Extending Process Algebra with an undefined action

http://www.cse.iitd.ac.in/~sak/

S. Arun-Kumar
Department of Computer Science and Engineering
I. I. T. Delhi, Hauz Khas, New Delhi 110 016.

July 8, 2022

Outline

- Motivation
- Process Model
- Basic Extended Process Algera
- Prebisimilarity
- Logical Characterisation: PHML
- Conclusion

Divergence, Livelock and Deadlock

- $\approx$ [8] and weaker equivalences [4] are insensitive to "τ-cycles".
- "τ-cycles" ($T \Leftarrow \tau.T$) are identified with divergence [11] and has the same solution as $X \Leftarrow X$.
- "τ-cycles" could also be due to "livelock" i.e. infinite "internal chatter". $P \Leftarrow a.P$, $Q \Leftarrow \overline{a}.Q$, $R = (P|Q)\backslash a \sim \tau.R$
- In SCCS [7] the solution of $X \Leftarrow X$ is identified with deadlock.

Theoretical Computer Science 25 (1983) 267–310 267
North-Holland

CALCULI FOR SYNCHRONY AND ASYNCHRONY*

Robin MILNER
Department of Computer Science, Edinburgh University, Edinburgh EH9 3JZ, United Kingdom

Communicated by M. Nivat
Received February 1982
Revised August 1982

The derivation rule for recursion is as follows:

$$\frac{E_i\{\mathrm{fix}\,\tilde{X}\tilde{E}/\tilde{X}\} \xrightarrow{a} E'}{\mathrm{fix}_i\,\tilde{X}\tilde{E} \xrightarrow{a} E'}$$

page 274

(i) $F \equiv \mathrm{fix}\,XX$. The only instances of the rule which can yield an action for F have the form

$$\frac{F \xrightarrow{a} E'}{F \xrightarrow{a} E'}$$

Hence, since every derivation must be inferred by a finite proof, F in this case has no actions. It will turn out to be congruent to **0**.

Our view

Divergence (Ω). The least solution of $X \Leftarrow X$ (modulo $\sim$) should be a totally "undefined" process.

Livelock. The least solution (modulo $\sim$) of $T \Leftarrow \tau.T$ should be a process that can only perform "τ-cycles".

Deadlock ($\mathbf{0}$). A deadlocked process performs no computation unlike a live-locked process which consumes computational cycles and energy.

Hence any strong behavioural relation on processes should ensure that

- divergence, deadlock and livelock are distinguished from each other.
- Ω is the least defined process (modulo $\sim$),
- Deadlock ($\mathbf{0}$) and livelock ($T \Leftarrow \tau.T$) are both well-defined processes and mutually incomparable.

LTS

- $\mathbb{L}[L] = \langle S, L, \longrightarrow \rangle$
- $s \xrightarrow{\ell} \_ = \{t \in S \mid s \xrightarrow{\ell} t\}$ is the set of ℓ-*successors* of s.
- $L(s) = \{\ell \in L \mid \exists t[s \xrightarrow{\ell} t]\}$ is the set of labels from s.
- $Succ(s) = \bigcup_{\ell \in L} s \xrightarrow{\ell} \_ = \{t \mid \exists \ell \in L[s \xrightarrow{\ell} t]\}$ is the set of *successors* of s
- $Targets(\longrightarrow) = \{t \in S \mid \exists s \in S[t \in Succ(s)]\}$.
- $Der(s) = \{s\} \cup \bigcup_{t \in Succ(s)} Der(t)$ is the set of *derivatives* of s.

A **sub-LTS** of $\mathbb{L}[L]$ at a state $s_0 \in S$ is the rooted LTS $\langle Der(s_0), L, \longrightarrow, s_0 \rangle$.
By convention $s \xrightarrow{\epsilon} s$ and for any $x = ay \in L^+$, $s \xrightarrow{x} s'$ if $s \xrightarrow{a} s'' \xrightarrow{y} s'$ for some $s'' \in S$.

