Sapienza Università di Roma
Dipartimento di Informatica

CONCURRENT SYSTEMS
LECTURE 14

Prof. Daniele Gorla

An inference system for checking bisimilarity

Sapienza
Università di Roma
Dipartimento di Informatica

We shall only consider finite processes (processes without recursive definitions)

- A limited handling of recursion is possible
- Deciding bisimilarity for general processes is undecidable

Inference system = axioms + inference rules

- Soundness: whatever I infer is correct (i.e., bisimiar)
- Completeness: whatever is bisimilar, it can be inferred

Axioms & Rules for Strong Bisimilarity

Axioms for Sum:

$$\vdash M + \mathbf{0} = M$$
$$\vdash M_1 + M_2 = M_2 + M_1$$
$$\vdash M_1 + (M_2 + M_3) = (M_1 + M_2) + M_3$$
$$\vdash M + M = M$$

Axioms for Restriction:

$$\vdash \mathbf{0}\backslash a = \mathbf{0}$$
$$\vdash (\sum_i \alpha_i.P_i)\backslash a = \sum_i (\alpha_i.P_i)\backslash a$$
$$\vdash (\alpha.P)\backslash a = \begin{cases} \mathbf{0} & \text{if } \alpha \in \{a, \bar{a}\} \\ \alpha.(P\backslash a) & \text{otherwise} \end{cases}$$

Axiom for Parallel:

$$\vdash \sum_i \alpha_i.P_i \mid \sum_j \beta_j.Q_j \quad = \quad \sum_i \alpha_i(P_i \mid \sum_j \beta_j.Q_j) \ + \ \sum_j \beta_j(\sum_i \alpha_i.P_i \mid Q_j) \ + \ \sum_{\alpha_i = \overline{\beta_j}} \tau(P_i \mid Q_j)$$

Inference Rules:

$$\vdash P = P \qquad \frac{\vdash P = Q}{\vdash Q = P}$$

$$\frac{\vdash P = Q \quad \vdash Q = R}{\vdash P = R}$$

$$\frac{\vdash P = Q}{\vdash C[P] = C[Q]}$$

Theorem (Soundness): If $\vdash P = Q$ then $P \sim Q$.

Proof.

- for every axiom $\vdash LHS = RHS$, let us consider the relation $\{(LHS, RHS)\} \cup Id$ and prove that it is a bisimulation;
- the inference rules hold for bisimilarity, since it is an equivalence and a congruence. $\square$

P is in standard form *if and only if* $P \triangleq \sum_i \alpha_i.P_i$ *and* $\forall_i$ P_i *is in standard form.*

Lemma 5.2. *$\forall P \; \exists \, P'$ in standard form such that $\vdash P = P'$*

Proof. By induction on the structure of P.

Base case:($P \triangleq \mathbf{0}$). It suffices to consider $P' \triangleq \mathbf{0}$ and conclude by reflexivity.

Inductive step: We have to consider three cases.

2. $P \triangleq P_1|P_2$. By induction, we have that $\exists \; P_1', P_2'$ in standard form such that $\vdash P_1 = P_1'$ and $\vdash P_2 = P_2'$, where $P_1' = \sum_i \alpha_i.R_i$ and $P_2' = \sum_j \beta_j.Q_j$.

   From these facts, by context closure, it follows that $\vdash P_1|P_2 = P_1'|P_2$ and $\vdash P_1'|P_2 = P_1'|P_2'$; hence, by transitivity:

$$
\begin{array}{rcl}
\vdash \; \overbrace{P_1|P_2}^{P} & = & \sum_i \alpha_i.R_i \mid \sum_j \beta_j.Q_j \\
& = & \sum_i \alpha_i(R_i \mid \sum_j \beta_j.Q_j) + \sum_j \beta_j(\sum_i \alpha_i.R_i \mid Q_j) + \sum_{\alpha_i = \overline{\beta_j}} \tau(R_i \mid Q_j) \\
& = & \dots \\
& = & P'
\end{array}
$$

   where the elimination of the parallel from standard forms is repeated until there are no more occurrences of '|' in the process.

