A Logical Characterization of Efficiency Preorders

Neelesh Korade
neelesh_korade@persistent.co.in
Persistent Systems Private Limited
Pune, India 411 016.
and
S. Arun-Kumar
sak@cse.iitd.ernet.in
Department of Computer Science and Engineering
I. I. T. Delhi, Hauz Khas, New Delhi 110 016.
http://www.cse.iitd.ernet.in/∼sak

September 23, 2004

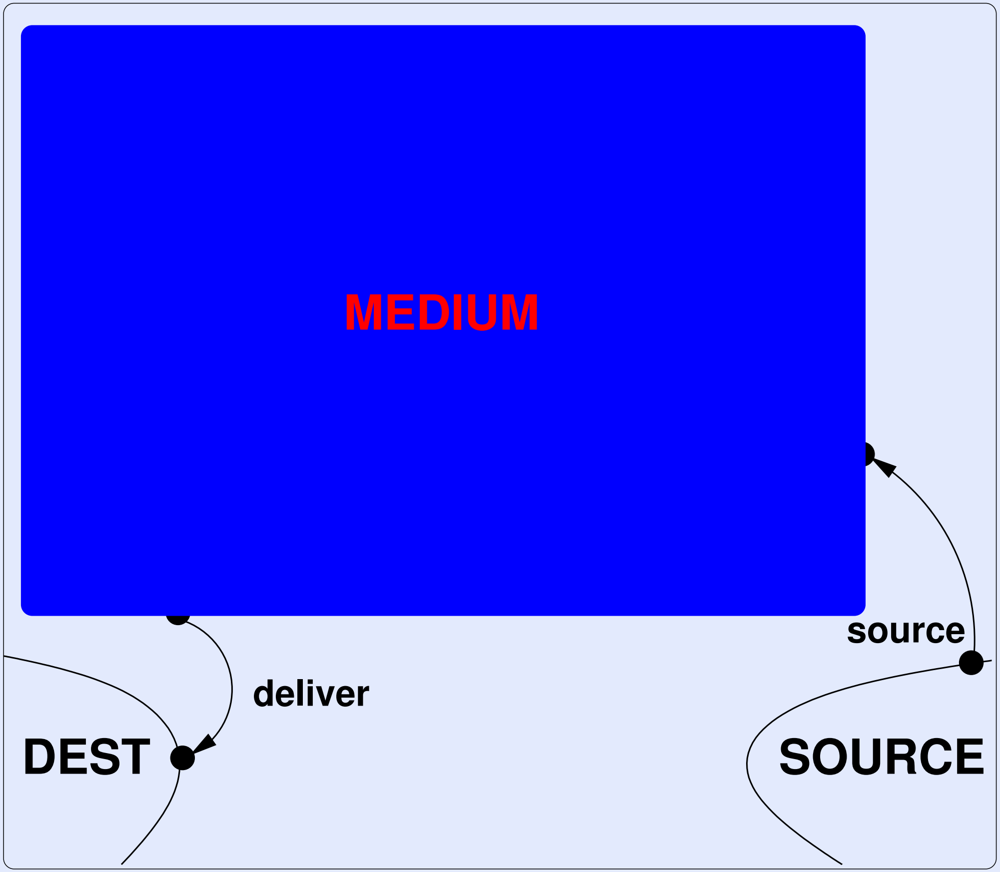
Outline

- Introduction
- Efficiency-based Preorders
- Simulations & HML^{-}
- Principle of Containment
- Strong Bisimulations and HML
- Derived Operators & Expressiveness
- Weak Bisimulations & $OHML$
- Elaboration
- Efficiency Prebisimulations
- Conclusion

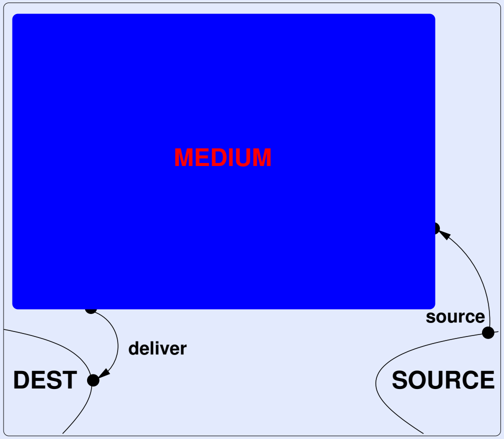
Colour Coding

- Behavioural relations.
- Logic related symbols and terms.
- Hyperlinks or references
- Headings of slides and general highlighting
- anything that needs emphasis
- Terms being Defined
- Everything else.