Natural Bisimilarity

Definition 0.1: Natural bisimulation

A symmetric binary relation $\mathcal{R} \subseteq S \times T$ between (sub-)LTSs $\mathbb{L}[L] = \langle S, L, \longrightarrow \rangle$ and $\mathbb{M}[L] = \langle T, L, \longrightarrow \rangle$ such that $s\mathcal{R}t$ implies for all labels $\ell \in L$, $s \stackrel{\ell}{\longrightarrow} s' \Rightarrow \exists t' \in T[t \stackrel{\ell}{\longrightarrow} t' \wedge s'\mathcal{R}t']$. Notation: $\mathcal{R} \vdash s \sim t$).

Fact 0.1

Unions, relational converses and (relational) compositions of natural bisimulations are also natural bisimulations. Natural bisimilarity ($\sim$) is the largest natural bisimulation and is an equivalence relation.

Actions and Traces

Definition 0.2: Actions

$A_\perp = A \cup \{\perp\}$ where A is a countable set of (uninterpreted but) *well-defined* actions and $\perp \notin A$ is a special *undefined action* with $\perp < a$ for each $a \in A$.

- Ω can perform only $\perp$.
- *Traces* are words from $A^* \perp^*$.
- $A^* \perp^? := A^* \perp^* / (x\perp\perp = x\perp)$ the set of normal forms of traces.

Definition 0.3: Ordering

$\leq \subseteq A^*\perp^? \times A^*\perp^?$ be the smallest relation such that for all $x, y \in A^*$, $x \leq x$ and $x\perp \leq xy\perp \leq xy$. $u < v$ if $u \leq v \not\leq u$ for all $u, v \in A^*\perp^?$.

Definition 0.4: Process

- A **(partial) process** is a sub-LTS $\langle Der(s_0), A_\perp, \longrightarrow, s_0\rangle$ satisfying the **irrecoverability** constraint

$$\forall s \in Der(s_0)[s \stackrel{\perp}{\longrightarrow} s' \Rightarrow A_\perp(s') = \{\perp\}] \qquad (1)$$

- The process is **total** if $s \stackrel{\perp}{\not\longrightarrow} s'$ for all $s, s' \in Der(s_0)$.
- If $s_0 \stackrel{u}{\longrightarrow} t$ for $t \in Der(s_0)$ and $u \in A^{*}\perp^{?}$, then $s_0 \stackrel{u}{\longrightarrow} t$ is a **behaviour** of the process.

