Extending process algebra with an undefined action (Extended Report)

S. Arun-Kumar*

July 2, 2022

Abstract

The quest for extensional equivalences in process algebra, over the last few decades has led authors to sometimes conflate divergence with deadlock [9], divergence with livelock [4, 5, 13] and deadlock with livelock [10]. For example, under observational equivalence which is insensitive to "τ-cycles" or infinite internal chatter, a closed system that engages in infinite internal chatter is observationally equivalent to a deadlocked one.

In this paper we take a more nuanced approach to these notions while retaining compositionality as central to the development of systems. Following Scott's [11, 12] notion of partially defined objects in the case of sequential programs, we take divergence to mean undefinedness. We define a basic extended process algebra (BXPA) to include "partially" defined processes and their behaviours. We define a behavioural preorder (actually a pre-bisimilarity) and show that it is a precongruence on BXPA. Divergent processes are the least elements in the preorder and lie below both deadlocks and livelocks which are mutually incomparable.

We extend the notion of logical characterisations of behavioural equivalences to that of behavioural preorders using a Hennessy-Milner Logic (HML) and prove the characterisation for image-finite processes using known techniques ([10], [7], [3]). Our logical characterisation of the behavioural preorder therefore, provides a gradation on the sets of well-defined observable properties of processes that reflects the behavioural preorder.

Keywords: concurrency, process algebra, Hennessy-Milner Logic, prebisimulations, prebisimilarity

1 Motivation and Related Work

Divergence, deadlock and livelock in Process Algebra. In denotational semantics [11, 12], divergence is identified with undefinedness, i.e. the least solution of the equation $X \Leftarrow X$ is the function that is undefined everywhere.

*Department of Computer Science and Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110 016,India, Email: sak@cse.iitd.ac.in

A blocked or deadlocked process is one that is incapable of engaging in any action, while a livelocked process is one that may engage in infinite "internal chatter" ($\mathbf{T} = \tau.\mathbf{T}$) but is incapable of engaging with the environment[1]. A livelocked process consumes computational cycles and energy. In the testing framework [4, 5] and the pre-bsimulation framework of Walker [13], both the equations $X \Leftarrow X$ and $X \Leftarrow \tau.X$ have the same least solution.

We take the view that both deadlock and livelock are well-defined and incomparable with each other and from divergence. Further divergent processes are less defined than any well-defined process.

Taking a line through denotational semantics we adjoin a special undefined action $\perp$ to the set of actions and use it to define partially defined processes. A divergent process is one that cannot perform any other action. A partially defined process is one that may evolve into a divergent one.

In this paper we attempt to formulate an expanded view of processes to include partially defined processes in the above sense, and try to distinguish divergence, deadlock and livelock from each other.

Organisation and contribution

- We start with some basic notions and notations of labelled transition systems (LTS) in section 2.
- In section 3, we define a basic process algebra (BPA) to include partially defined processes, by expanding the action set to include an undefined action.
- Section 4 defines various parallel composition operators using strictness conditions to ensure that expansion laws continue to hold.
- We define a strong pre-bisimulation relation on processes (in section 5) which yields a preorder $\sqsubseteq$ that we call lifted strong bisimilarity and is a precongruence on BPA. BPA/$\sqsubseteq$ yields a partial order on processes with a least element viz. the totally undefined process Ω which is is distinct from both deadlock and livelock.
- We define another pre-bisimilarity relation named divergent strong bisimilarity in section 6. It is "syntactically" inspired by the divergence predicate first defined by Milner ([8]) and applied to our setting. We show that it is equivalent to lifted strong bisimilarity. This relation allows a mild form of abstraction from undefined actions.
- We also define a parameterised Hennessy-Milner Logic [3] in section 7 with two possible semantic relations – called satisfaction and affirmation (the latter is "syntactically" inspired by [8]). We then show that both of them logically characterise the previously defined bisimilarity relations. In particular, undefinedness, deadlock and livelock (in the sense explained above) are mutually distinguishable. Affirmation allows us to abstract away from undefinedness in properties.
- Section 8 is the conclusion.

To provide a more coherent reading experience we have relegated most proofs to the appendix. In addition to the usual notions of theorem, proposition, lemma and corollary we also have "fact(s)" which typically follow directly from definitions or some previous remarks. We use "claims" within proofs to divide up and structure proofs to make them more perspicuous.