Example: 3buffer Medium

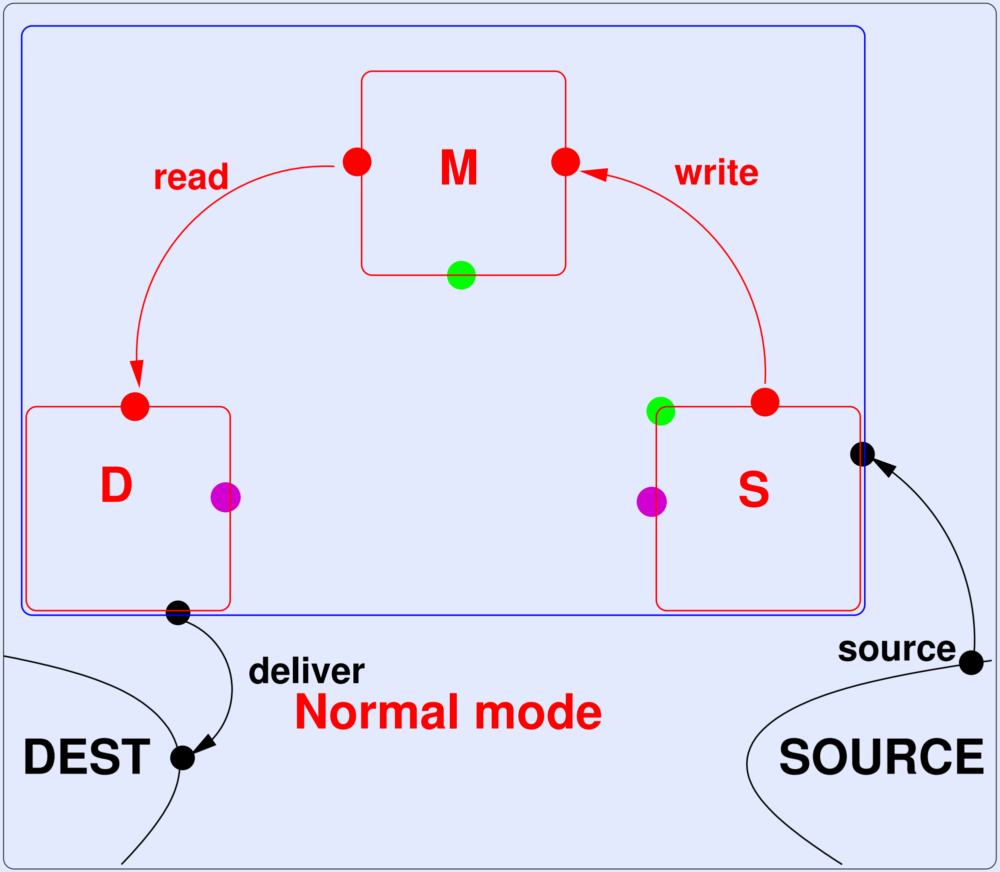


Example: 3buffer Medium SPEC

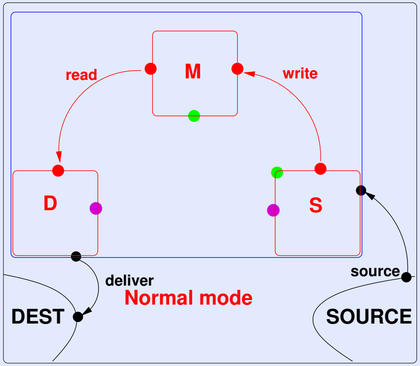


$$
\begin{aligned}
MEDIUM(\varepsilon) &= source(d).MEDIUM(d)\\
MEDIUM(d) &= source(m).MEDIUM(dm)+\\
&\quad \overline{deliver(d)}.MEDIUM(\varepsilon)\\
MEDIUM(dm) &= source(s).MEDIUM(dms)+\\
&\quad \overline{deliver(d)}.MEDIUM(m)\\
MEDIUM(dms) &= \overline{deliver(d)}.MEDIUM(ms)
\end{aligned}
$$