Fact 0.2: Closed-under-transitions

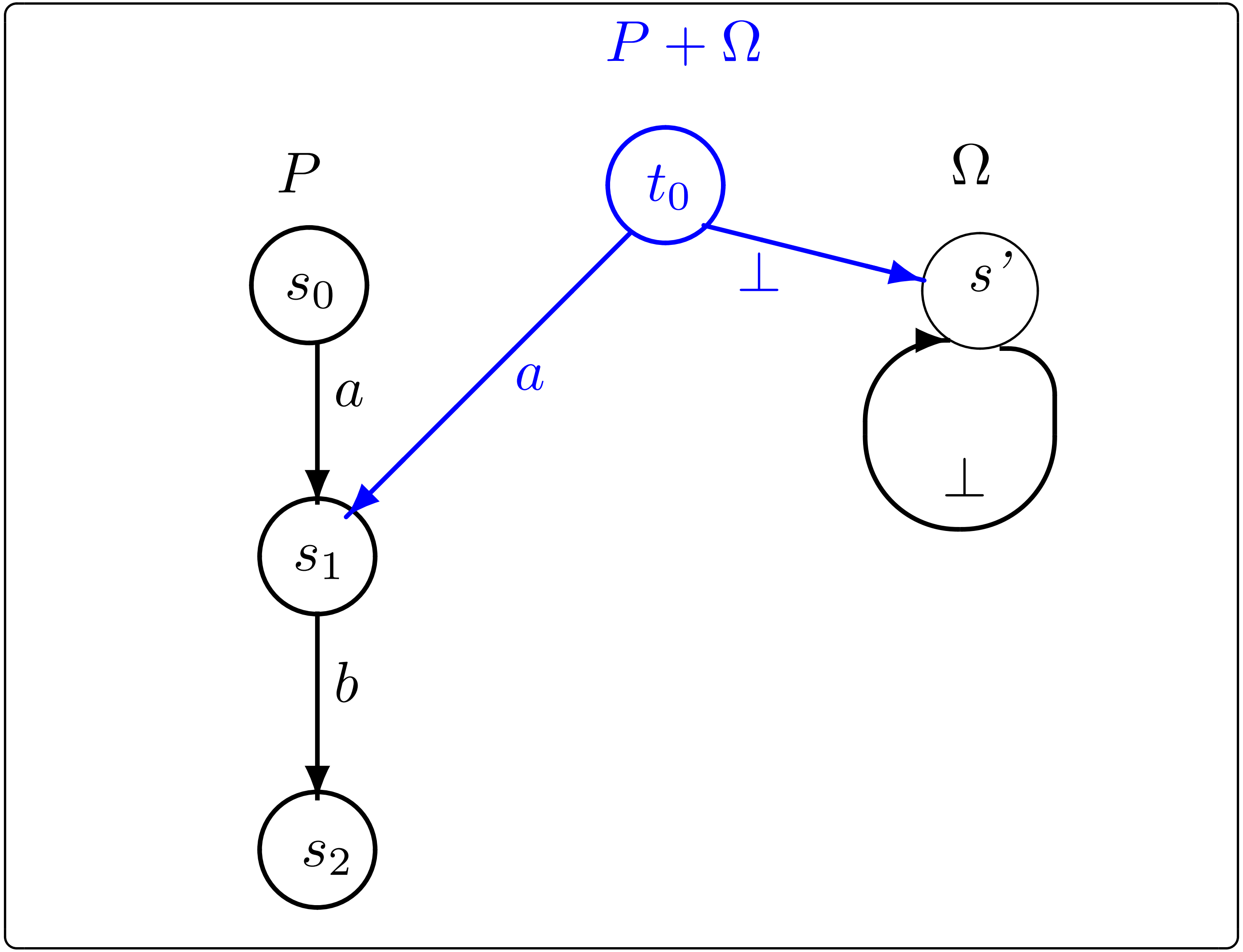
If $\langle S, A_\perp, \longrightarrow, s_0\rangle$ is a process then so is $\langle Der(s), A_\perp, \longrightarrow, s\rangle$ for any $s \in S$.

Definition 0.5: Basic Extended Process Algebra (BXPA)

