Bisimulation and Logic
Lecture 2

Colin Stirling

Laboratory for Foundations of Computer Science (LFCS)
School of Informatics
Edinburgh University

Summer School on Model Checking
Ziyu Hotel, Beijing
Oct 11–16 2010

Two independent origins of bisimulation

- Behavioural equivalence between concurrent processes (Park, Hennessy + Milner)
- Model theory of modal logic (van Benthem)

Process Calculi

- Introduced syntax of CCS: prefix, sum, parallel composition, restriction, renaming
- Introduced two types of transition $\xrightarrow{a}$ and $\xRightarrow{a}$ and rules for their derivation
- Introduced two types of transition graph that abstracts from derivation of transitions

Process Calculi

- Introduced syntax of CCS: prefix, sum, parallel composition, restriction, renaming
- Introduced two types of transition $\xrightarrow{a}$ and $\xRightarrow{a}$ and rules for their derivation
- Introduced two types of transition graph that abstracts from derivation of transitions
- Lots of variants such as ACP, CSP, ...
- Lots of extensions (time, probabilities, locations ...)

Process equivalence: motivation

- "The sequence of actions $a_1 \dots a_n$ must be carried out cyclically starting with a_1" (the scheduler)
- This property cannot be expressed in temporal logic; is expressible in mu-calculus)

Process equivalence: motivation

- "The sequence of actions $a_1 \ldots a_n$ must be carried out cyclically starting with a_1" (the scheduler)
- This property cannot be expressed in temporal logic; is expressible in mu-calculus)
- More natural way of specifying this:
  When all actions but $a_1, \ldots, a_n$ are restricted, the system should "behave like" the process P, defined by

$$P \stackrel{\mathrm{def}}{=} \mathtt{a}_1.\mathtt{a}_2.\ldots.\mathtt{a}_\mathtt{n}.P$$

Process equivalence: motivation

- "The sequence of actions $a_1 \ldots a_n$ must be carried out cyclically starting with a_1" (the scheduler)
- This property cannot be expressed in temporal logic; is expressible in mu-calculus)
- More natural way of specifying this:
  When all actions but $a_1, \ldots, a_n$ are restricted, the system should "behave like" the process P, defined by

$$P \stackrel{\text{def}}{=} \mathtt{a}_1.\mathtt{a}_2.\ldots.\mathtt{a}_\mathtt{n}.P$$

- Generally: many systems are informally specified by "behave like" statements.

Process equivalence: motivation

- "The sequence of actions $a_1 \ldots a_n$ must be carried out cyclically starting with a_1" (the scheduler)
- This property cannot be expressed in temporal logic; is expressible in mu-calculus)
- More natural way of specifying this:
  When all actions but $a_1, \ldots, a_n$ are restricted, the system should "behave like" the process P, defined by

$$P \stackrel{\text{def}}{=} \mathtt{a}_1.\mathtt{a}_2.\ldots.\mathtt{a}_\mathtt{n}.P$$

- Generally: many systems are informally specified by "behave like" statements.

Process equivalence: motivation

- "The sequence of actions $a_1 \ldots a_n$ must be carried out cyclically starting with a_1" (the scheduler)
- This property cannot be expressed in temporal logic; is expressible in mu-calculus)
- More natural way of specifying this:
  When all actions but $a_1, \ldots, a_n$ are restricted, the system should "behave like" the process P, defined by

$$P \stackrel{\text{def}}{=} \mathtt{a}_1.\mathtt{a}_2.\ldots.\mathtt{a}_\mathtt{n}.P$$

- Generally: many systems are informally specified by "behave like" statements.
- But how to formalise "behavioural equivalence"?

Wish list

1. Behavioural equivalence should be an equivalence relation, reflexive, symmetric and transitive.

Wish list

1. Behavioural equivalence should be an equivalence relation, reflexive, symmetric and transitive.
2. Processes that may terminate (deadlock) should not be equivalent to processes that may not terminate (deadlock).

Wish list

1. Behavioural equivalence should be an equivalence relation, reflexive, symmetric and transitive.
2. Processes that may terminate (deadlock) should not be equivalent to processes that may not terminate (deadlock).
3. Congruence: if a component Q of P is replaced by an equivalent component Q' yielding P', then P and P' should also be equivalent.

Wish list

1. Behavioural equivalence should be an equivalence relation, reflexive, symmetric and transitive.
2. Processes that may terminate (deadlock) should not be equivalent to processes that may not terminate (deadlock).
3. Congruence: if a component Q of P is replaced by an equivalent component Q' yielding P', then P and P' should also be equivalent.
4. Two processes should be equivalent iff they satisfy exactly the same properties (such as expressible in modal or temporal logic)

Wish list

1. Behavioural equivalence should be an equivalence relation, reflexive, symmetric and transitive.
2. Processes that may terminate (deadlock) should not be equivalent to processes that may not terminate (deadlock).
3. Congruence: if a component Q of P is replaced by an equivalent component Q' yielding P', then P and P' should also be equivalent.
4. Two processes should be equivalent iff they satisfy exactly the same properties (such as expressible in modal or temporal logic)
5. It should abstract from silent actions.