1. $P \triangleq \sum_{i \in I} \alpha_i.P_i$. By induction, we have that $\forall P_i \ \exists\, P_i'$ in standard form such that $\vdash P_i = P_i'$.
   From $\vdash P_1 = P_1'$, by context closure w.r.t. context $\alpha_1.\Box + \sum_{i \in I \setminus \{1\}} \alpha_i.P_i$, we have that

$$\vdash \alpha_1.P_1 + \sum_{i \in I \setminus \{1\}} \alpha_i.P_i = \alpha_1.P_1' + \sum_{i \in I \setminus \{1\}} \alpha_i.P_i$$

   From $\vdash P_2 = P_2'$, by context closure w.r.t. context $\alpha_2.\Box + (\alpha_1.P_1' + \sum_{i \in I \setminus \{1,2\}} \alpha_i.P_i)$, we have that

$$\vdash \alpha_2.P_2 + (\alpha_1.P_1' + \sum_{i \in I \setminus \{1,2\}} \alpha_i.P_i) = \alpha_2.P_2' + (\alpha_1.P_1' + \sum_{i \in I \setminus \{1,2\}} \alpha_i.P_i)$$

   By transitivity and commutativity of choices, we have that

$$\vdash \sum_{i \in I} \alpha_i.P_i = \sum_{i \in \{1,2\}} \alpha_i.P_i' + \sum_{i \in I \setminus \{1,2\}} \alpha_i.P_i$$

From $\vdash P_3 = P_3'$, by context closure w.r.t. context $\alpha_3.\Box+(\sum_{i\in\{1,2\}}\alpha_i.P_i' + \sum_{i\in I\setminus\{1,2,3\}}\alpha_i.P_i)$, we have that

$$\vdash \alpha_3.P_3+(\sum_{i\in\{1,2\}}\alpha_i.P_i' + \sum_{i\in I\setminus\{1,2,3\}}\alpha_i.P_i) = \alpha_3.P_3'+(\sum_{i\in\{1,2\}}\alpha_i.P_i' + \sum_{i\in I\setminus\{1,2,3\}}\alpha_i.P_i)$$

and

$$\vdash \sum_{i\in I}\alpha_i.P_i = \sum_{i\in\{1,2,3\}}\alpha_i.P_i' + \sum_{i\in I\setminus\{1,2,3\}}\alpha_i.P_i$$

We can repeat this reasoning until we obtain

$$\vdash \underbrace{\sum_{i\in I}\alpha_i.P_i}_{P} = \underbrace{\sum_{i\in I}\alpha_i.P_i'}_{P'}$$

3. $P \triangleq Q\backslash a$. By induction, we have that $\exists\, Q'$ in standard form such that $\vdash Q = Q'$, where $Q' = \sum_{i \in I} \alpha_i.R_i$. From this and by congruence, it follows that

$$
\begin{aligned}
\vdash \overbrace{Q\backslash a}^{P} &= Q'\backslash a \\
&= \sum_{i \in I} (\alpha_i.R_i)\backslash a \\
&= \sum_{i \in I'} \alpha_i(R_i\backslash a)
\end{aligned}
$$

where $I' \triangleq \{i \in I : \alpha_i \notin \{a, \bar{a}\}\}$ and the elimination of restriction is repeated until such an operator is totally removed from the process. $\square$

Theorem (Completeness): If $P \sim Q$ then $\vdash P = Q$.

Proof. Because of the previous Lemma, we have that $\exists\, P', Q'$ in standard form such that $\vdash Q = Q'$ and $\vdash P = P'$, where

$$P' \triangleq \sum_{i=1}^{n} \alpha_i.P_i \quad \text{and} \quad Q' \triangleq \sum_{j=1}^{m} \beta_j.Q_j$$

We only have to prove that $\vdash P' = Q'$ and, by transitivity, we would obtain $\vdash P = Q$. This proof is done by induction over the maximum height of the syntactic tree that describes P' and Q', i.e. over $max\{h(P'), h(Q')\}$.