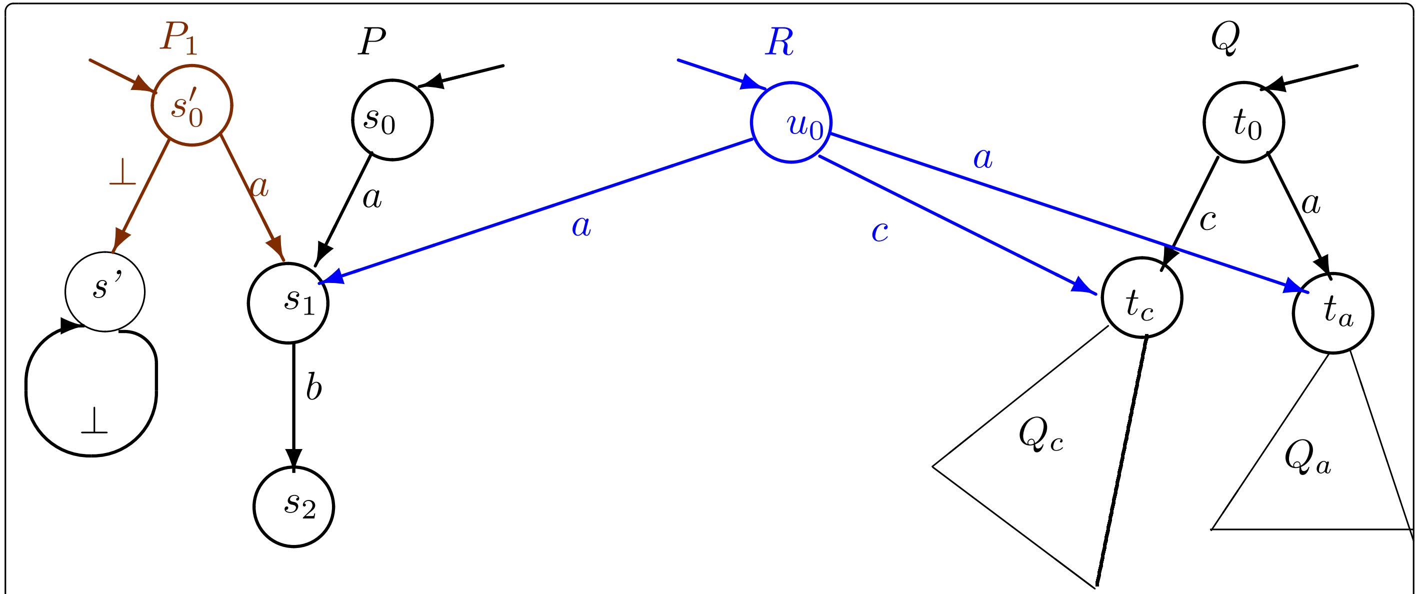
$\mathbf{P}[A_\perp] = \langle \mathbb{P}[A_\perp], \Omega, \mathbf{0}, \{a.\_ \mid a \in A\}, \sum \rangle$ where

- $\Omega \stackrel{df}{=} \langle \{s_0\}, A_\perp, \{s_0 \stackrel{\perp}{\longrightarrow} s_0\}, s_0 \rangle$ the totally undefined process,
- $\mathbf{0} \stackrel{df}{=} \langle \{s_0\}, A_\perp, \emptyset, s_0 \rangle$ is the "terminated" or "deadlocked" process.
- $a.P \stackrel{df}{=} \langle S \cup \{s'_0\}, A_\perp, \longrightarrow \cup \{s'_0 \stackrel{a}{\longrightarrow} s_0\}, s'_0 \rangle$, for any $P = \langle S, A_\perp, \longrightarrow, s_0 \rangle$, $a \in A$, and $s'_0 \notin S$,
- $\sum_{i \in I} P_i \stackrel{df}{=} \langle S, A_\perp, \longrightarrow s_0 \rangle$ where $P_i = \langle S^i, A_\perp, \longrightarrow_i, s^i_0 \rangle$, $i \in I$ and
  - $s_0 \notin \bigcup_{i \in I} S^i$ and $S = Der(s_0) = \{s_0\} \cup \biguplus_{i \in I} Targets(\longrightarrow_i)$,
  - $s_0 \stackrel{a}{\longrightarrow} t$ if for some P_i, $i \in I$, $s^i_0 \stackrel{a}{\longrightarrow}_i t \in S^i$,
  - $s \stackrel{a}{\longrightarrow} t$ if $s \stackrel{a}{\longrightarrow}_i t$ for some $i \in I$, $s, t \in Der(s_0)$.

Example: Summation 1



Example: Summation 2



$P \equiv a.b.\mathbf{0}$

$P_1 \equiv \Omega + P$

$Q \equiv c.Q_c + a.Q_a$

$R \equiv P + Q$

BXPA: Basic Identities

Proposition 0.1

$\mathbb{P}[A_\perp]$ is an idempotent abelian monoid under $+$ with $\mathbf{0}$ as identity. Further

1. $P \xrightarrow{a} P', a \in A$ implies $P \sim a.P' + P$.
2. $P \xrightarrow{\perp} P'$ implies $P' \sim \Omega$ and hence $P \sim \Omega + P$.
3. **(Canonical form modulo $\sim$).** $P \sim [\Omega+] \sum_{a \in A, P \xrightarrow{a} P_a} a.P_a$

   where "$[\Omega+]$" indicates that Ω occurs only if $P \xrightarrow{\perp}$.

