$ if for all $i \in I : p \models \varphi_i$.
- $p \models \neg\varphi$ if $p \not\models \varphi$.

Hennessy-Milner logic

DEFINITION (Hennessy-Milner logic, HML)

The set $\mathcal{L}_{HM}$ of Hennessy-Milner logical formulas is defined by:

- If $\varphi_i \in \mathcal{L}_{HM}\ \forall i \in I$ then $\bigwedge_{i \in I} \varphi_i \in \mathcal{L}_{HM}$.
- If $a \in Act$ and $\varphi \in \mathcal{L}_{HM}$ then $a\varphi \in \mathcal{L}_{HM}$.
- If $\varphi \in \mathcal{L}_{HM}$ then $\neg\varphi \in \mathcal{L}_{HM}$.

DEFINITION (Satisfaction relation)

For each lts $\mathbb{P}$, the satisfaction relation $\models \subseteq \mathbb{P} \times \mathcal{L}_{HM}$ is defined by:

- $p \models a\varphi$ if there exists $q \in \mathbb{P} : p \xrightarrow{a} q$ and $q \models \varphi$;
- $p \models \bigwedge_{i \in I} \varphi_i$ if for all $i \in I : p \models \varphi_i$.
- $p \models \neg\varphi$ if $p \not\models \varphi$.

Semantics Defined by a Logic

Definition

Any subset $\mathcal{L}$ of $\mathcal{L}_{HM}$ induces a logical semantics for processes, given by the preorder $\sqsubseteq_{\mathcal{L}}$:

$$p \sqsubseteq_{\mathcal{L}} q \text{ iff } \forall \varphi \in \mathcal{L}, (p \models \varphi \Rightarrow q \models \varphi).$$

We say that $\mathcal{L}$ and $\mathcal{L}'$ are equivalent, and we write $\mathcal{L} \sim \mathcal{L}'$, if they induce the same semantics, that is $\sqsubseteq_{\mathcal{L}} = \sqsubseteq_{\mathcal{L}'}$.

Van Glabbeek's logical characterizations

Formulas \ Semantics ($\mathcal{Z}$)	T	S	CT	CS	F	FT	R	RT	PW	RS	PF	2S	B
$\top \in \mathcal{L}_{\mathcal{Z}}$	•	ν	•	ν	•	•	•	•	ν	ν	ν	ν	ν
$\mathbf{0} \in \mathcal{L}_{\mathcal{Z}}$			•	•	ν	ν	ν	ν	ν	ν	ν	ν	ν
$\varphi \in \mathcal{L}_{\mathcal{Z}},\ a \in Act \Rightarrow$ $a\varphi \in \mathcal{L}_{\mathcal{Z}}$	•	•	•	•	•	•	•	•	ν	•	•	•	•
$X \subseteq Act \Rightarrow$ $\widetilde{X} \in \mathcal{L}_{\mathcal{Z}}$					•	ν	ν	ν	ν	ν	ν	ν	ν
$X \subseteq Act \Rightarrow$ $X \in \mathcal{L}_{\mathcal{Z}}$							•	ν	•	•	ν	ν	ν
$\varphi \in \mathcal{L}_{\mathcal{Z}},\ X \subseteq Act \Rightarrow$ $\widetilde{X}\varphi \in \mathcal{L}_{\mathcal{Z}}$						•		ν	ν	ν		ν	ν
$\varphi \in \mathcal{L}_{\mathcal{Z}},\ X \subseteq Act \Rightarrow$ $X\varphi \in \mathcal{L}_{\mathcal{Z}}$								•	ν	ν		ν	ν
$\varphi_i \in \mathcal{L}_{\mathcal{Z}}\ \forall i \in I \Rightarrow$ $\bigwedge_{i\in I} \varphi_i \in \mathcal{L}_{\mathcal{Z}}$		•		•						•		•	•
$X \subseteq Act,\ \varphi_a \in \mathcal{L}_{PW}\ \forall a \in X \Rightarrow$ $\bigwedge_{a\in X} a\varphi_a \in \mathcal{L}_{\mathcal{Z}}$									•	ν		ν	ν
$\varphi_i, \varphi_j \in \mathcal{L}_T\ \forall i \in I \forall j \in J \Rightarrow$ $\bigwedge_{i\in I} \varphi_i \wedge \bigwedge_{j\in J} \neg\varphi_j \in \mathcal{L}_{\mathcal{Z}}$											•	ν	ν
$\varphi \in \mathcal{L}_S \Rightarrow$ $\neg\varphi \in \mathcal{L}_{\mathcal{Z}}$												•	ν
$\varphi \in \mathcal{L}_{\mathcal{Z}} \Rightarrow$ $\neg\varphi \in \mathcal{L}_{\mathcal{Z}}$													•

Unified Definition of the Logics

- In order to obtain a uniform presentation of the logics characterizing each of the semantics we look for (simple) set of formulas as large as possible.
- The key point to get the different logics is to use in the proper way negations and conjunctions.
- Whenever a semantics is finer than other, the logic characterizing the first will contain that for the latter, thus making trivial that relationship.

Unified Definition of the Logics

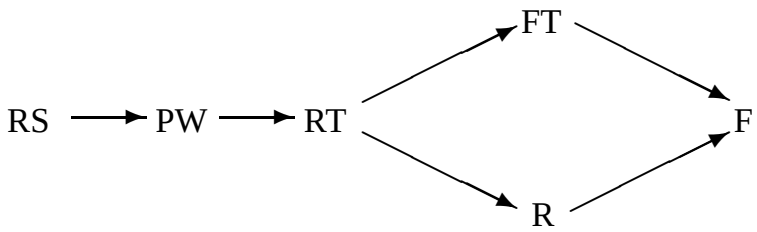
- In order to obtain a uniform presentation of the logics characterizing each of the semantics we look for (simple) set of formulas as large as possible.
- The key point to get the different logics is to use in the proper way negations and conjunctions.
- Whenever a semantics is finer than other, the logic characterizing the first will contain that for the latter, thus making trivial that relationship.

Unified Definition of the Logics

- In order to obtain a uniform presentation of the logics characterizing each of the semantics we look for (simple) set of formulas as large as possible.
- The key point to get the different logics is to use in the proper way negations and conjunctions.
- Whenever a semantics is finer than other, the logic characterizing the first will contain that for the latter, thus making trivial that relationship.

The layer of RS provides some illustrative examples

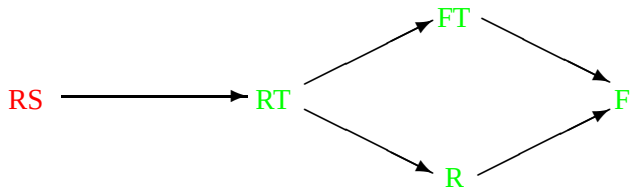
- The most important layer in the extended spectrum is that corresponding to ready simulation.

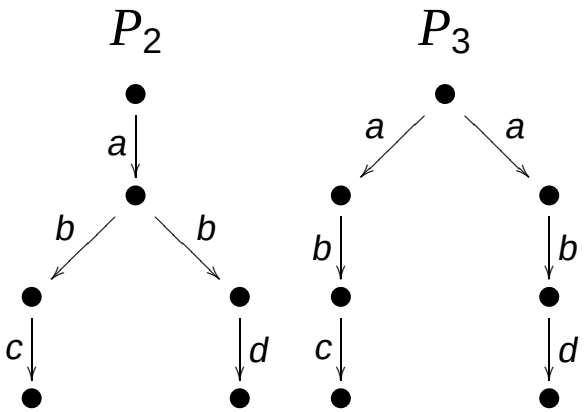


The layer corresponding to ready simulation

- $\mathcal{L}_l = \{a\top \mid a \in Act\}$.