Wish list

1. Behavioural equivalence should be an equivalence relation, reflexive, symmetric and transitive.
2. Processes that may terminate (deadlock) should not be equivalent to processes that may not terminate (deadlock).
3. Congruence: if a component Q of P is replaced by an equivalent component Q' yielding P', then P and P' should also be equivalent.
4. Two processes should be equivalent iff they satisfy exactly the same properties (such as expressible in modal or temporal logic)
5. It should abstract from silent actions.

We deal first with conditions $1 - 4$

A first candidate: trace equivalence

- A trace of a process E is a sequence w of actions such that $E \xrightarrow{w} F$ for some process F.

A first candidate: trace equivalence

- A trace of a process E is a sequence w of actions such that $E \xrightarrow{w} F$ for some process F.
- E and F are trace-equivalent if they have the same traces.

A first candidate: trace equivalence

- A trace of a process E is a sequence w of actions such that $E \xrightarrow{w} F$ for some process F.
- E and F are trace-equivalent if they have the same traces.
- This notion satisfies 1 and 3, but not 2.

A first candidate: trace equivalence

- A trace of a process E is a sequence w of actions such that $E \xrightarrow{w} F$ for some process F.
- E and F are trace-equivalent if they have the same traces.
- This notion satisfies 1 and 3, but not 2.
- Counterexample. $\mathtt{Cl}$, $\mathtt{Cl}'$ trace equivalent

$$\begin{aligned} \mathtt{Cl} &\stackrel{\text{def}}{=} \mathtt{tick}.\mathtt{Cl} \\ \mathtt{Cl}' &\stackrel{\text{def}}{=} \mathtt{tick}.\mathtt{Cl}' + \mathtt{tick}.0 \end{aligned}$$

A second candidate: completed trace equivalence

- A completed trace of E is a sequence w of actions such that $E \stackrel{w}{\longrightarrow} F$ for some process F that cannot execute any action

A second candidate: completed trace equivalence

- A completed trace of E is a sequence w of actions such that $E \xrightarrow{w} F$ for some process F that cannot execute any action
- E and F are completed trace equivalent if they have the same traces and the same completed traces

A second candidate: completed trace equivalence

- A completed trace of E is a sequence w of actions such that $E \xrightarrow{w} F$ for some process F that cannot execute any action
- E and F are completed trace equivalent if they have the same traces and the same completed traces
- This notion satisfies 1 and 2, but not 3.

$$\begin{aligned}
\texttt{Ven}_1 &\stackrel{\text{def}}{=} \texttt{1p}.\texttt{1p}.(\texttt{tea}.\texttt{Ven}_1 + \texttt{coffee}.\texttt{Ven}_1) \\
\texttt{Ven}_2 &\stackrel{\text{def}}{=} \texttt{1p}.(\texttt{1p}.\texttt{tea}.\texttt{Ven}_2 + \texttt{1p}.\texttt{coffee}.\texttt{Ven}_2) \\
\texttt{Use} &\stackrel{\text{def}}{=} \overline{\texttt{1p}}.\overline{\texttt{1p}}.\overline{\texttt{tea}}.\overline{\texttt{ok}}.0
\end{aligned}$$

A second candidate: completed trace equivalence

- A completed trace of E is a sequence w of actions such that $E \xrightarrow{w} F$ for some process F that cannot execute any action
- E and F are completed trace equivalent if they have the same traces and the same completed traces
- This notion satisfies 1 and 2, but not 3.

$$
\begin{aligned}
\texttt{Ven}_1 &\stackrel{\text{def}}{=} \texttt{1p}.\texttt{1p}.(\texttt{tea}.\texttt{Ven}_1 + \texttt{coffee}.\texttt{Ven}_1) \\
\texttt{Ven}_2 &\stackrel{\text{def}}{=} \texttt{1p}.(\texttt{1p}.\texttt{tea}.\texttt{Ven}_2 + \texttt{1p}.\texttt{coffee}.\texttt{Ven}_2) \\
\texttt{Use} &\stackrel{\text{def}}{=} \overline{\texttt{1p}}.\overline{\texttt{1p}}.\overline{\texttt{tea}}.\overline{\texttt{ok}}.0
\end{aligned}
$$

- $\texttt{Ven}_1$ and $\texttt{Ven}_2$ are completed-trace equivalent, but $(\texttt{Ven}_1 \mid \texttt{Use})\backslash K$ and $(\texttt{Ven}_2 \mid \texttt{Use})\backslash K$, where $K = \{\texttt{1p}, \texttt{tea}, \texttt{coffee}\}$, are not.

A third candidate: bisimulation equivalence

- A binary relation B between processes is a bisimulation provided that, whenever $(E, F) \in B$ and $a \in \mathsf{A}$,

A third candidate: bisimulation equivalence

- A binary relation B between processes is a bisimulation provided that, whenever $(E, F) \in B$ and $a \in \mathsf{A}$,
- if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' such that $(E', F') \in B$ and

A third candidate: bisimulation equivalence

- A binary relation B between processes is a bisimulation provided that, whenever $(E, F) \in B$ and $a \in \mathsf{A}$,
- if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' such that $(E', F') \in B$

  and
- if $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ for some E' such that $(E', F') \in B$

A third candidate: bisimulation equivalence

- A binary relation B between processes is a bisimulation provided that, whenever $(E, F) \in B$ and $a \in \mathsf{A}$,
- if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' such that $(E', F') \in B$ and
- if $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ for some E' such that $(E', F') \in B$
- E and F are bisimulation equivalent (or bisimilar) if there is a bisimulation relation B such that $(E, F) \in B$.