Composition

Strictness condition (see irrecoverability)

$$(P \xrightarrow{\perp} \_ \vee Q \xrightarrow{\perp} \_) \implies ((P \otimes Q \xrightarrow{\perp} \Omega) \wedge (Q \otimes P \xrightarrow{\perp} \Omega)) \qquad (2)$$

guarantees that $\mathbb{P}[A_\perp]$ is closed under $\otimes$. Hence expansion laws under the various composition operators (e.g. $|||$, $|$, $\|$, $\times$) continue to hold.

Lifted Strong Bisimulation (LSB)

Definition 0.6: lifted strong bisimulations (LSB)

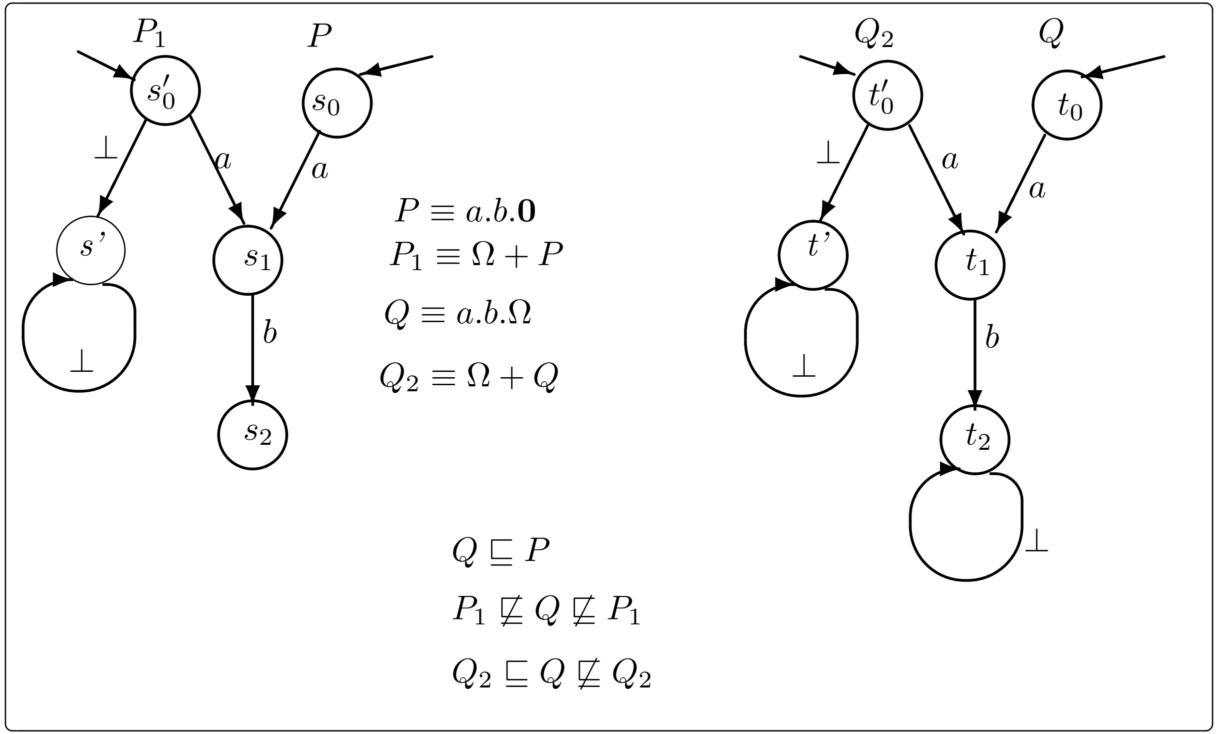
A binary relation $\mathcal{R}$ on processes is a **lifted strong bisimulation (LSB)** if for all states s, t, $s\mathcal{R}t$ implies for all $a, b \in A_{\perp,\epsilon}$,

$$s \xrightarrow{a} s' \Rightarrow \exists b, t'[a \leq b \wedge t \xrightarrow{b} t' \wedge s'\mathcal{R}t'] \quad (3)$$

$$t \xrightarrow{b} t' \Rightarrow \exists a, s'[a \leq b \wedge s \xrightarrow{a} s' \wedge s'\mathcal{R}t'] \quad (4)$$