Ready Simulation Logics



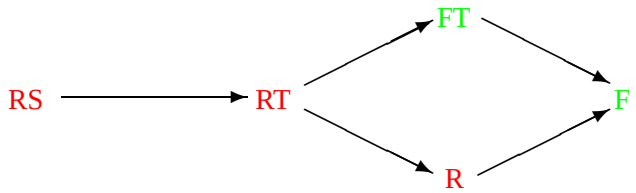


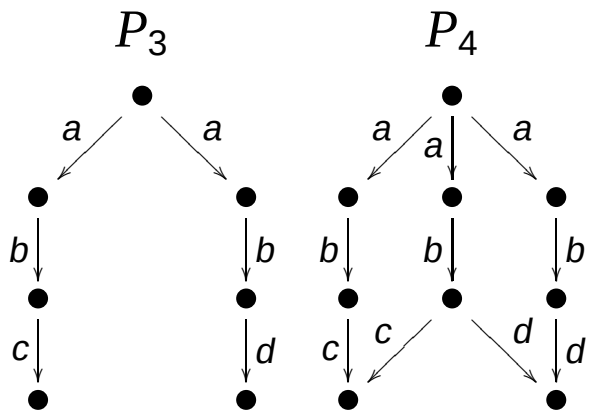
$$P_2 \not\preceq'_{RS} P_3$$

$P_2 \models a(bc \wedge bd)$, *but* P_3 *does not*.

- Branching semantics.
- Unrestricted use of conjunctions.

Readiness Semantics



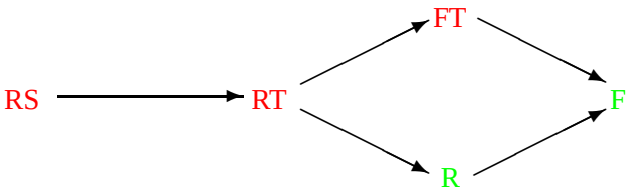


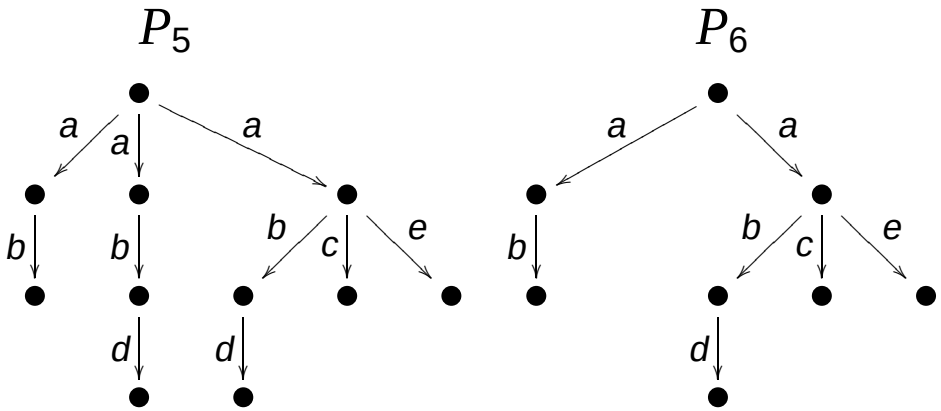
$$P_4 \not\sqsubseteq'_{\{RT,R\}} P_3$$

$P_4 \models ab(c \wedge d)$, *but* P_3 *does not*.

- Linear semantics (prefix operator).
- Positive information about initials actions.

Failure Trace Semantics



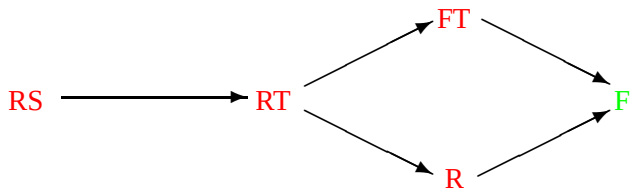


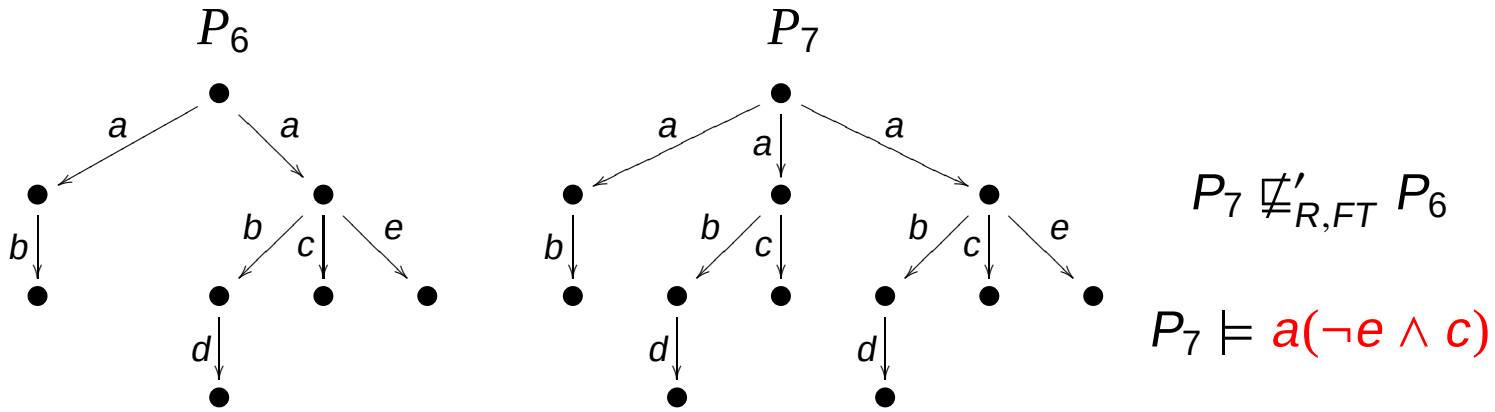
$$P_5 \not\preceq'_{\{RT,FT\}} P_6$$

$$P_5 \models a(\neg c \wedge b(\neg e \wedge d))$$

- Linear trace semantics (prefix operator).
- Negative information about initials actions.

Readiness and Failure Trace Semantics



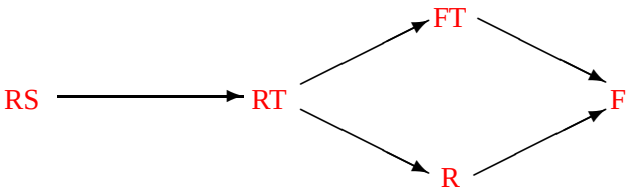


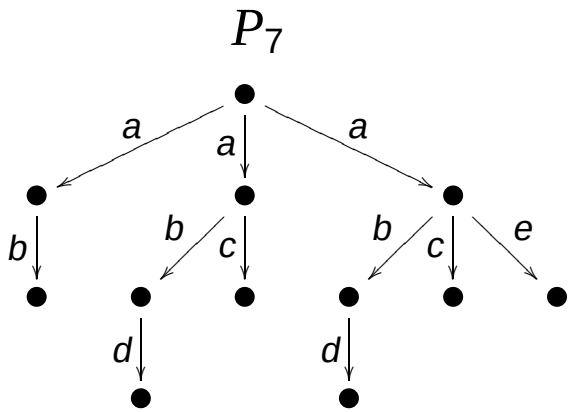
$P_7 \not\sqsubseteq'_{R,FT} P_6$

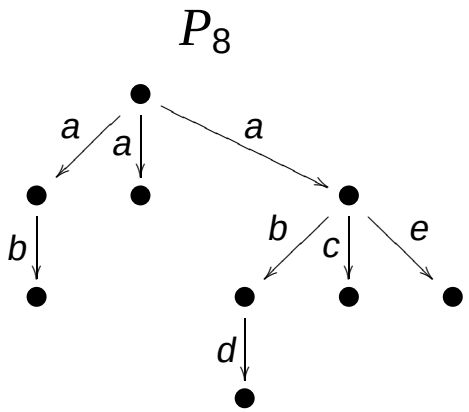
$P_7 \models a(\neg e \wedge c)$

- Linear (trace) semantics (prefix operator).
- Positive (resp. negative) information for case or R (resp. FT) about initials actions.