A third candidate: bisimulation equivalence

- A binary relation B between processes is a bisimulation provided that, whenever $(E, F) \in B$ and $a \in \mathrm{A}$,
- if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' such that $(E', F') \in B$ and
- if $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ for some E' such that $(E', F') \in B$
- E and F are bisimulation equivalent (or bisimilar) if there is a bisimulation relation B such that $(E, F) \in B$.
- We write $E \sim F$ if E and F are bisimilar

Examples

- $\mathtt{Cl} \stackrel{\text{def}}{=} \mathtt{tick.Cl} \qquad \mathtt{Cl}_2 \stackrel{\text{def}}{=} \mathtt{tick.tick.Cl}_2$

Examples

- $\texttt{Cl} \stackrel{\text{def}}{=} \texttt{tick.Cl} \qquad \texttt{Cl}_2 \stackrel{\text{def}}{=} \texttt{tick.tick.Cl}_2$
- $B_2 = \{(\texttt{Cl}, \texttt{Cl}_2), (\texttt{Cl}, \texttt{tick.Cl}_2)\}$ is a bisimulation.

Examples

- $\texttt{Cl} \stackrel{\text{def}}{=} \texttt{tick.Cl} \qquad \texttt{Cl}_2 \stackrel{\text{def}}{=} \texttt{tick.tick.Cl}_2$
- $B_2 = \{(\texttt{Cl}, \texttt{Cl}_2), (\texttt{Cl}, \texttt{tick.Cl}_2)\}$ is a bisimulation.
- $\mathsf{a.(b.0 + c.0)} \qquad \mathsf{a.b.0 + a.c.0}$

Examples

- $\mathtt{Cl} \stackrel{\text{def}}{=} \mathtt{tick.Cl} \qquad \mathtt{Cl_2} \stackrel{\text{def}}{=} \mathtt{tick.tick.Cl_2}$
- $B_2 = \{(\mathtt{Cl}, \mathtt{Cl_2}), (\mathtt{Cl}, \mathtt{tick.Cl_2})\}$ is a bisimulation.
- $\mathsf{a.(b.0 + c.0)} \qquad \mathsf{a.b.0 + a.c.0}$
- Not bisimilar

Game interpretation

Board:	Transition systems of E and F.
Material:	Two (identical) pebbles initially on the states E and F.
Players:	R (refuter) and V (verifier),
	R and V take turns, R moves first.
R-move:	Choose any of the two pebbles
	Move pebble across any transition
V-move:	Choose the other pebble
	choose a transition having the same label
	move pebble across it
R wins if:	V cannot reply to his last move.
V wins if:	R cannot move or
	the game goes on forever.
	(i.e., a draw counts as a win for V).
Theorem:	R can force a win iff E and F are not bisimilar.
	V can force a win iff E and F are bisimilar.

Bisimilarity is an equivalence relation

- Theorem : $E \sim E$

Bisimilarity is an equivalence relation

- Theorem : $E \sim E$
- Theorem: if $E \sim F$ then $F \sim E$.

Bisimilarity is an equivalence relation

- Theorem : $E \sim E$
- Theorem: if $E \sim F$ then $F \sim E$.
- Theorem : if $E \sim F$ and $F \sim G$, then $E \sim G$.

Bisimilarity is an equivalence relation

- Theorem : $E \sim E$
- Theorem: if $E \sim F$ then $F \sim E$.
- Theorem : if $E \sim F$ and $F \sim G$, then $E \sim G$.
  Proof: Since $E \sim F$, $(E, F) \in B_1$ for some bisimulation B_1. Since $F \sim G$, $(F, G) \in B_2$ for some bisimulation B_2. So $(E, G) \in B_1 \circ B_2$. We show that $B_1 \circ B_2$ is a bisimulation.

Bisimilarity is an equivalence relation

- Theorem : $E \sim E$
- Theorem: if $E \sim F$ then $F \sim E$.
- Theorem : if $E \sim F$ and $F \sim G$, then $E \sim G$.
  Proof: Since $E \sim F$, $(E, F) \in B_1$ for some bisimulation B_1. Since $F \sim G$, $(F, G) \in B_2$ for some bisimulation B_2. So $(E, G) \in B_1 \circ B_2$. We show that $B_1 \circ B_2$ is a bisimulation. Let $(H_1, H_2) \in B_1 \circ B_2$ and $H_1 \xrightarrow{a} H_1'$. We find H_2' such that $H_2 \xrightarrow{a} H_2'$ and $(H_1', H_2') \in B_1 \circ B_2$. Since $(H_1, H_2) \in B_1 \circ B_2$, there is H such that $(H_1, H) \in B_1$ and $(H, H_2) \in B_2$. Since B_1 is bisimulation, there is H' such that $H \xrightarrow{a} H'$ and $(H_1', H') \in B_1$. Since B_2 is bisimulation, there is H_2' such that $H_2 \xrightarrow{a} H_2'$ and $(H', H_2') \in B_2$. Since $(H_1', H') \in B_1$ and $(H', H_2') \in B_2$, we have $(H_1', H_2') \in B_1 \circ B_2$.