- $s \sqsubseteq t$ (equivalently $t \sqsupseteq s$) if there exists a LSB $\mathcal{R}$ such that $s\mathcal{R}t$.
- $s \sqsubseteq\!\!\!\sqsupseteq t$ if $s \sqsubseteq t$ and $s \sqsupseteq t$.
- $s \sqsubset t$ if $s \sqsubseteq t$ and $s \not\sqsupseteq t$.

Examples



In addition, if τ is in A and $\mathbf{T} \Leftarrow \tau.\mathbf{T}$ denotes livelock, then we have $\bot < \tau$, $\bot < \epsilon$ and hence $\Omega \sqsubset \mathbf{0} \not\sqsubseteq \Omega$, $\Omega \sqsubset \mathbf{T} \not\sqsubseteq \Omega$ and $\mathbf{0} \not\sqsubseteq \mathbf{T} \not\sqsubseteq \mathbf{0}$

Precongruence

LSB is an instance of the more general (ρ, σ)-bisimulation [1] with $\rho = \sigma = \leq$. By theorem 4.1 part 1 in [1], $\sqsubseteq$ is a preorder.

Theorem 0.1: Precongruence.

The operators of $\mathbf{P}[A_\perp]$ are monotonic under $\sqsubseteq$ and the relation $\sqsubseteq$ is a precongruence on $\mathbf{P}[A_\perp]$.

Logical Characterisation

Definition 0.7

- $\mathcal{L}$ a logical language and
- $\models^X \subseteq \mathbb{P} \times \mathcal{L}$ a satisfaction relation
- $\mathcal{L}_X(P) = \{\phi \in \mathcal{L} \mid P \models^X \phi\}$
- $(\mathcal{L}, \models^X)$ **characterises** a behavioural preorder $\preceq$ over $\mathbb{P}$

$$P \preceq Q \Leftrightarrow \mathcal{L}_X(P) \subseteq \mathcal{L}_X(Q) \tag{5}$$

- $P \subseteq_X Q$ iff $\mathcal{L}_X(P) \subseteq \mathcal{L}_X(Q)$.

PHML: A Modal logic

Definition 0.8

- Negation-free modal logic $\mathcal{L}_{(\leq,\leq)}$[a]

$$\phi \;::=\; \mathtt{tt} \mid \mathtt{ff} \mid \langle a\rangle\phi \mid [a]\phi \mid \bigwedge_{i\in I}\phi_i \mid \bigvee_{i\in I}\phi_i \tag{6}$$

where $a \in A_{\perp,\epsilon}$ and I is an indexing set,

- $\bigwedge_{i\in\emptyset}\phi_i \equiv \mathtt{tt}$ and $\bigvee_{i\in\emptyset}\phi_i \equiv \mathtt{ff}$ by convention.

[a] For the present, we are assuming that every action in $A_{\perp,\epsilon}$ including the undefined action $\perp$ is observable; this may be relaxed.

Semantics: Satisfaction

Definition 0.9: Satisfaction

$P \models^S \mathit{tt}$ for each $P \in \mathbb{P}_{IF}$

$P \models^S \langle a \rangle \phi$ iff

$\exists b \in A_{\perp,\epsilon} : b \geq a, P' :$

$[P \xrightarrow{b} P' \wedge P' \models^S \phi]$

$P \models^S \bigwedge_{i \in I} \phi_i$ iff $\forall i \in I[P \models^S \phi_i]$