[1]The CCS process $R = (P|Q)\backslash a \sim \tau.R$, where $P = a.P$ and $Q = \bar{a}.Q$ is one such since $R \sim \mathbf{T}$.

2 Labelled Transition Systems: Basics

Definition 2.1 (Labelled Transition System (LTS))

- *A* **labelled transition system (LTS)** $\mathbb{L}[L]$ *over a set of* **labels** *L is a tuple* $\mathbb{L}[L] = \langle S, L, \longrightarrow\rangle$*, where S is a (possibly infinite) set of* **states** *and* $\longrightarrow \subseteq S \times L \times S$ *is the* **transition relation**. *(s, ℓ, t) is written $s \xrightarrow{\ell} t$; s is the* source*, ℓ is the* label *and t is the* target *of the transition.*
- *A* **rooted LTS** *is a LTS $\mathbb{L}[L]$ with a distinguished* **start state** *$s_0 \in S$ and denoted $\langle S, L, \longrightarrow, s_0\rangle$.*

Notational conventions and terminology. Let $\mathbb{L}[L] = \langle S, L, \longrightarrow\rangle$ be a LTS. _ denotes a place-holder in the following.

- $s \xrightarrow{\ell} \_ = \{t \in S \mid s \xrightarrow{\ell} t\}$ is the set of *ℓ-successors* of s.
- $s \xrightarrow{\ell}$ iff $s \xrightarrow{\ell} \_ \neq \emptyset$, otherwise $s \overset{\ell}{\not\longrightarrow}$.
- $L(s) = \{\ell \in L \mid s \xrightarrow{\ell}\}$ is the set of labels from s.
- $s \xrightarrow{-} \_ = \{(\ell, t) \in L \times S \mid s \xrightarrow{\ell} t\}$
- $s \not\longrightarrow$ iff $s \xrightarrow{-} \_ = \emptyset$
- $\_ \xrightarrow{\ell} t = \{s \in S \mid s \xrightarrow{\ell} t\}$ is the set of *ℓ-predecessors* of t.
- $\_ \xrightarrow{-} t = \{(s, \ell) \in S \times L \mid s \xrightarrow{\ell} t\}$
- $Succ(s) = \bigcup_{\ell \in L} s \xrightarrow{\ell} \_ = \{t \mid \exists \ell \in L[s \xrightarrow{\ell} t]\}$ is the set of *successors* of s
- $Der(s) = \{s\} \cup \bigcup_{t \in Succ(s)} Der(t)$ is the set of *derivatives* of s. $\bigcup_{t \in Succ(s)} Der(t)$ is the set of *proper derivatives* of s.
- $Sources(\longrightarrow) = \{s \in S \mid s \xrightarrow{-} \_ \neq \emptyset\}$
- $Targets(\longrightarrow) = \{t \in S \mid \_ \xrightarrow{-} t \neq \emptyset\}$.

Lemma 2.2 (LTS: Basic properties.)

- *For label sets $L \subseteq M$, a LTS over L is also a LTS over M.*
- *For any collection (indexed by a set I) of LTSs over a label set L, $\{\mathbb{L}_i[L] \mid i \in I\}$ such that $\mathbb{L}_i[L] = \langle S_i, L, \longrightarrow_i\rangle$, $(i \in I)$, their union is the LTS $\mathbb{L}[L] = \langle S, L, \longrightarrow\rangle$, where $S = \bigcup_{i \in I} S_i$ and $\longrightarrow = \bigcup_{i \in I} \longrightarrow_i$.*

Definition 2.3 (sub- and derived- LTSs) *Let $\mathbb{L}[L] = \langle S, L, \longrightarrow\rangle$ be a LTS. Then*

- *A* **sub-LTS** $\mathbb{L}'[L] = \langle S', L, \longrightarrow'\rangle$ *of a LTS $\mathbb{L}[L]$ is a LTS over L such that $S' \subseteq S$ and $s \xrightarrow{-} \_ \subseteq L \times S'$ for all $s \in S'$.*
- $\mathbb{L}[L^*] = \langle S, L^*, \longrightarrow\rangle$ *is the* **LTS derived** *from $\mathbb{L}[L]$ where $\longrightarrow$ is overloaded to refer to the least relation such that $s \xrightarrow{\epsilon} s$ for all $s \in S$ and for all $x \in A^+$, $s \xrightarrow{x} s'$ for $x = ay$ iff $\exists s'' : s \xrightarrow{a} s'' \xrightarrow{y} s'$.*