Bisimilarity is a congruence

Proposition: If $E \sim F$, then for any process G, for any set of actions K, for any action a and for any renaming function f,

1. $a.E \sim a.F$

Bisimilarity is a congruence

Proposition: If $E \sim F$, then for any process G, for any set of actions K, for any action a and for any renaming function f,

1. $a.E \sim a.F$
2. $E + G \sim F + G$

Bisimilarity is a congruence

Proposition: If $E \sim F$, then for any process G, for any set of actions K, for any action a and for any renaming function f,

1. $a.E \sim a.F$
2. $E + G \sim F + G$
3. $E \mid G \sim F \mid G$

Bisimilarity is a congruence

Proposition: If $E \sim F$, then for any process G, for any set of actions K, for any action a and for any renaming function f,

1. $a.E \sim a.F$
2. $E + G \sim F + G$
3. $E \mid G \sim F \mid G$
4. $E[f] \sim F[f]$

Bisimilarity is a congruence

Proposition: If $E \sim F$, then for any process G, for any set of actions K, for any action a and for any renaming function f,

1. $a.E \sim a.F$
2. $E + G \sim F + G$
3. $E \mid G \sim F \mid G$
4. $E[f] \sim F[f]$
5. $E \backslash K \sim F \backslash K$

Proof of case 3: if $E \sim F$ then $E \mid G \sim F \mid G$

We show $B = \{(E \mid G, F \mid G) \, : \, E \sim F\}$ is a bisimulation.

Proof of case 3: if $E \sim F$ then $E \mid G \sim F \mid G$

We show $B = \{(E \mid G, F \mid G) \,:\, E \sim F\}$ is a bisimulation.
Assume that $((E \mid G), (F \mid G)) \in B$ and $E \mid G \xrightarrow{a} E' \mid G'$

Proof of case 3: if $E \sim F$ then $E \mid G \sim F \mid G$

We show $B = \{(E \mid G, F \mid G) : E \sim F\}$ is a bisimulation.
Assume that $((E \mid G), (F \mid G)) \in B$ and $E \mid G \xrightarrow{a} E' \mid G'$

- $E \xrightarrow{a} E'$ and $G = G'$. Because $E \sim F$, we know that $F \xrightarrow{a} F'$ and $E' \sim F'$ for some F'. Therefore $F \mid G \xrightarrow{a} F' \mid G$, and so $((E' \mid G), (F' \mid G)) \in B$.

Proof of case 3: if $E \sim F$ then $E \mid G \sim F \mid G$

We show $B = \{(E \mid G, F \mid G) : E \sim F\}$ is a bisimulation.
Assume that $((E \mid G), (F \mid G)) \in B$ and $E \mid G \xrightarrow{a} E' \mid G'$

- $E \xrightarrow{a} E'$ and $G = G'$. Because $E \sim F$, we know that $F \xrightarrow{a} F'$ and $E' \sim F'$ for some F'. Therefore $F \mid G \xrightarrow{a} F' \mid G$, and so $((E' \mid G), (F' \mid G)) \in B$.
- $G \xrightarrow{a} G'$ and $E' = E$. So $F \mid G \xrightarrow{a} F \mid G'$, and by definition $((E \mid G'), (F \mid G')) \in B$.

Proof of case 3: if $E \sim F$ then $E \mid G \sim F \mid G$

We show $B = \{(E \mid G, F \mid G) : E \sim F\}$ is a bisimulation.
Assume that $((E \mid G), (F \mid G)) \in B$ and $E \mid G \xrightarrow{a} E' \mid G'$

- $E \xrightarrow{a} E'$ and $G = G'$. Because $E \sim F$, we know that $F \xrightarrow{a} F'$ and $E' \sim F'$ for some F'. Therefore $F \mid G \xrightarrow{a} F' \mid G$, and so $((E' \mid G), (F' \mid G)) \in B$.
- $G \xrightarrow{a} G'$ and $E' = E$. So $F \mid G \xrightarrow{a} F \mid G'$, and by definition $((E \mid G'), (F \mid G')) \in B$.
- $a = \tau$ and $E \xrightarrow{b} E'$ and $G \xrightarrow{\overline{b}} G'$. $F \xrightarrow{b} F'$ for some F' such that $E' \sim F'$, so $F \mid G \xrightarrow{\tau} F' \mid G'$, and therefore $((E' \mid G'), (F' \mid G')) \in B$.

Proof of case 3: if $E \sim F$ then $E \mid G \sim F \mid G$

We show $B = \{(E \mid G, F \mid G) \;:\; E \sim F\}$ is a bisimulation.
Assume that $((E \mid G), (F \mid G)) \in B$ and $E \mid G \xrightarrow{a} E' \mid G'$

- $E \xrightarrow{a} E'$ and $G = G'$. Because $E \sim F$, we know that $F \xrightarrow{a} F'$ and $E' \sim F'$ for some F'. Therefore $F \mid G \xrightarrow{a} F' \mid G$, and so $((E' \mid G), (F' \mid G)) \in B$.
- $G \xrightarrow{a} G'$ and $E' = E$. So $F \mid G \xrightarrow{a} F \mid G'$, and by definition $((E \mid G'), (F \mid G')) \in B$.
- $a = \tau$ and $E \xrightarrow{b} E'$ and $G \xrightarrow{\overline{b}} G'$. $F \xrightarrow{b} F'$ for some F' such that $E' \sim F'$, so $F \mid G \xrightarrow{\tau} F' \mid G'$, and therefore $((E' \mid G'), (F' \mid G')) \in B$.