Example: 3buffer Medium

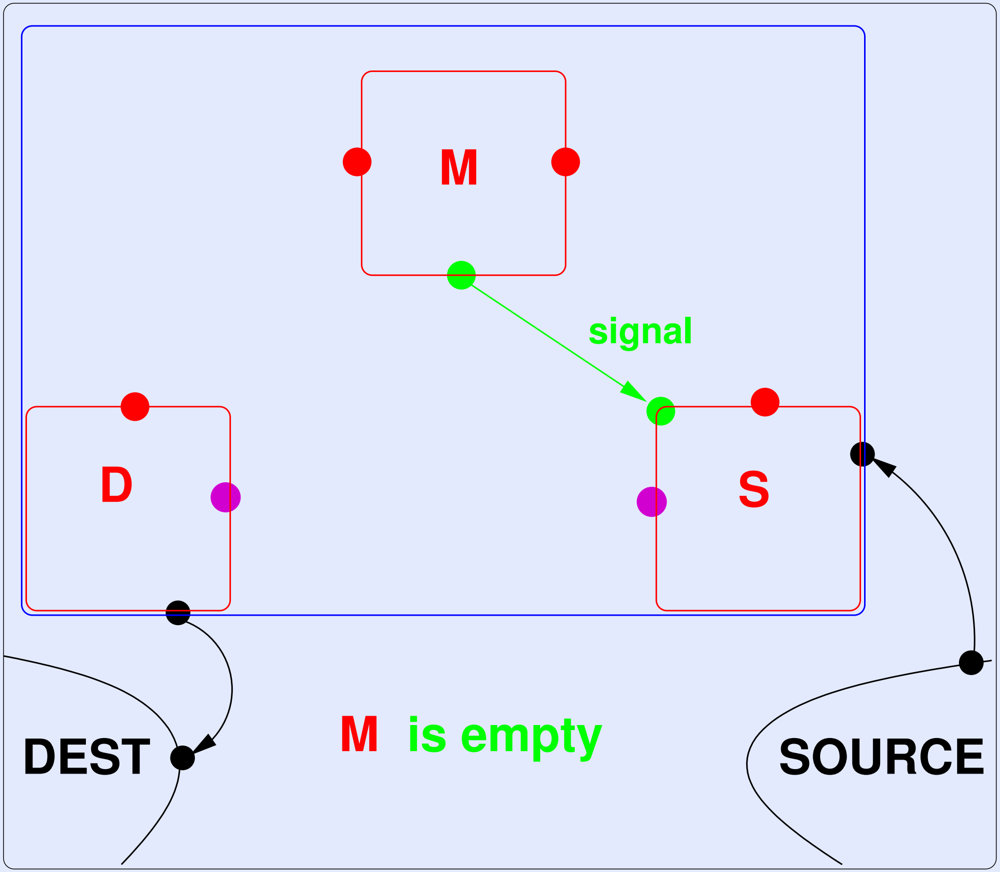


Example: 3buffer Medium N

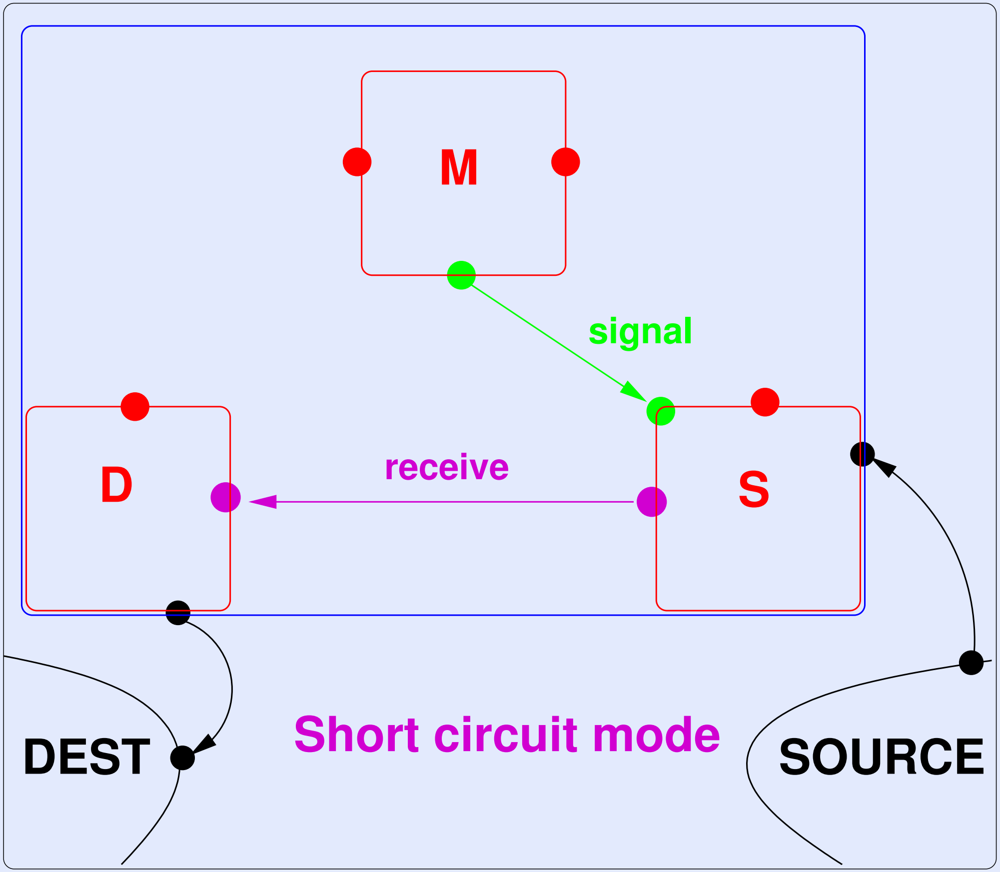


$$
\begin{aligned}
C(\bot) &= in(u).C(u) \\
C(u) &= \overline{u}.C(\bot) \\
D_1(u_\bot) &= C[read/in, deliver/out] \\
M_1(u_\bot) &= C[read/out, write/in] \\
S_1(u_\bot) &= C[source/in, write/out] \\
N(d,m,s) &= (D_1(d) \mid M_1(m) \mid S_1(s)) \backslash \{read, write\}
\end{aligned}
$$