Definition 2.4 *A binary relation $\mathcal{R} \subseteq S \times T$ between (sub-)LTSs $\mathbb{L}[L] = \langle S, L, \longrightarrow\rangle$ and $\mathbb{M}[L] = \langle T, L, \longrightarrow\rangle$ is a* **natural bisimulation** *if $s\mathcal{R}t$ implies for all labels $\ell \in L$, $s \xrightarrow{\ell} s' \Rightarrow \exists t' \in T[t \xrightarrow{\ell} t' \wedge s'\mathcal{R}t']$ and $t \xrightarrow{\ell} t'' \Rightarrow \exists s'' \in S[s \xrightarrow{\ell} s'' \wedge s''\mathcal{R}t'']$. s is said to be* **naturally bisimilar** *to t (denoted $s \sim t$) if $s\mathcal{R}t$ for some natural bisimulation $\mathcal{R}$ (notation: $\mathcal{R} \vdash s \sim t$).*

Facts 2.5

1. *Unions, relational converses and (relational) compositions of natural bisimulations are also natural bisimulations.*
2. *Natural bisimilarity ($\sim$) is the largest natural bisimulation and is an equivalence relation.*
3. *Every natural bisimulation on $\mathbb{L}[L]$ is also a natural bisimulation on $\mathbb{L}[L^*]$ and vice-versa.*

We identify the derived LTS $\mathbb{L}[L^*]$ with the LTS $\mathbb{L}[L]$ and may keep switching between them as per convenience.

3 Basic Extended Process Algebra (BXPA)

Definition 3.1 *Let $A_\bot = A \cup \{\bot\}$ be the set of all* **actions** *where A is a countable set of (uninterpreted but) well-defined actions and $\bot \notin A$ is a special undefined action with $\bot < a$ for each $a \in A$.*

Our notion of a process is a rooted (sub-)LTS over a set of actions. It is convenient for us to introduce an undefined action $\bot$ which is the only action that can be performed by the totally undefined process Ω (to be defined below). This action is less defined than any other action from the action set A. Once a process descends to a state that performs $\bot$, it remains in that state and can perform only $\bot$ then on and can never recover to a well-defined state. The traces that we need to consider are therefore from $A^*\bot^*$. Since a process may only perform sequences of actions of the form $x\bot^*$ for any $x \in A^*$ we find it convenient to quotient out the set $A^*\bot^*$ by the equation $\boxed{x\bot\bot = x\bot}$ to yield the set of normal forms $A^*\bot^? = A^* \cup A^*\bot$ of traces.
Notational convention. x, y, z denote words from A^* and u, v, w denote words from $A^*\bot^?$. In general any $u \in A^*\bot$ is of the form $u = x.\bot$ where $x \in A^*$.

Definition 3.2 *Let $\leq \subseteq A^*\bot^? \times A^*\bot^?$ be the smallest relation such that*

- *$x \leq x$ for all $x \in A^*$ and*
- *$x.\bot \leq x.y.\bot \leq x.y$ for all $x, y \in A^*$*

Facts 3.3 *Let $u < v$ denote $u \leq v \nleq u$ for all $u, v \in A^*\bot^?$.*

1. *For all $x, y \in A^*$, $x \leq y$ iff $x = y$.*
2. *$\bot < \epsilon$, where ϵ is the empty string.*
3. *$\bot < a$ for every $a \in A$*[2]*.*
4. *$a.\bot < a = a\epsilon = \epsilon a$.*

Lemma 3.4 (Partial ordering)

1. *$\langle A^*\bot^?, \leq\rangle$ is a partial order (proof in the appendix).*
2. *$\langle A_\bot, \leq\rangle$ and $\langle A_{\bot,\epsilon} \leq\rangle$ are both flat complete partial orders (cpo), where $A_{\bot,\epsilon} = A_\bot \cup \{\epsilon\}$.*

[2]In particular $\bot < \tau$ if $\tau \in A$.