$P \models^S \mathit{ff}$ for no $P \in \mathbb{P}_{IF}$

$P \models^S [a]\phi$ iff

$\forall b \in A_{\perp,\epsilon} : b \leq a, P' :$

$[P \xrightarrow{b} P' \Rightarrow P' \models^S \phi]$

$P \models^S \bigvee_{i \in I} \phi_i$ iff $\exists i \in I[P \models^S \phi_i]$

- P **satisfies** ϕ if $P \models^S \phi$ and
- $\mathcal{L}_S(P) = \{\phi \mid P \models^S \phi\}$
- $P \subseteq_S Q$ if $\mathcal{L}_S(P) \subseteq \mathcal{L}_S(Q)$ for processes P, Q,

PHML characterisation of LSB

Theorem 0.2: Logical characterisation of $\sqsubseteq$

$P \sqsubseteq Q$ if and only if $\mathcal{L}_S(P) \subseteq \mathcal{L}_S(Q)$ for image-finite processes.

Theorem 0.2 then directly follows from definition 5 and theorem 3 of [3].

But

- it is difficult to explain modalities like $\langle\bot\rangle$ and $[\bot]$.
- Also modalities like $\langle\epsilon\rangle$ and $[\epsilon]$ do not add much value to the notion of observation.

Redefining old notions

- s may **diverge** ($s\uparrow$) if $s \xrightarrow{\perp}$. Otherwise it converges ($s\downarrow$). Analogously for processes.
- A binary relation $\mathcal{R}$ on processes is a **divergent strong bisimulation (DSB)** if for all $s, t \in S$, $s\mathcal{R}t$ implies the following.

$$\forall a \in A[s \xrightarrow{a} s' \Rightarrow \exists t'[t \xrightarrow{a} t' \wedge s'\mathcal{R}t'] \tag{7}$$

$$s\downarrow \Rightarrow (t\downarrow \wedge \forall a \in A[t \xrightarrow{a} t' \Rightarrow \exists s'[s \xrightarrow{a} s' \wedge s'\mathcal{R}t']]) \tag{8}$$

- $s \sqsubseteq\!\!\sim t$ (equivalently $t \sqsupseteq\!\!\sim s$) if there exists a DSB $\mathcal{R}$ such that $s\mathcal{R}t$ (we write $\mathcal{R} \vdash s \sqsubseteq\!\!\sim t$ to denote this fact). $s \sqsubset\!\!\sqsupset\!\!\sim t$ if $s \sqsubseteq\!\!\sim t$ and $s \sqsupseteq\!\!\sim t$.

Equivalence of LSB and DSB

- $\{\Omega\} \times \mathbb{P}$ is a DSB and hence $\Omega \underset{\sim}{\sqsubset} P$ for all $P \in \mathbb{P}$.
- ($\mathcal{R}$ completion) $\mathcal{R}^{\perp} = \mathcal{R} \cup \{(s, t') \mid s\mathcal{R}t, s\uparrow, A(s) = \emptyset, t\downarrow, t' \in Der(t)\}$.

Lemma 0.1

1. If $\mathcal{R}$ is a DSB then so is $\mathcal{R}^{\perp}$.
2. $\mathcal{R}$ is a DSB implies $\mathcal{R}^{\perp}$ is a LSB.
3. If $\mathcal{R}$ is a LSB then so is $\mathcal{R}^{\perp}$.
4. Every LSB is also a DSB.

Theorem 0.3

$\underset{\sim}{\sqsubset} = \sqsubseteq$ and $\underset{\sim}{\sqcup} = \underline{\sqcup}$.

$\mathcal{L}^{-\perp}$ is $\mathcal{L}$ without $\langle\perp\rangle$ and $[\perp]$.

Definition 0.10: Affirmation

$\models^A \subseteq \mathbb{P} \times \mathcal{L}^{-\perp}$ is the smallest (infix) relation defined by induction on the structure of formulae for any process P and any action $a \in A_{\perp,\epsilon}$.