Failure Semantics







$P_8 \not\sqsubseteq'_F P_7$

$P_8 \models a(\neg b \wedge \neg c)$

- Linear semantics (prefix operator).
- Negative information about initials actions.

Uniform logical characterizations: Failures

DEFINITION (Negative closure $\mathcal{L}_N^{\neg}$)

Given a logical set $\mathcal{L}'_N$, we define $\mathcal{L}_N^{\neg}$ by:

- $\sigma \in \mathcal{L}'_N \Rightarrow \neg\sigma \in \mathcal{L}_N^{\neg}$
- $\sigma_i \in \mathcal{L}_N^{\neg}\ \forall i \in I \Rightarrow \bigwedge_{i\in I} \sigma_i \in \mathcal{L}_N^{\neg}$

DEFINITION (Failure semantics)

Inspired by the order $\leq_I^{lf\supseteq}$, we define the set of formulas $\mathcal{L}'_F$ by:

- $\top \in \mathcal{L}'_F$
- $\sigma \in \mathcal{L}^{\neg}{}_I \ \Rightarrow \sigma \in \mathcal{L}'_F$
- $\varphi \in \mathcal{L}'_F,\ \ a \in Act \Rightarrow a\varphi \in \mathcal{L}'_F$

Uniform logical characterizations: Failure Traces

DEFINITION (Negative closure $\mathcal{L}_N^{\neg}$)

Given a logical set $\mathcal{L}'_N$, we define $\mathcal{L}_N^{\neg}$ by:

- $\sigma \in \mathcal{L}'_N \Rightarrow \neg\sigma \in \mathcal{L}_N^{\neg}$
- $\sigma_i \in \mathcal{L}_N^{\neg}\ \forall i \in I \Rightarrow \bigwedge_{i\in I} \sigma_i \in \mathcal{L}_N^{\neg}$

DEFINITION (Failure trace semantics)

Inspired by the order $\leq_I^{\sqsupseteq}$, we define the set of formulas $\mathcal{L}'_{FT}$ by:

- $\top \in \mathcal{L}'_{FT}$
- $\varphi \in \mathcal{L}'_{FT},\quad \sigma \in \mathcal{L}^{\neg}{}_I \quad \Rightarrow \sigma \wedge \varphi \in \mathcal{L}'_{FT}$
- $\varphi \in \mathcal{L}'_{FT},\quad a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{FT}$

Uniform logical characterizations: Readiness

DEFINITION (Symmetric closure $\mathcal{L}_N^{\equiv}$)

Given a logical set $\mathcal{L}'_N$, we define $\mathcal{L}_N^{\equiv}$ by:

- $\sigma \in \mathcal{L}'_N \Rightarrow \sigma \in \mathcal{L}_N^{\equiv}$
- $\sigma \in \mathcal{L}'_N \Rightarrow \neg\sigma \in \mathcal{L}_N^{\equiv}$
- $\sigma_i \in \mathcal{L}_N^{\equiv}\ \forall i \in I \Rightarrow \bigwedge_{i\in I} \sigma_i \in \mathcal{L}_N^{\equiv}$

DEFINITION (Readiness semantics)

Inspired by the order $\leq_I^{If}$, we define the set of formulas $\mathcal{L}'_R$ by:

- $\top \in \mathcal{L}'_R$
- $\sigma \in \mathcal{L}_I^{\equiv} \Rightarrow \sigma \in \mathcal{L}'_R$
- $\varphi \in \mathcal{L}'_R,\ \ a \in Act \Rightarrow a\varphi \in \mathcal{L}'_R$

Uniform logical characterizations: Ready Traces

DEFINITION (Symmetric closure $\mathcal{L}_N^{\equiv}$)

Given a logical set $\mathcal{L}'_N$, we define $\mathcal{L}_N^{\equiv}$ by:

- $\sigma \in \mathcal{L}'_N \Rightarrow \sigma \in \mathcal{L}_N^{\equiv}$
- $\sigma \in \mathcal{L}'_N \Rightarrow \neg\sigma \in \mathcal{L}_N^{\equiv}$
- $\sigma_i \in \mathcal{L}_N^{\equiv}\ \forall i \in I \Rightarrow \bigwedge_{i\in I} \sigma_i \in \mathcal{L}_N^{\equiv}$

DEFINITION (Ready trace semantics)

Inspired by the order $\leq_I^I$, we define the set of formulas $\mathcal{L}'_{RT}$ by:

- $\top \in \mathcal{L}'_{RT}$
- $\varphi \in \mathcal{L}'_{RT},\ \ \sigma \in \mathcal{L}^{\equiv}{}_I \ \ \Rightarrow \sigma \wedge \varphi \in \mathcal{L}'_{RT}$
- $\varphi \in \mathcal{L}'_{RT},\ \ a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{RT}$

Uniform logical characterizations: Ready Simulation

DEFINITION (Ready simulation semantics)

Given a set of formulas $\mathcal{L}'_I$ defining a semantics I, we define the set of formulas $\mathcal{L}'_{IS}$ which is also denoted by $\mathcal{L}'_{RS}$ that defines the I-constrained simulation semantics by

- $\sigma \in \mathcal{L}'_I \Rightarrow \sigma \in \mathcal{L}'_{RS}$
- $\sigma \in \mathcal{L}'_I \Rightarrow \neg\sigma \in \mathcal{L}'_{RS}$
- *Given a set I,* $\varphi_i \in \mathcal{L}'_{RS}\ \forall i \in I \Rightarrow \bigwedge_{i\in I} \varphi_i \in \mathcal{L}'_{RS}$
- $\varphi \in \mathcal{L}'_{RS}, \quad a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{RS}$

THEOREM

We have (1) $\mathcal{L}_{RS} \sim \mathcal{L}'_{RS}$; (2) $\mathcal{L}_{RT} \sim \mathcal{L}'_{RT}$; (3) $\mathcal{L}_{FT} \sim \mathcal{L}'_{FT}$; (4) $\mathcal{L}_R \sim \mathcal{L}'_R$ and (5) $\mathcal{L}_F \sim \mathcal{L}'_F$.

Uniform logical characterizations: Ready Simulation

DEFINITION (Ready simulation semantics)

Given a set of formulas $\mathcal{L}'_I$ defining a semantics I, we define the set of formulas $\mathcal{L}'_{IS}$ which is also denoted by $\mathcal{L}'_{RS}$ that defines the I-constrained simulation semantics by

- $\sigma \in \mathcal{L}'_I \Rightarrow \sigma \in \mathcal{L}'_{RS}$
- $\sigma \in \mathcal{L}'_I \Rightarrow \neg\sigma \in \mathcal{L}'_{RS}$
- *Given a set I,* $\varphi_i \in \mathcal{L}'_{RS}\ \forall i \in I \Rightarrow \bigwedge_{i\in I} \varphi_i \in \mathcal{L}'_{RS}$
- $\varphi \in \mathcal{L}'_{RS}, \quad a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{RS}$

THEOREM

We have (1) $\mathcal{L}_{RS} \sim \mathcal{L}'_{RS}$; (2) $\mathcal{L}_{RT} \sim \mathcal{L}'_{RT}$; (3) $\mathcal{L}_{FT} \sim \mathcal{L}'_{FT}$; (4) $\mathcal{L}_R \sim \mathcal{L}'_R$ and (5) $\mathcal{L}_F \sim \mathcal{L}'_F$.