Symmetrically for a transition $F \mid G \xrightarrow{a} F' \mid G'$.

Showing bisimilarity

To establish $E \sim F$

1. Present a candidate relation R with $(E, F) \in R$
2. Prove that indeed it obeys the hereditary conditions

Showing bisimilarity

To establish $E \sim F$

1. Present a candidate relation R with $(E, F) \in R$
2. Prove that indeed it obeys the hereditary conditions

Example: $(A|B)\backslash c \sim C_1$

$$
\begin{aligned}
A &\stackrel{\text{def}}{=} a.\overline{\mathtt{c}}.A \\
B &\stackrel{\text{def}}{=} c.\overline{\mathtt{b}}.B \\
C_0 &\stackrel{\text{def}}{=} \overline{\mathtt{b}}.C_1 + a.C_2 \\
C_1 &\stackrel{\text{def}}{=} a.C_3 \\
C_2 &\stackrel{\text{def}}{=} \overline{\mathtt{b}}.C_3 \\
C_3 &\stackrel{\text{def}}{=} \tau.C_0
\end{aligned}
$$

Showing bisimilarity

To establish $E \sim F$

1. Present a candidate relation R with $(E, F) \in R$
2. Prove that indeed it obeys the hereditary conditions

Example: $(A|B)\backslash c \sim C_1$

$$
\begin{aligned}
A &\stackrel{\text{def}}{=} a.\overline{\texttt{c}}.A \\
B &\stackrel{\text{def}}{=} c.\overline{\texttt{b}}.B \\
C_0 &\stackrel{\text{def}}{=} \overline{\texttt{b}}.C_1 + a.C_2 \\
C_1 &\stackrel{\text{def}}{=} a.C_3 \\
C_2 &\stackrel{\text{def}}{=} \overline{\texttt{b}}.C_3 \\
C_3 &\stackrel{\text{def}}{=} \tau.C_0
\end{aligned}
$$

R below is a bisimulation

$$
\begin{aligned}
&\{((A|B)\backslash c, C_1), ((\overline{\texttt{c}}.A|B)\backslash c, C_3) \\
&((A|\overline{\texttt{b}}.B)\backslash c, C_0), ((\overline{\texttt{c}}.A|\overline{\texttt{b}}.B)\backslash c, C_2)\}
\end{aligned}
$$

Another example: $\mathtt{Cnt} \sim \mathtt{Ct}'_0$

$$\begin{aligned}
\mathtt{Cnt} &\stackrel{\text{def}}{=} \mathtt{up}.(\mathtt{Cnt} \mid \mathtt{down}.0) \\
\mathtt{Ct}'_0 &\stackrel{\text{def}}{=} \mathtt{up}.\mathtt{Ct}'_1 \\
\mathtt{Ct}'_{i+1} &\stackrel{\text{def}}{=} \mathtt{up}.\mathtt{Ct}'_{i+2} + \mathtt{down}.\mathtt{Ct}'_i \quad i \geq 0.
\end{aligned}$$

Another example: $\texttt{Cnt} \sim \texttt{Ct}'_0$

$$
\begin{aligned}
\texttt{Cnt} &\stackrel{\text{def}}{=} \texttt{up}.(\texttt{Cnt} \mid \texttt{down}.0) \\
\texttt{Ct}'_0 &\stackrel{\text{def}}{=} \texttt{up}.\texttt{Ct}'_1 \\
\texttt{Ct}'_{i+1} &\stackrel{\text{def}}{=} \texttt{up}.\texttt{Ct}'_{i+2} + \texttt{down}.\texttt{Ct}'_i \quad i \geq 0.
\end{aligned}
$$

$$
\begin{aligned}
P_0 &= \{\texttt{Cnt} \mid 0^j \;:\; j \geq 0\} \\
P_{i+1} &= \{E \mid 0^j \mid \texttt{down}.0 \mid 0^k \;:\; E \in P_i \text{ and } j \geq 0 \text{ and } k \geq 0\}
\end{aligned}
$$

where $F \mid 0^0 = F$ and $F \mid 0^{i+1} = F \mid 0^i \mid 0$ and brackets are dropped between parallel components.