Definition 3.5 (Process) *Let $\mathbb{L}[A_\perp] = \langle S, A_\perp, \longrightarrow\rangle$ be a LTS. A* **(partial) process** *is a rooted sub-LTS $\langle Der(s_0), A_\perp, \longrightarrow, s_0\rangle$ satisfying the constraint*

$$\boxed{\textbf{Irrecoverability. } \forall s \in Der(s_0)[s \xrightarrow{\perp} s' \Rightarrow A_\perp(s') = \{\perp\}] \qquad (3.1)}$$

- *The process is* **total** *if $s \stackrel{\perp}{\not\longrightarrow} s'$ for all $s, s' \in Der(s_0)$.*
- *If $s_0 \xrightarrow{u} t$ for $t \in Der(s_0)$ and $u \in A^*\perp^?$, then $s_0 \xrightarrow{u} t$ is a* **behaviour** *of the process.*
- *The process is* **image-finite** *if $\forall s \in Der(s_0)\forall a \in A_\perp[|s \xrightarrow{a}| < \infty]$.*

Fact 3.6 *If $\langle S, A_\perp, \longrightarrow, s_0\rangle$ is a process (resp. image-finite), then for any $s \in S$, so is $\langle Der(s), A_\perp, \longrightarrow, s\rangle$.*

Notational conventions and terminology.

1. $s \xrightarrow{a} s'$ is an *undefined transition* if $a = \perp$, otherwise it is *well-defined*.
2. Upper-case latin letters P, Q, R etc. (possibly decorated) denote processes.
3. Lower-case initial latin letters a, b, c etc. (possibly decorated) denote individual actions (including $\perp$ and the empty trace ϵ).
4. Processes are identified with their start states (fact 3.6) and all relations between processes are also relations between their start states. Hence, if P and Q are processes with start states s_0 and t_0 resp. such that $s_0 \xrightarrow{a} t_0$ then we simply write $P \xrightarrow{a} Q$. Similarly $P \sim Q$ if their respective start states are naturally bisimilar.

Definition 3.7 (Basic Extended Process Algebra (BXPA)) *The structure*
$\mathbf{P}[A_\perp] = \langle \mathbb{P}[A_\perp], \Omega, \mathbf{0}, \{a.\_ \mid a \in A\}, \sum\rangle$ *where*

- **Omega.** $\Omega \overset{df}{=} \langle\{s_0\}, A_\perp, \{s_0 \xrightarrow{\perp} s_0\}, s_0\rangle$ *is the totally undefined process,*
- **Nil.** $\mathbf{0} \overset{df}{=} \langle\{s_0\}, A_\perp, \emptyset, s_0\rangle$ *is the "terminated", "blocked", "deadlocked" or "stop" process,*
- **Prefixing.** $a.P \overset{df}{=} \langle S \cup \{s_0'\}, A_\perp, \longrightarrow \cup \{s_0' \xrightarrow{a} s_0\}, s_0'\rangle$*, for any $P = \langle S, A_\perp, \longrightarrow, s_0\rangle$, $a \in A$, and $s_0' \notin S$,*
- **Summation.** *For any sequence $[P_i \mid i \in I, P_i = \langle S^i, A_\perp, \longrightarrow_i, s_0^i\rangle]$ of processes indexed by a set I, their sum is $\sum\limits_{i \in I} P_i \overset{df}{=} \langle S, A_\perp, \longrightarrow, s_0\rangle$ where $s_0 \notin \bigcup\limits_{i \in I} S^i$ and*
  - $S = Der(s_0) = \{s_0\} \cup \biguplus\limits_{i \in I} Targets(\longrightarrow_i)$*,*
  - $s_0 \xrightarrow{a} t$ *iff for some P_i, $i \in I$, $s_0^i \xrightarrow{a}_i t \in S^i$,*
  - $s \xrightarrow{a} t$ *if $s \xrightarrow{a}_i t$ for some $i \in I$, $s, t \in Der(s_0)$.*

is called **Basic Extended Process Algebra (BXPA)**.