Uniform logical characterizations of all the semantics

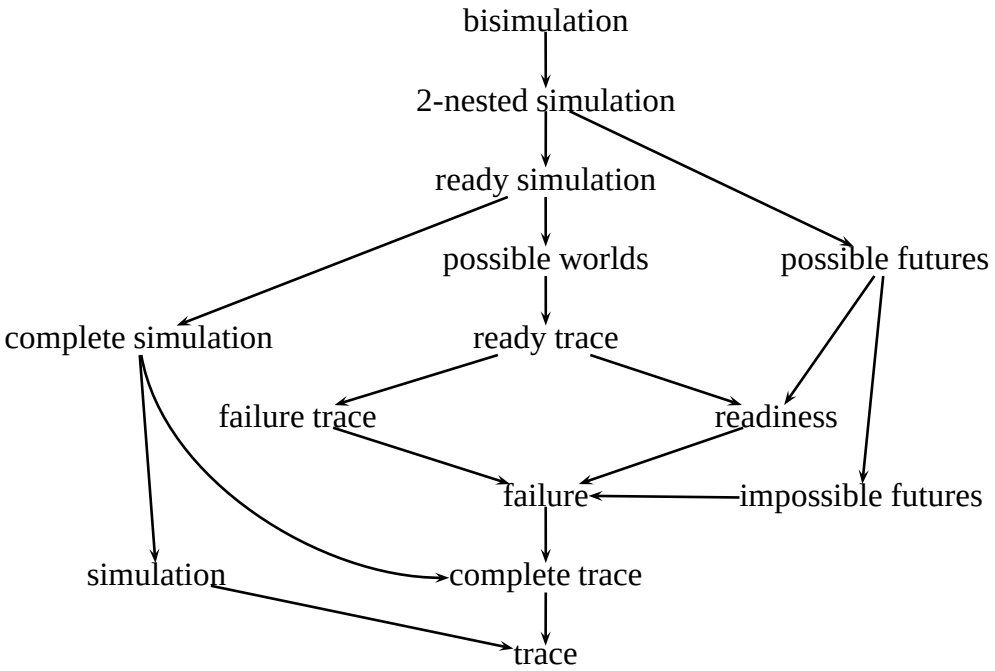
Logical characterizations of the semantics used as constraints in the N-constrained semantics

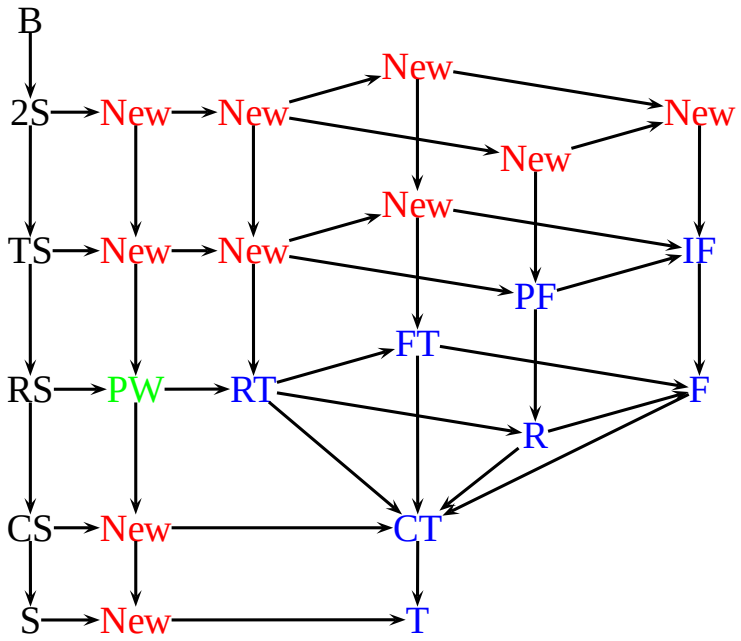
Formulas \ Constraints ($\mathcal{N}$)	U	C	I	T	S	B
$\top \in \mathcal{L}'_{\mathcal{N}}$	•	•	•	•	ν	ν
$\neg\top = \bot \in \mathcal{L}'_{\mathcal{N}}$	ν	ν	ν	ν	ν	ν
$\neg 0 \in \mathcal{L}'_{\mathcal{N}}$		•	•	ν	ν	ν
$a \in Act \Rightarrow a\top \in \mathcal{L}'_{\mathcal{N}}$			•	ν	ν	ν
$\varphi \in \mathcal{L}'_{\mathcal{N}},\ a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{\mathcal{N}}$				•	•	•
$\varphi_i \in \mathcal{L}'_{\mathcal{N}}\ \forall i \in I \Rightarrow \bigwedge_{i\in I} \varphi_i \in \mathcal{L}'_{\mathcal{N}}$					•	•
$\varphi \in \mathcal{L}'_{\mathcal{N}} \Rightarrow \neg\varphi \in \mathcal{L}'_{\mathcal{N}}$						•

Uniform logical characterizations of all the semantics

Formulas \ Semantics($\mathcal{Y}_N$)	$\leq_N^{\mathbf{lf}\supseteq}$	$\leq_N^{\mathbf{lf}}$	$\leq_N^{\mathbf{l}\supseteq}$	$\leq_N^{\mathbf{l}}$	D_N	NS	$N \in \{U, C, I, T, S\}$
	F	R	FT	RT	PW	RS	when $N = I$
$\top \in \mathcal{L}'_{\mathcal{Y}_N}$	•	•	•	•	•	ν	
$\varphi \in \mathcal{L}'_{\mathcal{Y}_N},\ a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{\mathcal{Y}_N}$	•	•	•	•	ν	•	
$\varphi \in \mathcal{L}_N^{\neg} \Rightarrow \varphi \in \mathcal{L}'_{\mathcal{Y}_N}$	•	ν	ν	ν	ν	ν	
$\varphi \in \mathcal{L}_N^{\equiv} \Rightarrow \varphi \in \mathcal{L}'_{\mathcal{Y}_N}$		•		ν	ν	ν	
$\varphi \in \mathcal{L}'_{\mathcal{Y}_N},\ \sigma \in \mathcal{L}_N^{\neg} \Rightarrow \sigma \wedge \varphi \in \mathcal{L}'_{\mathcal{Y}_N}$			•	ν	ν	ν	
$\varphi \in \mathcal{L}'_{\mathcal{Y}_N},\ \sigma \in \mathcal{L}_N^{\equiv} \Rightarrow \sigma \wedge \varphi \in \mathcal{L}'_{\mathcal{Y}_N}$				•	•	ν	
$X \subseteq Act,\ \varphi_a \in \mathcal{L}'_{\mathcal{Y}_N}\ \forall a \in X \Rightarrow \bigwedge_{a\in X} a\varphi_a \in \mathcal{L}'_{\mathcal{Y}_N}$					•	ν	
$\varphi_i \in \mathcal{L}'_{\mathcal{Y}_N}\ \forall i \in I \Rightarrow \bigwedge_{i\in I} \varphi_i \in \mathcal{L}'_{\mathcal{Y}_N}$						•	
$\varphi \in \mathcal{L}_N \Rightarrow \varphi \in \mathcal{L}'_{\mathcal{Y}_N}$						•	
$\varphi \in \mathcal{L}_N \Rightarrow \neg\varphi \in \mathcal{L}'_{\mathcal{Y}_N}$						•	

Linear time-branching time spectrum of semantics





Simulation is not enough

Plain simulations

- Compare processes based on the simple premise:

  "*you are better if you can do as much as me, and perhaps some additional new things*"

- Assume that all the executable actions are controlled by the user
  - No distinction between input and output actions.

Motivating examples

Machines

$$\begin{aligned} \text{onecoke} &: \text{coin?} \to \text{coke!} \to 0 \\ \text{cokeorlemonade} &: \text{coin?} \to ((\text{coke!} \to 0) + (\text{lemonade!} \to 0)) \end{aligned}$$

- onecoke $\lesssim_S$ cokeorlemonade.

  . . . But maybe I don't like lemonade.

- coke! and lemonade! should be considered as output actions.
  - I cannot choose between them!

Our solution

We defined two new notions of simulations and their logical characterizations in terms of a Hennessy-Milner logic.

Covariant-contravariant simulations

- Distinguishing between input and output actions.

Covariant-contravariant simulations

Given $P = (P, A, \rightarrow_P)$ and $Q = (Q, A, \rightarrow_Q)$, and $\{A^r, A^l, A^{bi}\}$ a partition of A, pSq implies

- For all $a \in A^r \cup A^{bi}$ and all $p \xrightarrow{a} p'$ there exists $q \xrightarrow{a} q'$ with $p' S q'$.
- For all $a \in A^l \cup A^{bi}$, and all $q \xrightarrow{a} q'$ there exists $p \xrightarrow{a} p'$ with $p' S q'$.