Another example: $\texttt{Cnt} \sim \texttt{Ct}'_0$

$$
\begin{aligned}
\texttt{Cnt} &\stackrel{\text{def}}{=} \texttt{up}.(\texttt{Cnt} \mid \texttt{down}.0) \\
\texttt{Ct}'_0 &\stackrel{\text{def}}{=} \texttt{up}.\texttt{Ct}'_1 \\
\texttt{Ct}'_{i+1} &\stackrel{\text{def}}{=} \texttt{up}.\texttt{Ct}'_{i+2} + \texttt{down}.\texttt{Ct}'_i \quad i \geq 0.
\end{aligned}
$$

$$
\begin{aligned}
P_0 &= \{\texttt{Cnt} \mid 0^j \;:\; j \geq 0\} \\
P_{i+1} &= \{E \mid 0^j \mid \texttt{down}.0 \mid 0^k \;:\; E \in P_i \text{ and } j \geq 0 \text{ and } k \geq 0\}
\end{aligned}
$$

where $F \mid 0^0 = F$ and $F \mid 0^{i+1} = F \mid 0^i \mid 0$ and brackets are dropped between parallel components.

$$B = \{(E, \texttt{Ct}'_i) \;:\; i \geq 0 \text{ and } E \in P_i\} \text{ is a bisimulation}$$

Some Results

$$
\begin{array}{lcl}
Id & = & \{(E,E)\} \\
B^{-1} & = & \{(E,F) : (F,E) \in B\} \\
B_1 B_2 & = & \{(E,G) : \text{there is } F.\ (E,F) \in B_1 \\
 & & \text{and } (F,G) \in B_2\}
\end{array}
$$

Proposition Assume B_i $(i = 1, 2, \ldots)$ is a bisimulation. Then the following are bisimulations:

1. Id
2. B_i^{-1}
3. $B_1 B_2$
4. $\bigcup\{B_i : i \geq 1\}$

Corollary $\sim$ is the largest bisimulation

More Properties I

Proposition

1. $E + F \sim F + E$
2. $E + (F + G) \sim (E + F) + G$
3. $E + 0 \sim E$
4. $E + E \sim E$

More Properties I

Proposition

1. $E + F \sim F + E$
2. $E + (F + G) \sim (E + F) + G$
3. $E + 0 \sim E$
4. $E + E \sim E$

Proposition

1. $E|F \sim F|E$
2. $E|(F|G) \sim (E|F)|G$
3. $E|0 \sim E$

More Properties II

Proposition

1. $(E + F)\backslash K \sim E\backslash K + F\backslash K$
2. $(a.E)\backslash K \sim 0$ if $a \in K \cup \overline{K}$
3. $(a.E)\backslash K \sim a.(E\backslash K)$ if $a \notin K \cup \overline{K}$

Expansion law

- Assume $x_i \sim \sum\{a_{ij}.x_{ij} \,:\, 1 \leq j \leq n_i\}$ for $i : 1 \leq i \leq m$

Expansion law

- Assume $x_i \sim \sum\{a_{ij}.x_{ij} \ : \ 1 \leq j \leq n_i\}$ for $i : 1 \leq i \leq m$
- Then $x_1 \mid \ldots \mid x_m \ \sim \mathrm{SUM1} + \mathrm{SUM2}$

Expansion law

- Assume $x_i \sim \sum\{a_{ij}.x_{ij} \ : \ 1 \leq j \leq n_i\}$ for $i : 1 \leq i \leq m$
- Then $x_1 \mid \ldots \mid x_m \ \sim \mathsf{SUM1} + \mathsf{SUM2}$
- SUM1 is $\sum\{a_{ij}.y_{ij} \ : \ 1 \leq i \leq m \text{ and } 1 \leq j \leq n_i\}$

Expansion law

- Assume $x_i \sim \sum\{a_{ij}.x_{ij} \; : \; 1 \leq j \leq n_i\}$ for $i : 1 \leq i \leq m$
- Then $x_1 \mid \ldots \mid x_m \; \sim \mathsf{SUM1} + \mathsf{SUM2}$
- SUM1 is $\sum\{a_{ij}.y_{ij} \; : \; 1 \leq i \leq m \text{ and } 1 \leq j \leq n_i\}$
- SUM2 is $\sum\{\tau.y_{klij} \; : \; 1 \leq k < i \leq m \text{ and } a_{kl} = \bar{a}_{ij}\}$

Expansion law

- Assume $x_i \sim \sum\{a_{ij}.x_{ij} \;:\; 1 \leq j \leq n_i\}$ for $i : 1 \leq i \leq m$
- Then $x_1 \mid \ldots \mid x_m \;\sim \mathsf{SUM1} + \mathsf{SUM2}$
- SUM1 is $\sum\{a_{ij}.y_{ij} \;:\; 1 \leq i \leq m \text{ and } 1 \leq j \leq n_i\}$
- SUM2 is $\sum\{\tau.y_{klij} \;:\; 1 \leq k < i \leq m \text{ and } a_{kl} = \overline{a}_{ij}\}$
- $y_{ij} \;=\; x_1 \mid \ldots \mid x_{i-1} \mid x_{ij} \mid x_{i+1} \mid \ldots \mid x_m$

Expansion law

- Assume $x_i \sim \sum\{a_{ij}.x_{ij} \ : \ 1 \leq j \leq n_i\}$ for $i : 1 \leq i \leq m$
- Then $x_1 \mid \ldots \mid x_m \ \sim \mathsf{SUM1} + \mathsf{SUM2}$
- SUM1 is $\sum\{a_{ij}.y_{ij} \ : \ 1 \leq i \leq m \text{ and } 1 \leq j \leq n_i\}$
- SUM2 is $\sum\{\tau.y_{klij} \ : \ 1 \leq k < i \leq m \text{ and } a_{kl} = \overline{a}_{ij}\}$
- $y_{ij} \ = \ x_1 \mid \ldots \mid x_{i-1} \mid x_{ij} \mid x_{i+1} \mid \ldots \mid x_m$
- $y_{klij} \ = \ x_1 \mid \ldots \mid x_{k-1} \mid x_{kl} \mid x_{k+1} \mid \ldots \mid x_{ij} \mid x_{i+1} \mid \ldots \mid x_m$