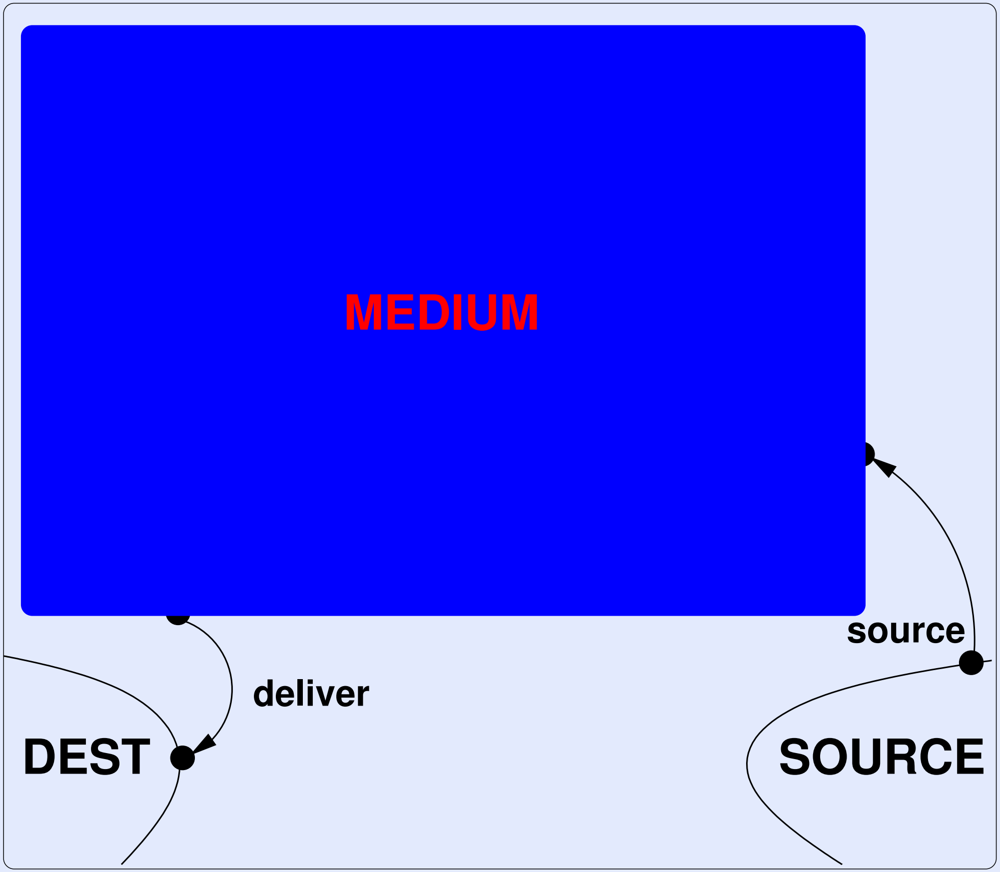
Example: 3buffer Medium



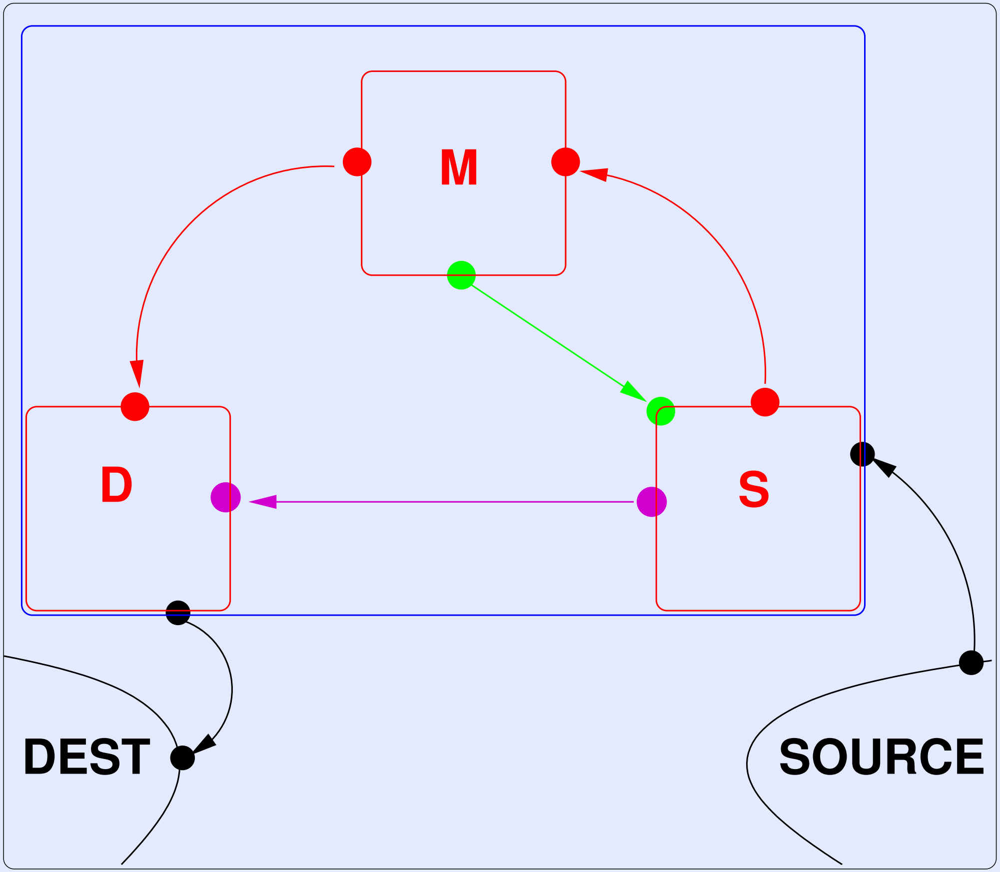
Example: 3buffer Medium



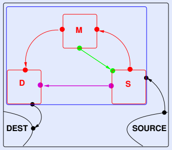
Example: 3buffer Medium



Example: 3buffer Medium



Example: 3buffer Medium SC



$$
\begin{array}{lcl}
D_2(\perp) & = & read(u).D_2(u) + receive(v).D(v) \\
D_2(w) & = & \overline{deliver(w)}.D(\perp) \\
M_2(\perp) & = & write(u).M_2(u) \\
M_2(u) & = & \overline{read(u)}.\overline{mempty()}.M(\perp) \\
S_2(\perp, 0) & = & source(u).S_2(u, 0) + mempty().S_2(\perp, 1) \\
S_2(u, 0) & = & mempty().S_2(u, 1) \\
S_2(u, 1) & = & \overline{receive(u)}.S_2(\perp, 1) + \overline{write(u)}.S_2(\perp, 0) \\
S_2(\perp, 1) & = & source(u).S_2(u, 1) \\
Internals & = & \{read, write, receive, mempty\} \\
SC(d, m, s, 1) & = & (D_2(d) \mid M_2(m) \mid S_2(s, 1)) \backslash Internals
\end{array}
$$

Elaborations

- $R \subseteq \mathbb{P} \times \mathbb{P}$ is a weak bisimulation iff for every $\langle p, q \rangle \in R$ and $a \in Act$,

$$p \xrightarrow{a} p' \Rightarrow \exists q' : q \stackrel{\hat{a}}{\Longrightarrow} q' \wedge p'Rq' \quad and$$
$$q \xrightarrow{a} q' \Rightarrow \exists p' : p \stackrel{\hat{a}}{\Longrightarrow} p' \wedge p'Rq'$$

- $R \subseteq \mathbb{P} \times \mathbb{P}$ is an elaboration[2] if for every $\langle p, q \rangle \in R$ and $a \in Act$,

$$\boxed{p \xrightarrow{a} p' \Rightarrow \exists q' : q \stackrel{\hat{a}}{\Longrightarrow} q' \wedge p'Rq'}$$
$$\boxed{q \xrightarrow{a} q' \Rightarrow \exists p' : p \stackrel{a}{\Longrightarrow} p' \wedge p'Rq'}$$

- $p \lessapprox q$ if $\langle p, q \rangle$ belongs to an elaboration
- $\lessapprox$ is a preorder close to $\approx$ but *finer*.
- $$\begin{array}{ccc} \tau.p \approx p & \text{and of course,} & p \approx \tau.p \\ \tau.p \lessapprox p & \text{but} & p \not\lessapprox \tau.p \end{array}$$