- If $A^r = A$ we have plain simulation.
- If $A^l = A$ we have plain "anti-simulation".
- If $A^{bi} = A$ we have bisimulation.

Conformance Simulations

- Reduction of non-determinism: $ap \gtrsim ap + aq$.

Conformance simulations

Given $P = (P, A, \rightarrow_P)$ and $Q = (Q, A, \rightarrow_Q)$. pRq implies:

- For all $a \in A$, if $p \xrightarrow{a}$, then $q \xrightarrow{a}$.
- For all $a \in A$ such that $q \xrightarrow{a} q'$ and $p \xrightarrow{a}$, there exists some p' with $p \xrightarrow{a} p'$ and $p' R q'$.

- $I(p) \subseteq I(q)$ guarantees Q has at least all the behaviors of P:

  $0 \lesssim_{CS} a$, $a \lesssim_{CS} a + b$, etc.

- The second clause is contravariant and establishes that a process can be "improved" by reducing the nondeterminism in it.

  $ab \gtrsim_{CS} ab + ac$, etc.

Conformance Simulations

- Reduction of non-determinism: $ap \gtrsim ap + aq$.

Conformance simulations

Given $P = (P, A, \rightarrow_P)$ and $Q = (Q, A, \rightarrow_Q)$. pRq implies:

- For all $a \in A$, if $p \xrightarrow{a}$, then $q \xrightarrow{a}$.
- For all $a \in A$ such that $q \xrightarrow{a} q'$ and $p \xrightarrow{a}$, there exists some p' with $p \xrightarrow{a} p'$ and $p' R q'$.

- $I(p) \subseteq I(q)$ guarantees Q has at least all the behaviors of P:

  $0 \lesssim_{CS} a$, $a \lesssim_{CS} a + b$, etc.

- The second clause is contravariant and establishes that a process can be "improved" by reducing the nondeterminism in it.

  $ab \gtrsim_{CS} ab + ac$, etc.

Conformance Simulations

- Reduction of non-determinism: $ap \gtrsim ap + aq$.

Conformance simulations

Given $P = (P, A, \rightarrow_P)$ and $Q = (Q, A, \rightarrow_Q)$. pRq implies:

- For all $a \in A$, if $p \xrightarrow{a}$, then $q \xrightarrow{a}$.
- For all $a \in A$ such that $q \xrightarrow{a} q'$ and $p \xrightarrow{a}$, there exists some p' with $p \xrightarrow{a} p'$ and $p' R q'$.

- $I(p) \subseteq I(q)$ guarantees Q has at least all the behaviors of P:

  $0 \lesssim_{CS} a$, $a \lesssim_{CS} a + b$, etc.

- The second clause is contravariant and establishes that a process can be "improved" by reducing the nondeterminism in it.

  $ab \gtrsim_{CS} ab + ac$, etc.

Simulation and anti-simulation logics

Plain simulation logic

- $p \models \mathrm{tt}$.
- $p \models \bigwedge_{i \in I} \varphi_i$ if $p \models \varphi_i$ for all $i \in I$.
- $p \models \langle a \rangle \varphi$, $a \in A^r \cup A^{bi}$, if there exists $p \xrightarrow{a} p'$ and $p' \models \varphi$.

Plain "anti-simulation" logic

- $p \not\models \mathrm{ff}$.
- $p \models \bigvee_{i \in I} \varphi_i$ if $p \models \varphi_i$ for some $i \in I$.
- $p \models [a] \varphi$, $a \in A^l \cup A^{bi}$, if $p' \models \varphi$ for all p' such that $p \xrightarrow{a} p'$.

Simulation and anti-simulation logics

Plain simulation logic

- $p \models \mathsf{tt}$.
- $p \models \bigwedge_{i \in I} \varphi_i$ if $p \models \varphi_i$ for all $i \in I$.
- $p \models \langle a \rangle \varphi$, $a \in A^r \cup A^{bi}$, if there exists $p \xrightarrow{a} p'$ and $p' \models \varphi$.

Plain "anti-simulation" logic

- $p \not\models \mathsf{ff}$.
- $p \models \bigvee_{i \in I} \varphi_i$ if $p \models \varphi_i$ for some $i \in I$.
- $p \models [a] \varphi$, $a \in A^l \cup A^{bi}$, if $p' \models \varphi$ for all p' such that $p \xrightarrow{a} p'$.

Simulation and anti-simulation logics

Plain simulation logic

- $p \models \mathsf{tt}$.
- $p \models \bigwedge_{i \in I} \varphi_i$ if $p \models \varphi_i$ for all $i \in I$.
- $p \models \langle a \rangle \varphi$, $a \in A^r \cup A^{bi}$, if there exists $p \xrightarrow{a} p'$ and $p' \models \varphi$.

Plain "anti-simulation" logic

- $p \not\models \mathsf{ff}$.
- $p \models \bigvee_{i \in I} \varphi_i$ if $p \models \varphi_i$ for some $i \in I$.
- $p \models [a] \varphi$, $a \in A^l \cup A^{bi}$, if $p' \models \varphi$ for all p' such that $p \xrightarrow{a} p'$.

Simulation and anti-simulation logics

Plain simulation logic

- $p \models \mathsf{tt}$.
- $p \models \bigwedge_{i \in I} \varphi_i$ if $p \models \varphi_i$ for all $i \in I$.
- $p \models \langle a \rangle \varphi$, $a \in A^r \cup A^{bi}$, if there exists $p \xrightarrow{a} p'$ and $p' \models \varphi$.

Plain "anti-simulation" logic

- $p \not\models \mathsf{ff}$.
- $p \models \bigvee_{i \in I} \varphi_i$ if $p \models \varphi_i$ for some $i \in I$.
- $p \models [a] \varphi$, $a \in A^l \cup A^{bi}$, if $p' \models \varphi$ for all p' such that $p \xrightarrow{a} p'$.

Simulation and anti-simulation logics

Plain simulation logic

- $p \models \mathsf{tt}$.
- $p \models \bigwedge_{i \in I} \varphi_i$ if $p \models \varphi_i$ for all $i \in I$.
- $p \models \langle a \rangle \varphi$, $a \in A^r \cup A^{bi}$, if there exists $p \xrightarrow{a} p'$ and $p' \models \varphi$.

Plain "anti-simulation" logic

- $p \not\models \mathsf{ff}$.
- $p \models \bigvee_{i \in I} \varphi_i$ if $p \models \varphi_i$ for some $i \in I$.
- $p \models [a] \varphi$, $a \in A^l \cup A^{bi}$, if $p' \models \varphi$ for all p' such that $p \xrightarrow{a} p'$.

Covariant-contravariant logic

- Making the union of plain simulation and plain anti-simulation logics we obtain covariant-contravariant logic.

Covariant-contravariant logic

- $p \models \mathrm{tt}$ and $p \not\models \mathrm{ff}$.
- $p \models \bigwedge_{i \in I} \varphi_i$ if $p \models \varphi_i$ for all $i \in I$.
- $p \models \bigvee_{i \in I} \varphi_i$ if $p \models \varphi_i$ for some $i \in I$.
- $p \models \langle a \rangle \varphi$ if there exists $p \xrightarrow{a} p'$ and $p' \models \varphi$.
- $p \models [a] \varphi$ if $p' \models \varphi$ for all p' such that $p \xrightarrow{a} p'$.

Covariant-contravariant logic

- Making the union of plain simulation and plain anti-simulation logics we obtain covariant-contravariant logic.

Covariant-contravariant logic

- $p \models \mathrm{tt}$ and $p \not\models \mathrm{ff}$.
- $p \models \bigwedge_{i \in I} \varphi_i$ if $p \models \varphi_i$ for all $i \in I$.
- $p \models \bigvee_{i \in I} \varphi_i$ if $p \models \varphi_i$ for some $i \in I$.
- $p \models \langle a \rangle \varphi$ if there exists $p \xrightarrow{a} p'$ and $p' \models \varphi$.
- $p \models [a] \varphi$ if $p' \models \varphi$ for all p' such that $p \xrightarrow{a} p'$.