Expansion law

- Assume $x_i \sim \sum\{a_{ij}.x_{ij} \;:\; 1 \leq j \leq n_i\}$ for $i : 1 \leq i \leq m$
- Then $x_1 \mid \ldots \mid x_m \;\sim \text{SUM1} + \text{SUM2}$
- SUM1 is $\sum\{a_{ij}.y_{ij} \;:\; 1 \leq i \leq m \text{ and } 1 \leq j \leq n_i\}$
- SUM2 is $\sum\{\tau.y_{klij} \;:\; 1 \leq k < i \leq m \text{ and } a_{kl} = \overline{a}_{ij}\}$
- $y_{ij} \;=\; x_1 \mid \ldots \mid x_{i-1} \mid x_{ij} \mid x_{i+1} \mid \ldots \mid x_m$
- $y_{klij} \;=\; x_1 \mid \ldots \mid x_{k-1} \mid x_{kl} \mid x_{k+1} \mid \ldots \mid x_{ij} \mid x_{i+1} \mid \ldots \mid x_m$
- Example

$$\begin{aligned} x_1 &\sim a.x_{11} + b.x_{12} + a.x_{13} \\ x_2 &\sim \overline{a}.x_{21} + c.x_{22}, \end{aligned}$$

Expansion law

- Assume $x_i \sim \sum\{a_{ij}.x_{ij} \ : \ 1 \leq j \leq n_i\}$ for $i : 1 \leq i \leq m$
- Then $x_1 \mid \ldots \mid x_m \ \sim \mathrm{SUM1} + \mathrm{SUM2}$
- SUM1 is $\sum\{a_{ij}.y_{ij} \ : \ 1 \leq i \leq m \text{ and } 1 \leq j \leq n_i\}$
- SUM2 is $\sum\{\tau.y_{klij} \ : \ 1 \leq k < i \leq m \text{ and } a_{kl} = \overline{a}_{ij}\}$
- $y_{ij} \ = \ x_1 \mid \ldots \mid x_{i-1} \mid x_{ij} \mid x_{i+1} \mid \ldots \mid x_m$
- $y_{klij} \ = \ x_1 \mid \ldots \mid x_{k-1} \mid x_{kl} \mid x_{k+1} \mid \ldots \mid x_{ij} \mid x_{i+1} \mid \ldots \mid x_m$
- Example

$$\begin{aligned} x_1 &\sim a.x_{11} + b.x_{12} + a.x_{13} \\ x_2 &\sim \overline{a}.x_{21} + c.x_{22}, \end{aligned}$$

- $$\begin{aligned} x_1|x_2 \quad \sim \quad & a.(x_{11}|x_2) + b.(x_{12}|x_2) + a.(x_{13}|x_2) + \\ & \overline{a}.(x_1|x_{21}) + \\ & c.(x_1|x_{22}) + \tau.(x_{11}|x_{21}) + \tau.(x_{13}|x_{21}). \end{aligned}$$

Weak (observable) bisimulations

- A binary relation B between processes is a weak (or observable) bisimulation provided that, whenever $(E, F) \in B$ and $a \in \mathsf{O} \cup \{\varepsilon\}$,

Weak (observable) bisimulations

- A binary relation B between processes is a weak (or observable) bisimulation provided that, whenever $(E, F) \in B$ and $a \in \mathsf{O} \cup \{\varepsilon\}$,
- if $E \overset{a}{\Longrightarrow} E'$ then $F \overset{a}{\Longrightarrow} F'$ for some F' such that $(E', F') \in B$ and

Weak (observable) bisimulations

- A binary relation B between processes is a weak (or observable) bisimulation provided that, whenever $(E, F) \in B$ and $a \in \mathrm{O} \cup \{\varepsilon\}$,
- if $E \stackrel{a}{\Longrightarrow} E'$ then $F \stackrel{a}{\Longrightarrow} F'$ for some F' such that $(E', F') \in B$ and
- if $F \stackrel{a}{\Longrightarrow} F'$ then $E \stackrel{a}{\Longrightarrow} E'$ for some E' such that $(E', F') \in B$

Weak (observable) bisimulations

- A binary relation B between processes is a weak (or observable) bisimulation provided that, whenever $(E, F) \in B$ and $a \in \mathrm{O} \cup \{\varepsilon\}$,
- if $E \stackrel{a}{\Longrightarrow} E'$ then $F \stackrel{a}{\Longrightarrow} F'$ for some F' such that $(E', F') \in B$ and
- if $F \stackrel{a}{\Longrightarrow} F'$ then $E \stackrel{a}{\Longrightarrow} E'$ for some E' such that $(E', F') \in B$
- Two processes E and F are weak bisimulation equivalent (or weakly bisimilar) if there is a weak bisimulation relation B such that $(E, F) \in B$. We write $E \approx F$ if E and F are weakly bisimilar