$P \models^A \mathtt{tt}$ for each $P \in \mathbb{P}$

$P \models^A \mathtt{ff}$ for no $P \in \mathbb{P}$

$P \models^A \langle a\rangle\phi$ iff $\exists P'[P \xrightarrow{a} P' \wedge P' \models^A \phi]$

$P \models^A [a]\phi$ iff $P\!\downarrow \wedge\, \forall P'[P \xrightarrow{a} P' \Rightarrow P' \models^A \phi]$

$P \models^A \bigwedge_{i\in I} \phi_i$ iff $\forall i \in I[P \models^A \phi_i]$

$P \models^A \bigvee_{i\in I} \phi_i$ iff $\exists i \in I[P \models^A \phi_i]$

P **affirms** ϕ if $P \models^A \phi$ and $\mathcal{L}_A(P) = \{\phi \mid P \models^A \phi\}$. $P \subseteq_A Q$ if $\mathcal{L}_A(P) \subseteq \mathcal{L}_A(Q)$ for processes P and Q.

Characterisation

Definition 0.11

1. $P \subseteq_S^{-\perp} Q$ iff $\mathcal{L}_S^{-\perp}(P) \subseteq \mathcal{L}_S^{-\perp}(Q)$.
2. $P \subseteq_A^{-\perp} Q$ iff $\mathcal{L}_A^{-\perp}(P) \subseteq \mathcal{L}_A^{-\perp}(Q)$.

Theorem 0.4

$\boxed{\subseteq_S^{-\perp} = \sqsubseteq = \subseteq_A^{-\perp}}$ i.e. $\mathcal{L}^{-\perp}$ characterises the preorder $\sqsubseteq$.

Conclusions.

- Recursion not explicitly considered (since the model allows processes with infinite behaviours).
- But easy to see that guarded recursion (made up only of well-defined actions) will yield unique fixpoints.
- If $\tau \in A$ then $X \Leftarrow X$ and $X \Leftarrow \tau.X$ will yield different least solutions.

Future Work.

- $\sqsubset$ could be used as a refinement relation that allows the progression from a totally undefined process to a well-defined process satisfying certain modal properties.
- $\mathbb{P}[A_\perp]$ is closed under various composition operations. This allows the possibility of using more than one parallel composition operator in the specification of systems.

References

[1] S. Arun-Kumar. On bisimilarities induced by relations on actions. In *Proceedings 4th IEEE International Conference on Software Engineering and Formal Methods, Pune, India*. IEEE Computer Society Press, 2006.

[2] S. Arun-Kumar and Divyanshu Bagga. Parameterised bisimulations: Some applications. In W. Kahl P. Hofner, P. Jipsen and M. E. Muller, editors, *14th International Conference on Relational and Algebraic Methods in Computer Science*, volume 8428 of *Lecture Notes in Computer Science*. Springer-Verlag, 2014.

[3] Divyanshu Bagga and S. Arun-Kumar. Logical characterization of parameterised bisimulations. In *International Colloquium on Theoretical Aspects of Computing*, volume 10580, pages 99–112. Lecture Notes in Computer Science, 2017.

[4] M.C.B. Hennessy. *Algebraic Theory of Processes*. MIT Press, Boston, 1988.

[5] Neelesh Korade and S. Arun-Kumar. A logical characterization of efficiency preorders. In *International Colloquium on Theoretical Aspects of Computing*, volume 3407, pages 99–112. Lecture Notes in Computer Science, 2004.

[6] R. Milner. A modal characterisation of observable machine-behaviour. In *CAAP 1981*, volume 112 of *Lecture Notes in Computer Science*, Berlin Heidelberg, 1981. Springer-Verlag.

[7] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.

[8] R. Milner. *Communication and Concurrency*. Prentice-Hall International, 1989.

[9] D. S. Scott. Data types as lattices. *SIAM Journal on Computing*, 5(3):522–587, 1976.

[10] D. S. Scott. Domains for denotational semantics. In *ICALP 1982*, volume 140 of *Lecture Notes in Computer Science*, Berlin Heidelberg, 1982. Springer-Verlag.

[11] D.J. Walker. Bisimulation and divergence in CCS. In *Third Annual Symposium on Logic in Computer Science*, pages 186–192, Edinburgh, Scotland, July 1988. IEEE Computer Society Press.

Thank You!

Any Questions?