$\mathbb{P}[A_\perp]$ contains processes with infinite behaviours too. $\mathbb{P}_{IF}[A_\perp]$ denotes the set of image-finite processes. We do not allow prefixing with the undefined action. In

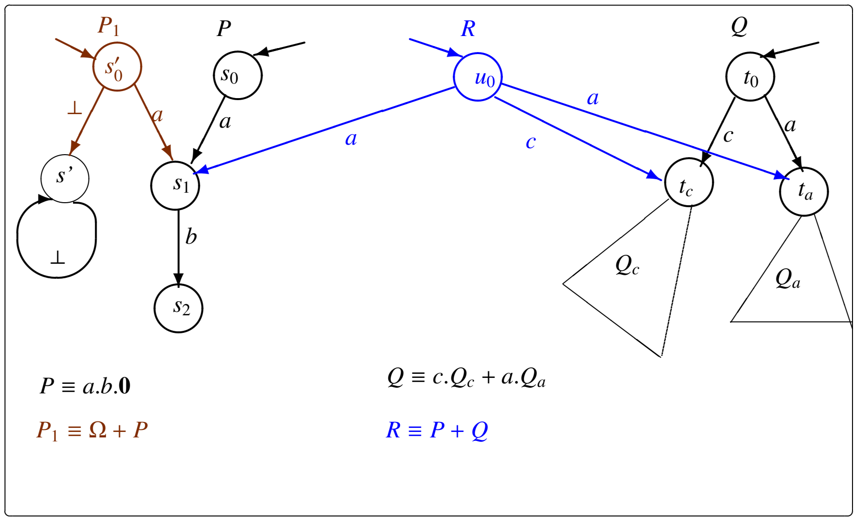

Figure 1: Summation illustration

the case of summation, the root s_0 is a new state representing the effect of coalescing all the start states s_0^i, $i \in I$. However, whether an s_0^i belongs to S depends upon whether it is a proper derivative of itself. For example (see figure 1) given $\Omega \equiv \langle \{s'\}, A_\perp, \{s' \xrightarrow{\perp} s'\}, s'\rangle$ the start state s' is also in the target set of Ω and will hence be included in the set of states of $P_1 \equiv \Omega + P$ with a new start state s'_0 which is neither in the states of Ω nor in the states of P. However the start state of P viz. s_0 is not included in the set of states of P_1 since it is not present in the targets of P. We do sometimes consider the sub-structure $\mathbf{P}[A] = \langle \mathbb{P}[A], \mathbf{0}, \{a.\_ \mid a \in A\}, \sum\rangle$ consisting of **total** processes, none of whose states may perform the undefined action. In this structure the natural bisimilarity relation between states reduces to the usual notion of strong bisimilarity.

Some notation.

1. By convention $\sum_{i \in \emptyset} P_i \equiv \mathbf{0}$ and (by abuse of notation) $\Omega \equiv \perp.\Omega$.
2. When $|I| = 2$ we use the binary infix symbol "+", (e.g. $P_1 + P_2$) to denote their sum.
3. Let $[P_i \mid 1 \le i \le n]$ be any finite sequence of processes. We write $P_1 + \cdots + P_n$ to denote $\sum_{1 \le i \le n} P_i$.

Proposition 3.8 (Basic Process identities) *The following identities hold for all processes P, Q, R.*

$$\begin{array}{llcl}
(\sim\text{+Associativity}) & (P+Q)+R & \sim & P+(Q+R) \\
(\sim\text{+Commutativity}) & P+Q & \sim & Q+P \\
(\sim\text{+Identity}) & P+\mathbf{0} & \sim & P \\
(\sim\text{+Idempotence}) & P+P & \sim & P
\end{array}$$

Fact 3.9 *The following are easy to prove.*

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* $P \sim P + \Omega$
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* $P \xrightarrow{\perp}$.

4 Parallel Composition

Since our motivations come primarily from concurrent, parallel and distributed systems, we define a few parallel composition operators well-known from the literature. The composition of two or more processes needs to yield a process that satisfies the irrecoverability (3.1) constraint of definition 3.5. For any parallel composition operator $\otimes$ we impose the following pointwise *strictness* condition which guarantees that $\mathbb{P}[A_\perp]$ is closed under $\otimes$.

$$\boxed{\textbf{Strictness}.\ (P \xrightarrow{\perp} \_ \vee Q \xrightarrow{\perp} \_) \implies ((P \otimes Q \xrightarrow{\perp} \Omega) \wedge (Q \otimes P \xrightarrow{\perp} \Omega)) \quad (4.1)}$$

In this section we assume that $P = \langle S, A_\perp, \longrightarrow, s_0 \rangle$ and $Q = \langle T, A_\perp, \longrightarrow, t_0 \rangle$ are partial processes in the set $\mathbb{P}[A^{*?}_{\perp}]$. We may redefine the set A as appropriate.