Properties of weak bisimilarity

- Weak bisimilarity is an equivalence relation

Properties of weak bisimilarity

- Weak bisimilarity is an equivalence relation
- Weak bisimilarity is a congruence with respect to all operators of CCS with the exception of $+$

$$\tau.\mathrm{a}.0 \approx \mathrm{a}.0 \quad \text{but} \quad \tau.\mathrm{a}.0 + \mathrm{b}.0 \not\approx \mathrm{a}.0 + \mathrm{b}.0$$

Showing weak bisimilarity $\approx$

1. Present a candidate relation R with $(E, F) \in R$

Showing weak bisimilarity $\approx$

1. Present a candidate relation R with $(E, F) \in R$
2. Prove that indeed it obeys the hereditary conditions

Showing weak bisimilarity $\approx$

1. Present a candidate relation R with $(E, F) \in R$
2. Prove that indeed it obeys the hereditary conditions
3. Example

$$
\begin{aligned}
A_0 &\stackrel{\text{def}}{=} a.A_0 + b.A_1 + \tau.A_1 \\
A_1 &\stackrel{\text{def}}{=} a.A_1 + \tau.A_2 \\
A_2 &\stackrel{\text{def}}{=} b.A_0
\end{aligned}
$$

$$
\begin{aligned}
B_1 &\stackrel{\text{def}}{=} a.B_1 + \tau.B_2 \\
B_2 &\stackrel{\text{def}}{=} b.B_1
\end{aligned}
$$

Showing weak bisimilarity $\approx$

1. Present a candidate relation R with $(E, F) \in R$
2. Prove that indeed it obeys the hereditary conditions
3. Example

$$
\begin{aligned}
A_0 &\stackrel{\text{def}}{=} a.A_0 + b.A_1 + \tau.A_1 \\
A_1 &\stackrel{\text{def}}{=} a.A_1 + \tau.A_2 \\
A_2 &\stackrel{\text{def}}{=} b.A_0
\end{aligned}
$$

$$
\begin{aligned}
B_1 &\stackrel{\text{def}}{=} a.B_1 + \tau.B_2 \\
B_2 &\stackrel{\text{def}}{=} b.B_1
\end{aligned}
$$

4. $A_0 \approx B_1$

$$\{(A_0, B_1), (A_1, B_1), (A_2, B_2)\}$$

is a weak bisimulation

Protocol that may lose messages

$$
\begin{aligned}
\texttt{Sender} &\stackrel{\text{def}}{=} \texttt{in}(x).\overline{\texttt{sm}}(x).\texttt{Send1}(x)\\
\texttt{Send1}(x) &\stackrel{\text{def}}{=} \texttt{ms}.\overline{\texttt{sm}}(x).\texttt{Send1}(x) + \texttt{ok}.\texttt{Sender}\\
\texttt{Medium} &\stackrel{\text{def}}{=} \texttt{sm}(y).\texttt{Med1}(y)\\
\texttt{Med1}(y) &\stackrel{\text{def}}{=} \overline{\texttt{mr}}(y).\texttt{Medium} + \tau.\overline{\texttt{ms}}.\texttt{Medium}\\
\texttt{Receiver} &\stackrel{\text{def}}{=} \texttt{mr}(x).\overline{\texttt{out}}(x).\overline{\texttt{ok}}.\texttt{Receiver}
\end{aligned}
$$

$$
\texttt{Protocol} \equiv (\texttt{Sender} \mid \texttt{Medium} \mid \texttt{Receiver})\backslash\{\texttt{sm}, \texttt{ms}, \texttt{mr}, \texttt{ok}\}
$$

$$
\texttt{Cop} \stackrel{\text{def}}{=} \texttt{in}(x).\overline{\texttt{out}}(x).\texttt{Cop}
$$

$\texttt{Protocol} \approx \texttt{Cop}$

Let B be the following relation

$$
\begin{aligned}
&\{(\texttt{Protocol}, \texttt{Cop})\} \cup \\
&\{((\texttt{Send1}(m) \mid \texttt{Medium} \mid \overline{\texttt{ok}}.\texttt{Receiver})\backslash J, \\
&\qquad \texttt{Cop}) \; : \; m \in D\} \cup \\
&\{((\overline{\texttt{sm}}(m).\texttt{Send1}(m) \mid \texttt{Medium} \mid \texttt{Receiver})\backslash J, \\
&\qquad \overline{\texttt{out}}(m).\texttt{Cop}) \; : \; m \in D\} \cup \\
&\{((\texttt{Send1}(m) \mid \texttt{Med1}(m) \mid \texttt{Receiver})\backslash J, \\
&\qquad \overline{\texttt{out}}(m).\texttt{Cop}) \; : \; m \in D\} \cup \\
&\{((\texttt{Send1}(m) \mid \texttt{Medium} \mid \overline{\texttt{out}}(m).\overline{\texttt{ok}}.\texttt{Receiver})\backslash J, \\
&\qquad \overline{\texttt{out}}(m).\texttt{Cop}) \; : \; m \in D\} \cup \\
&\{((\texttt{Send1}(m) \mid \overline{\texttt{ms}}.\texttt{Medium} \mid \texttt{Receiver})\backslash J, \\
&\qquad \overline{\texttt{out}}(m).\texttt{Cop}) \; : \; m \in D\}
\end{aligned}
$$

B is a weak bisimulation