Conformance logic

Conformance logic

- $p \models \mathrm{tt}$.
- $p \models \bigwedge_{i \in I} \varphi_i$ if $p \models \varphi_i$ for all $i \in I$.
- $p \models \bigvee_{i \in I} \varphi_i$ if $p \models \varphi_i$ for some $i \in I$.
- $p \models a\varphi$ if $p \xrightarrow{a}$ and $p' \models \varphi$ for all $p \xrightarrow{a} p'$.

- Captures $ap \gtrsim_{CS} ap + aq$.
- It is equivalent to $[a] \wedge \langle a \rangle$.

Proposition

If $p \lesssim_{CS} q$, then $p \models \varphi \implies q \models \varphi$ for all φ.

Conformance logic

Conformance logic

- $p \models \mathrm{tt}$.
- $p \models \bigwedge_{i \in I} \varphi_i$ if $p \models \varphi_i$ for all $i \in I$.
- $p \models \bigvee_{i \in I} \varphi_i$ if $p \models \varphi_i$ for some $i \in I$.
- $p \models a\varphi$ if $p \xrightarrow{a}$ and $p' \models \varphi$ for all $p \xrightarrow{a} p'$.

- Captures $ap \gtrsim_{CS} ap + aq$.
- It is equivalent to $[a] \wedge \langle a \rangle$.

Proposition

If $p \lesssim_{CS} q$, then $p \models \varphi \implies q \models \varphi$ for all φ.

Examples: Covariant-contravariant logic

Machines

$$\begin{aligned} \text{onecoke} &: \text{coin?} \to \text{coke!} \to 0 \\ \text{cokeorlemonade} &: \text{coin?} \to ((\text{coke!} \to 0) + (\text{lemonade!} \to 0)) \end{aligned}$$

- $\text{cokeorlemonade} \lesssim_{CC} \text{onecoke}$.

Logical formulas

- $\text{onecoke} \models \langle \text{coin?} \rangle\, [\text{lemonade!}]\, \text{ff}$.
  - Given a coin, we are sure that we are not going to get a lemonade.
- $\text{cokeorlemonade} \not\models \langle \text{coin?} \rangle\, [\text{lemonade!}]\, \text{ff}$.

Examples: Conformance logic

Machines

onecoke	:	coin? → coke! → 0
choice_coke_lemonade	:	(coin? → coke! → 0)+
		(coin? → lemonade! → 0)

- choice_coke_lemonade $\lesssim_{CS}$ onecoke.

Logical formulas

- onecoke $\models$ coin? coke! tt.
  - We always get a coke for a coin (Non-trivially!).
- choice_coke_lemonade $\not\models$ coin? coke! tt.

Outline

1 Introduction and Motivation
2 Classic Semantics of Processes
3 Unification of the Observational Semantics
4 Unification of the Equational Semantics
5 Unification of the Logical Semantics
6 Covariant-contravariant Semantics
7 Several New Semantics
- Possible Worlds Logic
- Meet and Join Semantics
- Partial Offers Semantics

Uniform logical characterizations: Possible Worlds Logic

DEFINITION (Possible Worlds semantics)

We define the formulas of $\mathcal{L}'_{D_I}$ which is also denoted by $\mathcal{L}'_{PW}$ by:

- $\top \in \mathcal{L}'_{PW}$
- $\varphi \in \mathcal{L}'_{PW}, \quad \sigma \in \mathcal{L}^{\equiv}{}_{I} \quad \Rightarrow \sigma \wedge \varphi \in \mathcal{L}'_{PW}$
- $X \subseteq Act, \quad \varphi_a \in \mathcal{L}'_{PW} \ \forall a \in X \Rightarrow \bigwedge_{a \in X} a\varphi_a \in \mathcal{L}'_{PW}$

In this case $\mathcal{L}_{PW}$ and $\mathcal{L}'_{PW}$ are not equivalent, but this is caused by the fact that the original logical characterization $\mathcal{L}_{PW}$ was wrong!

Uniform logical characterizations: Possible Worlds Logic

DEFINITION (Possible Worlds semantics)

We define the formulas of $\mathcal{L}'_{D_I}$ which is also denoted by $\mathcal{L}'_{PW}$ by:

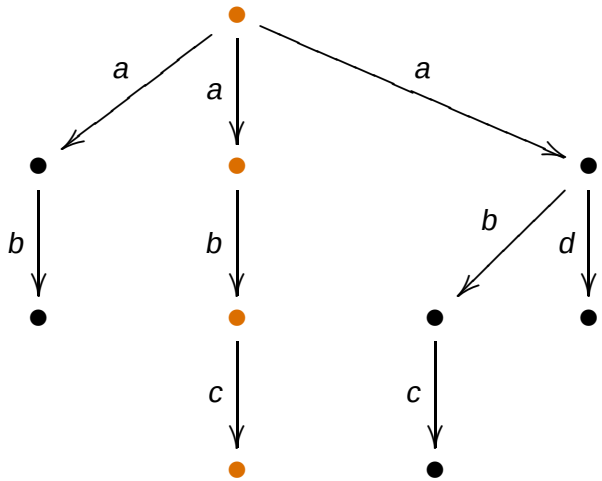
- $\top \in \mathcal{L}'_{PW}$
- $\varphi \in \mathcal{L}'_{PW}, \quad \sigma \in \mathcal{L}^{\equiv}{}_I \quad \Rightarrow \sigma \wedge \varphi \in \mathcal{L}'_{PW}$
- $X \subseteq Act, \quad \varphi_a \in \mathcal{L}'_{PW} \ \ \forall a \in X \Rightarrow \bigwedge_{a \in X} a\varphi_a \in \mathcal{L}'_{PW}$

In this case $\mathcal{L}_{PW}$ and $\mathcal{L}'_{PW}$ are not equivalent, but this is caused by the fact that the original logical characterization $\mathcal{L}_{PW}$ was wrong!

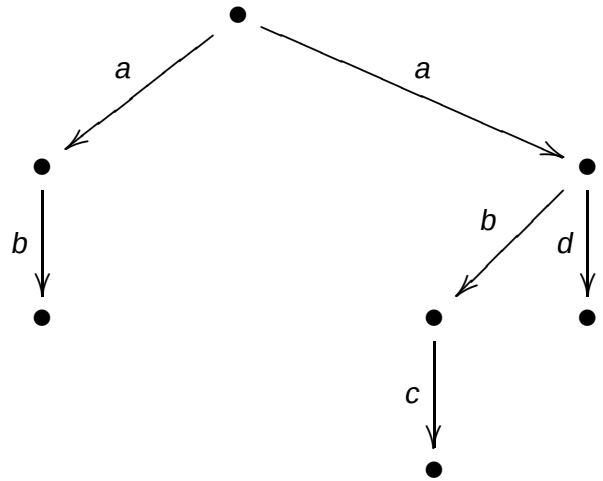
A Critical Example for Possible Worlds Logical Characterization