Definition 4.1 (Interleaving) $P \mid\mid\mid Q = \langle S \mid\mid\mid T, A_\perp, \longrightarrow, s_0 \mid\mid\mid t_0 \rangle$ *where* $S \mid\mid\mid T \subseteq S \times T$ *and* $\longrightarrow$ *is the smallest relation (subject to the strictness condition (4.1)) such that for all* $a, b \in A$*, the following hold.*

$$\mid\mid\mid\ \textbf{Left}.\ s \xrightarrow{a} s' \implies s \mid\mid\mid t \xrightarrow{a} s' \mid\mid\mid t \tag{4.2}$$

$$\mid\mid\mid\ \textbf{Right}.\ t \xrightarrow{b} t' \implies s \mid\mid\mid t \xrightarrow{b} s \mid\mid\mid t' \tag{4.3}$$

Definition 4.2 (CSP $\|$) [3] $P\|Q = \langle S\|T, A_\perp, \longrightarrow, s_0\|t_0 \rangle$ *where* $S\|T \subseteq S \times T$ *and* $\longrightarrow$ *(subject to the strictness condition (4.1)) is the smallest relation such that for all* $a, b, c \in A$*, the following hold.*

$$\|\ \textbf{Left}.\ s \xrightarrow{a} s' \wedge t \not\xrightarrow{a} \implies s\|t \xrightarrow{a} s'\|t \tag{4.4}$$

$$\|\ \textbf{Right}.\ s \not\xrightarrow{b} \wedge t \xrightarrow{b} t' \implies s\|t \xrightarrow{b} s\|t' \tag{4.5}$$

$$\|\ \textbf{Sync}.\ s \xrightarrow{c} s' \wedge t \xrightarrow{c} t' \implies s\|t \xrightarrow{c} s'\|t' \tag{4.6}$$

[3]For simplicity we assume that all actions in A are in the common action set of all processes.

CSP allows multi-way synchronization with the synchronizing actions kept "visible" (as opposed to the CCS case). CCS on the other hand, allows only point-to-point communication resulting in a new distinguished action τ whenever the processes synchronize and interleave otherwise. There is more structure in the action sets of CCS and SCCS.

Definition 4.3 (CCS |) *Let $A \supseteq Act = \Lambda \cup \overline{\Lambda} \cup \{\tau\}$*[4]*, where Λ and $\overline{\Lambda}$ are disjoint complementary sets of labels in bijection with each other under the* complementation *operation $\bar{\ }$ and $\tau \notin \Lambda \cup \overline{\Lambda}$ is a distinguished* internal *action (usually obtained by a synchronization of complementary actions related by the bijection). Complementation is extended so that $\overline{\tau} = \tau$, $\overline{\bot} = \bot$ and $\overline{\overline{a}} = a$ for all $a \in \Lambda$.*

Then $P|Q = \langle S|T, A_\bot, \longrightarrow, s_0|t_0\rangle$ where $S|T \subseteq S \times T$ and $\longrightarrow$ (subject to the strictness condition (4.1)), is the smallest relation such that for all $a, b, c \in A$, the following hold.

$$\textbf{| Left}. \qquad s \xrightarrow{a} s' \implies s|t \xrightarrow{a} s'|t \tag{4.7}$$

$$\textbf{| Right}. \qquad t \xrightarrow{b} t' \implies s|t \xrightarrow{b} s|t' \tag{4.8}$$

$$\textbf{| Sync}. \qquad s \xrightarrow{c} s' \wedge t \xrightarrow{\overline{c}} t' \implies s|t \xrightarrow{\tau} s'|t' \tag{4.9}$$

In the case of SCCS, the set A of well defined actions is a commutative group freely generated from a set Λ of particles and their inverses $\overline{\Lambda}$ by a commutative product operation $\times$ and containing the identity element 1 ($a \times \overline{a} = 1 = \overline{a} \times a$ for all $a \in \Lambda$). To this set A we adjoin the undefined action $\bot$ and extend the product and inverse operations to be strict on each operand, i.e. for all $a \in A_\bot$, $a \times \bot = \bot = \bot \times a$ and $\overline{\bot} = \bot$.