3buffer

Efficiency Prebisimulations[1]

- $R \subseteq P \times P$ is an efficiency prebisimulation (EP for short) if for every $\langle p, q \rangle \in R$ the following conditions are satisfied.

$$p \xrightarrow{\alpha} p' \Rightarrow \exists q' : q \xrightarrow{\alpha} q' \wedge p'Rq'$$

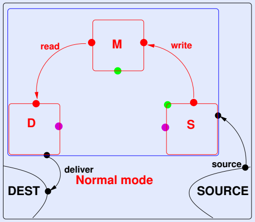
$$p \xrightarrow{\tau} p' \Rightarrow p'Rq' \vee \exists q' : q \xrightarrow{\tau} q' \wedge p'Rq'$$

$$q \xrightarrow{a} q' \Rightarrow \exists p' : p \stackrel{a}{\Longrightarrow} p' \wedge p'Rq'$$

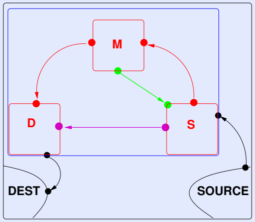
- $p \lesssim q$ if $\langle p, q \rangle$ belongs to an EP.
- $\lesssim$ is a preorder lying between $\sim$ and $\lessapprox$.

Home Page
ICTAC-2004
Page 14 of 37
Go Back
Full Screen
Close
Quit

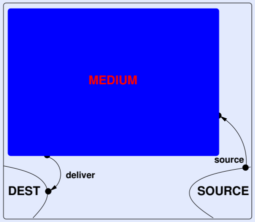
Example: 3buffer Comparisons



$\lesssim$



$\lesssim$



$$\begin{array}{lcl} N(dms) & \approx & SC(dms?) \\ SC(dms?) & \approx & MEDIUM(dms) \\ N(\bot\bot\bot) & \lesssim & SC(\bot\bot\bot\ 1) \\ SC(\bot\bot\bot\ 1) & \lesssim & MEDIUM(\varepsilon) \end{array}$$

$$N(\bot\bot\bot) \lesssim MEDIUM(\varepsilon)$$

- All of them are weakly bisimilar to each other,
- But SC can *sometimes* be "quicker" than N.
- $MEDIUM$ would be the "most efficient" in *every* run! *Pity it is not available as a basic process!*

LTS

- V : countable set of visible actions, $\alpha \in V$
- $\tau \notin V$: a distinguished invisible action
- $Act = V \cup \{\tau\}$: the set of *actions*, $a \in Act$
- $\langle \mathbb{P}, Act, \longrightarrow \rangle$: A labelled transition system (LTS) where
  - $\mathbb{P}$: a set of process states or processes
  - $\longrightarrow \subseteq \mathbb{P} \times Act \times \mathbb{P}$ is the transition relation.
  - $p \xrightarrow{a} q$: denotes $(p, a, q) \in \longrightarrow$ and
  - q is a strong a-derivative of p.