$$\varphi \equiv a(\neg d \wedge bc) \in \mathcal{L}'_{PW}$$

$P \models \varphi$



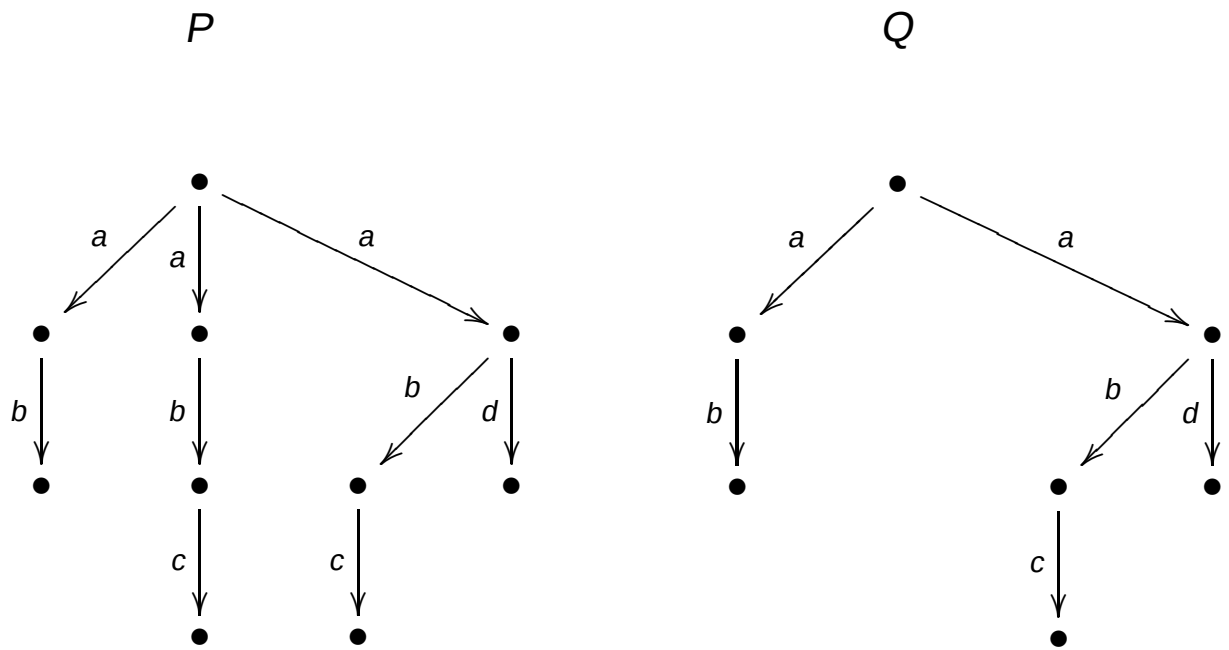
$Q \not\models \varphi$



- $\top \in \mathcal{L}'_{PW}$
- $\varphi \in \mathcal{L}'_{PW}, \quad \sigma \in \mathcal{L}^{\equiv}{}_{I} \quad \Rightarrow \sigma \wedge \varphi \in \mathcal{L}'_{PW}$
- $X \subseteq Act, \quad \varphi_a \in \mathcal{L}'_{PW} \quad \forall a \in X \Rightarrow \bigwedge_{a \in X} a\varphi_a \in \mathcal{L}'_{PW}$

A Critical Example for Possible Worlds Logical Characterization

$$\varphi \equiv a(\neg d \wedge bc) \notin \mathcal{L}_{PW}$$



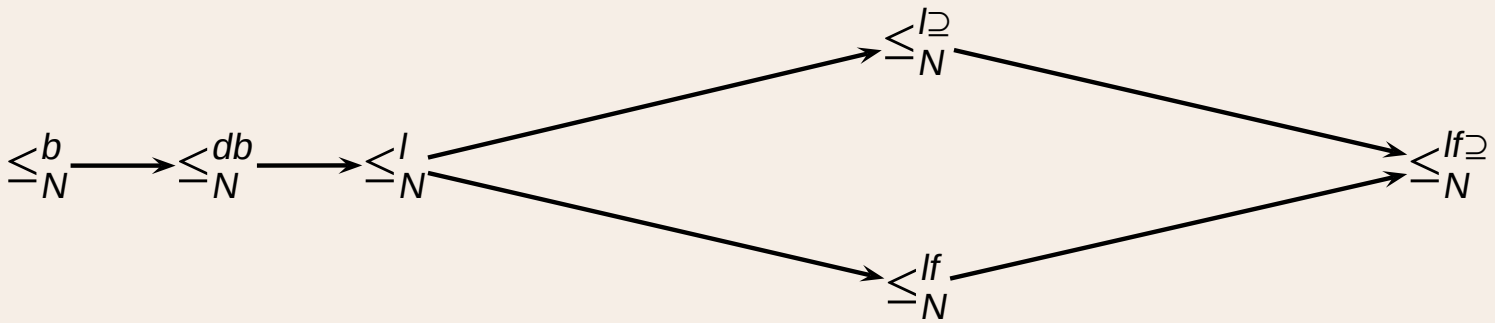
- $X \subseteq Act \Rightarrow \widetilde{X} \in \mathcal{L}_{PW}$
- $X \subseteq Act, \quad \varphi_a \in \mathcal{L}_{PW} \quad \forall a \in X \Rightarrow \bigwedge_{a \in X} a\varphi_a \in \mathcal{L}_{PW}$

Outline

1 Introduction and Motivation
2 Classic Semantics of Processes
3 Unification of the Observational Semantics
4 Unification of the Equational Semantics
5 Unification of the Logical Semantics
6 Covariant-contravariant Semantics
7 Several New Semantics
- Possible Worlds Logic
- Meet and Join Semantics
- Partial Offers Semantics

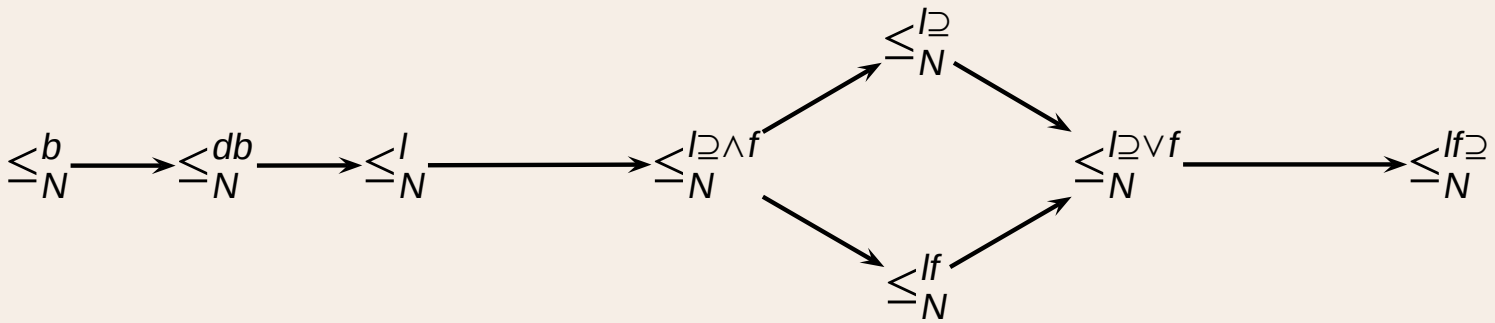
Meet and Join Semantics

Revising the Layer of the Extended Spectrum



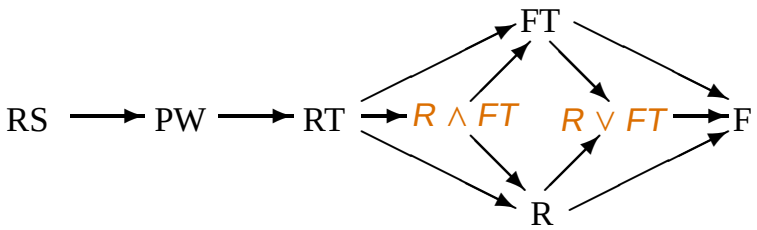
Meet and Join Semantics

Revising the Layer of the Extended Spectrum



Logical Characterization of Meet and join semantics

- There are another two semantics in each layer of the extended spectrum.
- For the particular case $N = I$, the meet semantics $R \vee F$ has been previously studied by Roscoe.



The double diamond below ready simulation

- These semantics are in the linear side of the spectrum, therefore they have a similar structure to those linear semantics studied before.