Definition 4.4 (Synchronous Product (SCCS)) *For A defined as above, $P \times Q = \langle S \times T, A_\bot, \longrightarrow, s_0 \times t_0\rangle$ where $(s,t) \in S \times T$ is written $s \times t$ and $\longrightarrow$ is the smallest relation (subject to the strictness condition (4.1)) such that for all $a, b \in A$,*

$$\textbf{Product}. \; s \xrightarrow{a} s' \wedge t \xrightarrow{b} t' \implies s \times t \xrightarrow{a \times b} s' \times t' \tag{4.10}$$

The following lemma shows that every process is naturally bisimilar to one in canonical form. It also shows that all the behaviours of the parallel compositions of processes may be captured upto natural bisimilarity by processes in $\mathbb{P}[A_\bot]$.

Lemma 4.5 (Expansion Laws). *Let $P \sim [\Omega +] \sum_{i \in I, a_i \in A} a_i.P_i$ and $Q \sim [\Omega +] \sum_{j \in J, b_j \in A} b_j.Q_j$ with $A(P) = \{a_i \in A \mid i \in I\}, A(Q) = \{b_j \in A \mid j \in J\}$, where the set A is appropriately*

[4]It is sometimes useful (see [6]) to assume $Act \subseteq A$ with complementation not defined for labels in $A - Act$.

chosen (see definitions (4.1, 4.2,4.3, 4.4)). Then

$$\begin{array}{lcl}
P \mid\mid\mid Q & \sim & [\Omega+]\sum_{i\in I} a_i.(P_i \mid\mid\mid Q) + \sum_{j\in J} b_j.(P \mid\mid\mid Q_j) \\
P \| Q & \sim & [\Omega+]\sum_{i\in I, a_i \in A(P)-A(Q)} a_i.(P_i \| Q) + \sum_{j\in J, b_j\in A(Q)-A(P)} b_j.(P \| Q_j) \\
 & & + \sum_{a_i = c = b_j \neq \perp, i\in I, j\in J} c.(P_i \| Q_j) \\
P | Q & \sim & [\Omega+]\sum_{i\in I} a_i.(P_i | Q) + \sum_{j\in J} b_j.(P | Q_j) + \sum_{a_i=\overline{b_j}, i\in I, j\in J} \tau.(P_i \| Q_j) \\
P \times Q & \sim & [\Omega+]\sum_{i\in I, j\in J} (a_i \times b_j).(P_i \times Q_j)
\end{array}$$

Without loss of generality, we assume in the sequel that (whenever required) every process is expressed in canonical form upto $\sim$. In the sequel, we therefore restrict ourselves to the set of (partial) processes $\mathbb{P}[A_\perp]$ (as in definition 3.7). Equivalently, since we identify a LTS with its derived LTS, $\mathbb{P}[A^{*}{\perp}^{?}]$ is the set of all partial processes over A.

5 Lifted Strong Bisimulations

In [1] bisimulation was generalised to (ρ, σ)-bisimulation for binary relations ρ and σ on the set of actions. Further in [2] many bisimilarities defined in the literature were shown to inherit their nice algebraic and relational properties from the properties of the underlying relations on actions.

Definition 5.1 (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 the following for all $a, b \in A_{\perp,\epsilon}$.*

$$s \xrightarrow{a} s' \Rightarrow \exists b, t'[a \leq b \wedge t \xrightarrow{b} t' \wedge s'\mathcal{R}t'] \tag{5.1}$$

$$t \xrightarrow{b} t' \Rightarrow \exists a, s'[a \leq b \wedge s \xrightarrow{a} s' \wedge s'\mathcal{R}t'] \tag{5.2}$$

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

We may equally well define a LSB in terms of the partial order $\langle A^{*}{\perp}^{?}, \leq\rangle$. Note that $\perp \leq \epsilon$ (part 2 of Facts 3.3) and for any terminated state t, $t \xrightarrow{\epsilon} t$ holds.