Base case (0): in this case, $P' = Q' = \mathbf{0}$ and we trivially conclude.

Induction:

$$\begin{aligned} P' \xrightarrow{\alpha_1} P_1 \quad & \Rightarrow \quad P \xrightarrow{\alpha_1} \hat{P} \ \text{ s.t. } \ P_1 \sim \hat{P} \\ & \Rightarrow \quad Q \xrightarrow{\alpha_1} \hat{Q} \ \text{ s.t. } \ \hat{P} \sim \hat{Q} \\ & \Rightarrow \quad Q' \xrightarrow{\alpha_1} Q'' \ \text{ s.t. } \ \hat{Q} \sim Q'' \end{aligned}$$

by definition of Q', it must be that $\alpha_1 = \beta_{j_1}$ and $Q'' = Q_{j_1}$, for some j_1.

By transitivity, we obtain that $P_1 \sim Q_{j_1}$; hence, by induction, it follows that $\vdash P_1 = Q_{j_1}$.

Let us now consider the context

$$C \triangleq \alpha_1[] + \sum_{i=2}^{n} \alpha_i.P_i$$

Then

$$\vdash \quad \overbrace{\alpha_1.P_1 + \sum_{i=2}^{n} \alpha_i.P_i}^{P'} \quad = \beta_{j_1}.Q_{j_1} + \sum_{i=2}^{n} \alpha_i.P_i$$

By iterating this reasoning on every summand of P', we can conclude that

$$\begin{aligned}
\vdash P' \quad &= \beta_{j_1}.Q_{j_1} + \sum_{i=2}^{n} \alpha_i.P_i \\
&= \beta_{j_1}.Q_{j_1} + \beta_{j_2}.Q_{j_2} + \sum_{i=3}^{n} \alpha_i.P_i \\
&= \beta_{j_1}.Q_{j_1} + \beta_{j_2}.Q_{j_2} + \beta_{j_3}.Q_{j_3} + \sum_{i=4}^{n} \alpha_i.P_i \\
&= \ldots \\
&= \sum_{i=1}^{n} \beta_{j_i}.Q_{j_i}
\end{aligned}$$

Similarly, we can prove that

$$\vdash Q' = \sum_{j=1}^{m} \alpha_{i_j}.P_{i_j}$$

If we now sum these equalities member-wise, we obtain

$$\vdash P' + \sum_{j=1}^{m} \alpha_{i_j}.P_{i_j} = Q' + \sum_{i=1}^{n} \beta_{j_i}.Q_{j_i}$$

that, by idempotency, implies $\vdash P' = Q'$. $\square$

Axioms & Rules for Weak Bisimilarity

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Axioms for Sum:

$$\vdash M + \mathbf{0} = M$$
$$\vdash M_1 + M_2 = M_2 + M_1$$
$$\vdash M_1 + (M_2 + M_3) = (M_1 + M_2) + M_3$$
$$\vdash M + M = M$$

Axioms for Restriction:

$$\vdash \mathbf{0}\backslash a = \mathbf{0}$$
$$\vdash (\sum_i \alpha_i.P_i)\backslash a = \sum_i (\alpha_i.P_i)\backslash a$$
$$\vdash (\alpha.P)\backslash a = \begin{cases} \mathbf{0} & \text{if } \alpha \in \{a, \bar{a}\} \\ \alpha.(P\backslash a) & \text{otherwise} \end{cases}$$

Axiom for Parallel:

$$\begin{aligned} \vdash \sum_i \alpha_i.P_i \mid \sum_j \beta_j.Q_j \quad = \quad & \sum_i \alpha_i(P_i \mid \sum_j \beta_j.Q_j) \quad + \\ & \sum_j \beta_j(\sum_i \alpha_i.P_i \mid Q_j) \quad + \\ & \sum_{\alpha_i = \overline{\beta_j}} \tau(P_i \mid Q_j) \end{aligned}$$