Strong Simulations and HML^{-}

- $R \subseteq \mathbb{P} \times \mathbb{P}$ is a strong simulation (SS) if for every $\langle p, q \rangle \in R, a \in Act$,

$$p \stackrel{a}{\longrightarrow} p' \Rightarrow \exists q' : q \stackrel{a}{\longrightarrow} q' \wedge p'Rq'$$

- $p \leq_{SS} q$ iff pRq for some strong simulation R.
- $\leq_{SS}$ is a preorder (reflexive and transitive).

The class L_{SS} of strong simulation formulae over Act:

$$\varphi ::= \ <a> \varphi \mid \bigwedge_{i \in I} \varphi_i$$

Note:

- index set I may be infinite.
- tt is the empty conjunction.

L_{SS}: Semantics

The satisfaction relation $\models \;\subseteq\; \mathbb{P} \times L_{SS}$

- $p \models tt$ for all $p \in \mathbb{P}$.
- $p \models \,<\!a\!>\, \varphi$ for $a \in Act$ if $\exists p' \in \mathbb{P} : p \xrightarrow{a} p'$ and $p' \models \varphi$.
- $p \models \bigwedge_{i \in I} \varphi_i$ if $p \models \varphi_i$ for all $i \in I$.

Alternatively the meaning of a formula is the set of processes that satisfy it:

$$
\begin{aligned}
[\![tt]\!] &= \mathbb{P} \\
[\![<\!a\!>\, \varphi]\!] &= \{p \mid \exists p' : p \xrightarrow{a} p', p' \in [\![\varphi]\!]\} \\
[\![\bigwedge_{i \in I} \varphi_i]\!] &= \bigcap_{i \in I} [\![\varphi_i]\!]
\end{aligned}
$$

L_{SS}: Characterization

Proposition 1

$$p \models \varphi \iff p \in [\![\varphi]\!]$$

□

- Consider the set of all formulae a process satisfies,

$$SS(p) = \{\varphi \in L_{SS} \mid p \models \varphi\}$$

- and define an ordering $\sqsubseteq_{SS}$ (induced by $SS(p)$) on processes:

$$p \sqsubseteq_{SS} q \text{ iff } SS(p) \subseteq SS(q)$$

- whose kernel $=_{SS}$, is

$$p =_{SS} q \text{ iff } SS(p) = SS(q)$$

L_{SS}: Characterization

L_{SS} characterizes the behavioural preorder $\leq_{SS}$.

Theorem 1 *(Hennessy-Milner[3], van Glabeek[4]).*

$$p \leq_{SS} q \iff p \sqsubseteq_{SS} q \iff SS(p) \subseteq SS(q)$$

Proof.

$(\Rightarrow)$. Assume $p \leq_{SS} q$ and $\varphi \in SS(p)$. Proceed by induction on the structure of φ.

$(\Leftarrow)$. Show that $\sqsubseteq_{SS}$ is a strong simulation.

□

Principle of Containment

- Containment. For any behavioural preorder $\leq_B$, a logic L_B characterizes $\leq_B$ if

$$p \leq_B q \text{ iff } B(p) \subseteq B(q)$$

where $B(p)$ is the set of formulae that p satisfies.

- Consequently, the kernel $=_B$, of the preorder $\leq_B$,

$$=_B \; = \; \leq_B \cap \leq_B{}^{-1}$$

is an equivalence relation and is characterized by

$$p =_B q \text{ iff } B(p) = B(q)$$

equality on sets of satisfying formulae.

Expressiveness

- A logic L_1 is as expressive as another L_2 if it can express every property that L_2 can. It is more expressive, denoted $L_1 \prec L_2$, if there are formulae in L_1 that cannot be expressed in L_2.
- Given behavioural preorders $\leq_1$, and $\leq_2$ characterized respectively by logics L_1 and L_2,

$$\leq_1 \ \subset \ \leq_2 \text{ iff } L_1 \prec L_2$$

since L_1 can allow for *finer* distinctions to be made than L_2.

Strong Bisimulations and HML

- $R \subseteq \mathbb{P} \times \mathbb{P}$ is a strong bisimulation (SB) if both R and R^{-1} are strong simulations.
- $p \sim q$ iff pRq for some strong bisimulation R.
- $\sim$ is an equivalence relation.

The class L_{SB} of strong bisimulation formulae over Act:

$$\varphi ::= \ <a>\varphi \mid \bigwedge_{i\in I}\varphi_i \mid \boxed{\neg\varphi}$$

c.f. L_{SS}

L_{SB}: Semantics

- As for L_{SS}. In addition
- $p \models \neg\varphi$ if $p \not\models \varphi$. Alternatively, $[\![\neg\varphi]\!] = \mathbb{P} - [\![\varphi]\!]$.
- $SB(p)$ is the set of all formulae of HML that p satisfies.
- $p \sqsubseteq_{SB} q$ iff $SB(p) \subseteq SB(q)$ and $p =_{SB} q$ iff $SB(p) = SB(q)$.

Proposition 2 *(van Glabeek)[4].* $p \sqsubseteq_{SB} q \iff p =_{SB} q$.