Logical Characterization of Meet and join semantics

DEFINITION (Join semantics)

We define the set of formulas $\mathcal{L}'_{\leq_I^{\supseteq \wedge f}}$*, as that generated by the clauses:*

- $\top \in \mathcal{L}'_{\leq_I^{\supseteq \wedge f}}$
- $\varphi \in \mathcal{L}'_{\leq_I^{\supseteq \wedge f}}, \quad \sigma \in \mathcal{L}^{\neg}{}_I \quad \Rightarrow \sigma \wedge \varphi \in \mathcal{L}'_{\leq_I^{\supseteq \wedge f}} \; ; \; \sigma \in \mathcal{L}^{\equiv}{}_I \quad \Rightarrow \sigma \in \mathcal{L}'_{\leq_I^{\supseteq \wedge f}}$
- $\varphi \in \mathcal{L}'_{\leq_I^{\supseteq \wedge f}}, \quad a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{\leq_I^{\supseteq \wedge f}}$

DEFINITION (Meet semantics)

We define the set of formulas $\mathcal{L}'_{\leq_I^{\supseteq \vee f}}$ *as that generated by the clauses:*

- $\top \in \mathcal{L}'_{\leq_I^{\supseteq \vee f}}$
- $\sigma, \sigma_j \in \mathcal{L}'{}_I \quad \forall j \in J \Rightarrow (\sigma \wedge \bigwedge_{j \in J} \neg \sigma_j \top) \in \mathcal{L}_{\leq_I^{\supseteq \vee f}}$
- $\varphi \in \mathcal{L}'_{\leq_I^{\supseteq \vee f}}, \quad a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{\leq_I^{\supseteq \vee f}}$

Logical Characterization of Meet and join semantics

DEFINITION (Join semantics)

We define the set of formulas $\mathcal{L}'_{\leq_N^{I\supseteq\wedge f}}$*, as that generated by the clauses:*

- $\top \in \mathcal{L}'_{\leq_N^{I\supseteq\wedge f}}$
- $\varphi \in \mathcal{L}'_{\leq_N^{I\supseteq\wedge f}}, \quad \sigma \in \mathcal{L}^{\neg}{}_N \quad \Rightarrow \sigma \wedge \varphi \in \mathcal{L}'_{\leq_N^{I\supseteq\wedge f}} \; ; \; \sigma \in \mathcal{L}^{\equiv}{}_N \quad \Rightarrow \sigma \in \mathcal{L}'_{\leq_N^{I\supseteq\wedge f}}$
- $\varphi \in \mathcal{L}'_{\leq_N^{I\supseteq\wedge f}}, \quad a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{\leq_N^{I\supseteq\wedge f}}$

DEFINITION (Meet semantics)

We define the set of formulas $\mathcal{L}'_{\leq_N^{I\supseteq\vee f}}$ *as that generated by the clauses:*

- $\top \in \mathcal{L}'_{\leq_N^{I\supseteq\vee f}}$
- $\sigma, \sigma_j \in \mathcal{L}'_N \quad \forall j \in J \Rightarrow (\sigma \wedge \bigwedge_{j\in J} \neg\sigma_j \top) \in \mathcal{L}_{\leq_N^{I\supseteq\vee f}}$
- $\varphi \in \mathcal{L}'_{\leq_N^{I\supseteq\vee f}}, \quad a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{\leq_N^{I\supseteq\vee f}}$

Meet and Join Semantics

Axiomatisation

$$I(x) = I(y) \Rightarrow x \sqsubseteq x + y$$

$$\left.\begin{array}{c} I(x) \supseteq I(y) \\ \vee \\ I(w) \subseteq I(y) \end{array}\right\} \Rightarrow a(x + y) \sqsubseteq ax + a(y + w)$$

$R \vee FT$

.

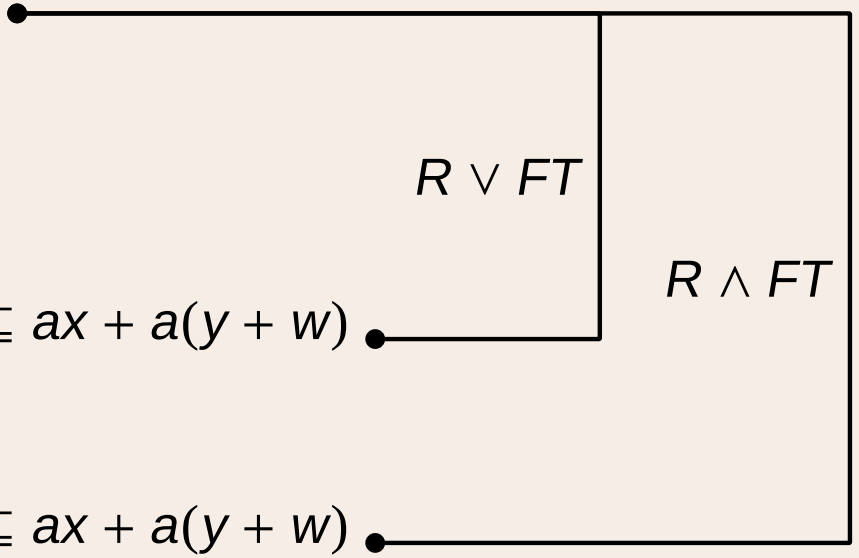
Meet and Join Semantics

Axiomatisation

$$I(x) = I(y) \Rightarrow x \sqsubseteq x + y$$

$$\left.\begin{array}{l} I(x) \supseteq I(y) \\ \quad\vee \\ I(w) \subseteq I(y) \end{array}\right\} \Rightarrow a(x+y) \sqsubseteq ax + a(y+w)$$

$$\left.\begin{array}{l} I(x) \supseteq I(y) \\ \quad\wedge \\ I(w) \subseteq I(y) \end{array}\right\} \Rightarrow a(x+y) \sqsubseteq ax + a(y+w)$$



Outline

1 Introduction and Motivation
2 Classic Semantics of Processes
3 Unification of the Observational Semantics
4 Unification of the Equational Semantics
5 Unification of the Logical Semantics
6 Covariant-contravariant Semantics
7 Several New Semantics
- Possible Worlds Logic
- Meet and Join Semantics
- Partial Offers Semantics

Partial offers trace and partial offer semantics

- Using a new closure we can define two more semantics at each layer of the spectrum.
- These semantics are defined by observing partial offers along a computation or just at its end.
- Duality between failures and partial offers causes the picture of the complete layer of linear semantics for each N to become two diamonds sharing the side corresponding to the readies-based semantics.

Partial offers trace semantics

DEFINITION (Positive closure $\mathcal{L}_N^{\surd}$)

Given a logical set $\mathcal{L}'_N$ with $N \in \{U, C, I, T, S\}$, we define:

- $\sigma \in \mathcal{L}'_N \Rightarrow \sigma \in \mathcal{L}_N^{\surd}$
- $\sigma_i \in \mathcal{L}_N^{\surd}\ \forall i \in I \Rightarrow \bigwedge_{i \in I} \sigma_i \in \mathcal{L}_N^{\surd}$

DEFINITION (Partial offers trace semantics)

For the constraint N is that defined by the logic $\mathcal{L}'_{POT}$ with

- $\top \in \mathcal{L}'_{POT}$
- $\varphi \in \mathcal{L}'_{POT},\ \sigma \in \mathcal{L}_N^{\surd} \Rightarrow \sigma \wedge \varphi \in \mathcal{L}'_{POT}$
- $\varphi \in \mathcal{L}'_{POT},\quad a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{POT}$

Partial offer semantics

DEFINITION (Positive closure $\mathcal{L}_N^{\surd}$)

Given a logical set $\mathcal{L}'_N$ with $N \in \{U, C, I, T, S\}$, we define:

- $\sigma \in \mathcal{L}'_N \Rightarrow \sigma \in \mathcal{L}_N^{\surd}$
- $\sigma_i \in \mathcal{L}_N^{\surd}\ \forall i \in I \Rightarrow \bigwedge_{i \in I} \sigma_i \in \mathcal{L}_N^{\surd}$

DEFINITION (Partial offer semantics)

For the constraint N is that defined by the logic $\mathcal{L}'_{PO}$ with

- $\top \in \mathcal{L}'_{PO}$
- $\sigma \in \mathcal{L}_N^{\surd} \Rightarrow \sigma \in \mathcal{L}'_{PO}$
- $\varphi \in \mathcal{L}'_{PO},\quad a \in Act \Rightarrow a\varphi \in \mathcal{L}'_{PO}$