Axioms for τ:

$$\vdash \alpha.P = \alpha.\tau.P$$
$$\vdash P + \tau.P = P$$
$$\vdash \alpha.(P + \tau.Q) = \alpha.(P + \tau.Q) + \alpha.Q$$

Inference Rules:

$$\vdash P = P \qquad \frac{\vdash P = Q}{\vdash Q = P}$$

$$\frac{\vdash P = Q \quad \vdash Q = R}{\vdash P = R}$$

$$\frac{\vdash P = Q}{\vdash C[P] = C[Q]}$$

Example

Sapienza Università di Roma
Dipartimento di Informatica

A server for exchanging messages, in its minimal version, receives a request for sending messages and delivers the confirmation of the reception

Specification:

$$Spec \triangleq send \,.\, \overline{rcv}$$

The behavior of such a server can be implemented by three processes in parallel:

- One handles the button send for sending;
- another one effectively sends the message (through the restricted action *put*) and waits fort the signal of message reception (through the restricted action *go*);
- the last one gives back to the user the outcome of the sending.

$$\left.\begin{array}{lcl} S & \triangleq & send \,.\, \overline{put} \\ M & \triangleq & put \,.\, \overline{go} \\ R & \triangleq & go \,.\, \overline{rcv} \end{array}\right\} Impl \triangleq (S|M|R)\backslash\{put, go\}$$

We now want to prove that the specification is equivalent (i.e., weekly bisimilar) to the implementation.

Let us consider the parallel of processes M and R; by using the axiom for parallel, we have

$$\vdash M|R = put.(\overline{go}|R) + go.(M|\overline{rcv})$$

By using the same axiom to the parallel of the three processes, we obtain

$$\vdash S|(M|R) = send.(\overline{put}|(M|R)) + put.(\overline{go}|R|S) + go.(\overline{rcv}|S|M)$$

By restricting put and go, and by using the second axiom for restriction, we have that

$$\begin{array}{rcll} \vdash Impl & = & (send.(\overline{put}|M|R))\backslash\{put, go\} & + \\ & & (put.(\overline{go}|R|S))\backslash\{put, go\} & + \\ & & (go.(\overline{rcv}|S|M))\backslash\{put, go\} & \end{array}$$

We now apply the third axiom for restriction to the three summands:

- $(send.(\overline{put}|M|R))\backslash\{put, go\} = send.(\overline{put}|(M|R))\backslash\{put, go\}$, since $send \notin \{put, \overline{put}, go, \overline{go}\}$;
- $(put.(\overline{go}|R|S))\backslash\{put, go\} = \mathbf{0}$;
- $(go.(\overline{rcv}|S|M))\backslash\{put, go\} = \mathbf{0}$.

Hence, $\vdash Impl = send.(\overline{put}|(M|R))\backslash\{put, go\}$.

We now work in a similar way on $(\overline{put}|M|R)\backslash\{put, go\}$

$$\vdash M|R = put.(\overline{go}|R) + go.(M|\overline{rcv})$$

$$\vdash (\overline{put}|M|R)\backslash\{put, go\} = \tau.(\overline{go}|R)\backslash\{put, go\}$$

$$\vdash Impl = send.\tau.(\overline{go}|R)\backslash\{put, go\}$$

By using the first axiom for weak bisimilarity, we obtain

$$\vdash Impl = send.(\overline{go}|R)\backslash\{put, go\}$$

Again, the processes synchronize, now on name *go*:

$$\vdash Impl = send.\tau.(\overline{rcv})\backslash\{put, go\}$$

As before, this leads to

$$\vdash Impl = send.(\overline{rcv}.\mathbf{0})\backslash\{put, go\}$$

We now simply use the third axiom for restriction and obtain

$$\vdash Impl = send.\overline{rcv}.\mathbf{0}\backslash\{put, go\}$$

Finally, by the first axiom for restriction, we have that

$$\vdash Impl = send.\overline{rcv}.\mathbf{0}$$