Proof. If $\varphi \in SB(q) - SB(p)$ then $\neg\varphi \in SB(p) - SB(q)$ which contradicts $p \sqsubseteq_{SB} q$. $\square$

L_{SB}: Characterization

Theorem 2 *(Hennessy-Milner[3], van Glabeek[4]).*

$$p \sim q \iff p \sqsubseteq_{SB} q \iff SB(p) \subseteq SB(q)$$

Proof.

($\Rightarrow$). Assume $p \sim q$ and $\varphi \in SB(p)$. Proceed by induction on the structure of φ.

($\Leftarrow$). Suffices to show that $=_{SB}$ is a strong bisimulation. By proposition 2 suffices to show that $\sqsubseteq_{SB}$ is a strong simulation since $=_{SB} = \sqsubseteq_{SB} = {\sqsubseteq_{SB}}^{-1}$.

$\square$

Derived Operators

- Duals:

$$[a]\varphi \equiv \neg <a> \neg\varphi$$
$$\bigvee_{i\in I}\varphi_i \equiv \neg \bigwedge_{i\in I} \neg\varphi_i$$

- Iteration:

$$<a>^0 \varphi \equiv \varphi$$
$$<a>^{m+1} \varphi \equiv <a><a>^m \varphi \quad m \in \mathbb{N}$$

- Weak Operators:

$$\ll\gg\varphi \equiv \bigvee_{m\geq 0} <\tau>^m \varphi$$
$$\ll a \gg\varphi \equiv \ll\gg<a>\ll\gg \varphi$$

- Duals:

$$[[\ ]]\varphi \equiv \neg \ll\gg \neg\varphi$$
$$[[a]]\varphi \equiv [[\ ]][a][[\ ]]\varphi$$

$\ll\hat{a}\gg$ denotes $\ll\gg$ if $a = \tau$ and $\ll a \gg$ otherwise.

Negation-free HML

- Negation may be pushed inward and finally eliminated altogether, provided duals are added to the language.
- The following negation-free logic L'_{SB} is *as expressive* as L_{SB}.

$$\varphi ::= \; <a> \varphi \mid \bigwedge_{i\in I} \varphi_i \mid \boxed{[a]\varphi} \mid \boxed{\bigvee_{i\in I} \varphi_i}$$

- With ff the empty disjunction, $\models$ is extended in the obvious fashion.
  - $p \models ff$ for no process p,
  - $p \models [a]\varphi$ for $a \in Act$ if $\forall p' \in \mathbb{P} : p \xrightarrow{a} p' \Rightarrow p' \models \varphi$.
  - $p \models \bigvee_{i\in I} \varphi_i$ if $\exists i \in I : p \models \varphi_i$.

Home Page
ICTAC-2004
Page 27 of 37
Go Back
Full Screen
Close
Quit

Weak Bisimulations

- $p \Longrightarrow p$ for all processes p
- $p \xrightarrow{\tau} q$ and $q \Longrightarrow r$ implies $p \Longrightarrow r$
- $p \stackrel{a}{\Longrightarrow} q$ if $p \Longrightarrow \xrightarrow{a} \Longrightarrow q$, and
- $p \stackrel{\hat{a}}{\Longrightarrow} q$ denotes $p \Longrightarrow q$ if $a = \tau$ and $p \stackrel{a}{\Longrightarrow} q$ otherwise.
- $R \subseteq \mathbb{P} \times \mathbb{P}$ is a weak simulation if for every $\langle p, q \rangle \in R$ and $a \in Act$,

$$p \xrightarrow{a} p' \Rightarrow \exists q' : q \stackrel{\hat{a}}{\Longrightarrow} q' \wedge p' R q'$$

  R is a weak bisimulation (WB) if both R and R^{-1} are weak simulations.

- $p \approx q$ if $\langle p, q \rangle$ belongs to a weak bisimulation R
- $\approx$ is an equivalence relation.

Observable HML

- The logic L_{WB} of weak bisimulation formulae over V.

$$\varphi \ ::= \ \boxed{\ll \alpha \gg \varphi} \ \mid \bigwedge_{i \in I} \varphi_i \mid \neg \varphi$$

- $\approx$ is coarser than $\sim$, so it requires a less expressive logic to characterize it.
- The semantics of $\ll \alpha \gg \varphi$ may be derived from HML and $\stackrel{\alpha}{\Longrightarrow}$

$$\boxed{p \models \ll \alpha \gg \varphi \text{ if } \exists p' : p \stackrel{\alpha}{\Longrightarrow} p' \wedge p' \models \varphi}$$

- $WB(p) = \{\varphi \in L_{WB} \mid p \models \varphi\}$.
- $p \sqsubseteq_{WB} q$ iff $WB(p) \subseteq WB(q)$
- $p =_{WB} q$ iff $WB(p) = WB(q)$.

Theorem 3 *L_{WB} characterizes weak bisimilarity i.e. $p \approx q$ iff $p =_{WB} q$.*

The Agenda

We require a gradation of logics

- with the following relationships:

Granularity	$\sim$	$\subset$	$\lesssim$	$\subset$	$\lessapprox$	$\subset$	$\approx$
Expressiveness	L_{SB}	$\prec$	L_{EP}	$\prec$	L_E	$\prec$	L_{WB}

- and satisfying the principle of containment:

$$
\begin{aligned}
p \sim q &\iff L_{SB}(p) = L_{SB}(q) \\
p \lesssim q &\iff L_{EP}(p) \subseteq L_{EP}(q) \\
p \lessapprox q &\iff L_{E}(p) \subseteq L_{E}(q) \\
p \approx q &\iff L_{WB}(p) = L_{WB}(q)
\end{aligned}
$$

Logic for Elaborations

- The class $L_E \supset L_{WB}$ of elaboration formulae over Act is given by the following two-level grammar, where $\alpha \in V$ and $a \in A$.

$$\varphi \ ::= \ \ll \alpha \gg \varphi \mid \bigwedge_{i \in I} \varphi_i \mid \neg\varphi$$

$$\pi \ ::= \ \varphi \mid \ll \hat{a} \gg \pi \mid \boxed{\epsilon^k \pi} \mid [[a]]\pi \mid \bigwedge_{i \in I} \pi_i \mid \bigvee_{i \in I} \pi_i$$

where $k > 0$.

- Semantics:

$$\boxed{p \models \epsilon^k \pi \text{ if } \forall p' \in \mathbb{P} : p \xrightarrow{\tau^j} p' \wedge p' \models \pi \Rightarrow j < k}$$

More perspicuously,

$$\boxed{p \models \epsilon^k \pi \text{ if } \forall p' \in \mathbb{P} : p \xrightarrow{\tau^j} p' \wedge j \geq k \Rightarrow p' \not\models \pi}$$

Expressiveness of L_E: Examples

- $\epsilon^1 tt \equiv p$ *is stable* i.e. p cannot perform a τ action.
- $\bigvee_{k>0} \epsilon^k tt \equiv p$ *converges* i.e. p cannot perform an infinite sequence of τ actions.
- $\epsilon^k ff \equiv p$ *may perform any finite sequence of* τ*'s*. (Caution! Does not necessarily imply that p *diverges*, unless p is also finitely branching).
- $\epsilon^j \pi \Rightarrow \epsilon^k \pi$ for all $0 < j < k$.
- $\bigwedge_{k \geq j} \epsilon^k \pi \iff \epsilon^j \pi, j > 0$.
- p *diverges* is not expressible
- Statements specifying lower bounds on number of consecutive τ actions not expressible.

Logic of EP

- The class L_{EP} of efficiency prebisimulation formulae over Act is given by the following grammar.

$$\varphi ::= \ll\alpha\gg \varphi \mid \bigwedge_{i\in I} \varphi_i \mid \neg\varphi$$

$$\pi ::= \varphi \mid \boxed{<\alpha> \pi} \mid \boxed{(\tau)\pi} \mid \epsilon^k\pi \mid [[a]]\pi \mid \bigwedge_{i\in I} \pi_i \mid \bigvee_{i\in I} \pi_i$$

c.f. L_E

- Semantics:

$$\boxed{p \models (\tau)\pi \text{ if } p \models \pi \vee (\exists p' \in \mathbb{P} : p \xrightarrow{\tau} p' \wedge p' \models \pi)}$$

Expressiveness of L_{EP}

- The use of $<\alpha>$ instead of $\ll\hat{a}\gg$ obviously adds more expressive power in terms of determining that a visible action is immediately possible.
- The possiblity operator (τ) does not allow for exact determination of the number of consecutive τ actions possible.
- Statements such as p *has a* τ^2 *derivative that satisfies* π are not expressible.
- HML formulae such as $\neg <\tau><\tau>\varphi$ are not expressible.

The Characterizations

Lemma 1 *$p \lessapprox q \Rightarrow p =_{E_{WB}} q$ and hence $p \lessapprox q \Rightarrow p \sqsubseteq_{E_{WB}} q$.*

□

Theorem 4 *$p \lessapprox q$ iff $p \sqsubseteq_{E} q$.*

□

Lemma 2 *$p \lesssim q \Rightarrow p =_{EP_{WB}} q$ and hence $p \lesssim q \Rightarrow p \sqsubseteq_{EP_{WB}} q$.*

□

Theorem 5 *$p \lesssim q$ iff $p \sqsubseteq_{EP} q$.*

□

- Nothing special in the proof techniques used here.
- Proof proceeds in two levels. The proof for the π level of the language requires the use of the preceding lemma for the φ.

Conclusion & Open Questions

- *Equivalences* have been done before, but *preorders* rarely.
- The principle of containment generalizes from equivalences to preorders very naturally.
- Can something akin to characteristic formulae of Kim Larsen be done in the setting of preorders?
- Solving equations has been done, but how does one solve preordering equations?

References

[1] S. Arun-Kumar and M.Hennessy. An efficiency preorder for processes. *Acta Informatica, Springer-Verlag*, 1992.

[2] S. Arun-Kumar and V. Natarajan. Conformance: A precongruence close to bisimilarity. In *Proceedings, Structures in Concurrency Theory, Springer Workshops in Computer Science Series*, 1995.

[3] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32:137–161, 1985.

[4] R. J. van Glabeek. The linear time – branching time spectrum I. The semantics of concrete, sequential processes. In *Handbook of Process Algebra*. Elsevier Science B. V., Netherlands., 2001.