Lemma 5.2 (Equivalent definition of LSB) *A binary relation $\mathcal{R}$ is a LSB if for all $s, t \in S$, $s\mathcal{R}t$ implies the following for all $u, v \in A^{*}{\perp}^{?}$.*

$$s \xrightarrow{u} s' \Rightarrow \exists v, t'[u \leq v \wedge t \xrightarrow{v} t' \wedge s'\mathcal{R}t'] \tag{5.3}$$

$$t \xrightarrow{v} t' \Rightarrow \exists u, s'[u \leq v \wedge s \xrightarrow{u} s' \wedge s'\mathcal{R}t'] \tag{5.4}$$

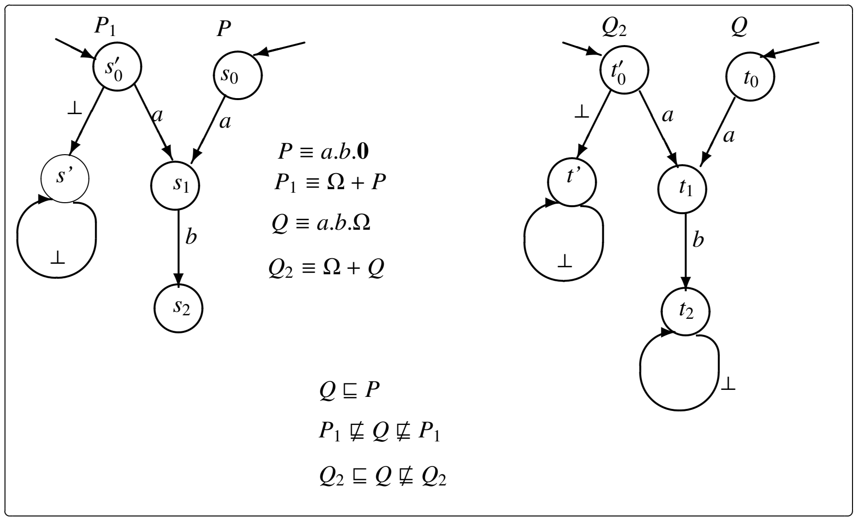

Figure 2: Processes incomparable and comparable under $\sqsubseteq$ resp.

We write $\mathcal{R} \vdash P \sqsubseteq Q$ to denote that $\mathcal{R}$ is a LSB containing the pair (P, Q) from which $P \sqsubseteq Q$ follows. If $\mathcal{R} \vdash P \sqsubseteq Q$ and $\mathcal{S} \vdash Q \sqsubseteq P$, we write $\mathcal{R} \vdash P \sqsupseteq\sqsubseteq Q \dashv \mathcal{S}$.

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.

Example 5.3 *(See figure 2). Let $P \overset{df}{=} a.b.\mathbf{0}$ and $Q \overset{df}{=} a.b.\Omega$ Then $Q \sqsubseteq P \not\sqsubseteq Q$. processes P_1 and Q are incomparable because they have different points at which they may exhibit an undefined behaviour; in particular $s_2 \not\sqsubseteq t_2 \sqsubseteq s_2$. However $\{(s_0, t_0), (s_1, t_1), (s_2, t_2), (s', t_1), (s', t_2)\} \vdash P_2 \sqsubseteq Q$ whereas $Q \not\sqsubseteq P_2$ since P_2 is more undefined than Q.*

Facts 5.4 *The following are easy to show from the results in [1], [10] and definition 5.1.*

1. *$\mathcal{R}$ is a LSB iff for all $u, v \in A^{*?}_{\perp}$, conditions (5.3) and (5.4) hold whenever $s\mathcal{R}t$.*
2. *$\sqsubseteq$ is the largest LSB.*
3. *$\sqsubseteq$ is a preorder on $\mathbb{P}$ (see theorem 4.1 in [1]).*
4. *Any LSB over $\mathbb{P}[A]$ is a strong bisimulation.*
5. *$\sqsubseteq$ restricted to $\mathbb{P}[A]$ is the strong bisimilarity relation $\sim$ defined in [10].*
6. *$\{(\Omega, P) \mid P \in \mathbb{P}[A_\perp]\}$ is a LSB. In particular, $\Omega \sqsubset \mathbf{0}$ i.e. $\Omega \sqsubseteq \mathbf{0} \not\sqsubseteq \Omega$.*
7. *$P \sqsupseteq\sqsubseteq \Omega$ iff $P \sim \Omega$.*
8. *Since $\sim \subset \sqsupseteq\sqsubseteq$ over $\mathbb{P}[A_\perp]$ the identities in proposition 3.8 and lemma 4.5 also hold when $\sim$ is replaced by $\sqsupseteq\sqsubseteq$. In particular, we also have $\Omega \sqsupseteq\sqsubseteq \Omega + \mathbf{0}$, though in general, for any process $P \in \mathbb{P}[A]$, $\Omega + P \sqsupseteq\sqsubseteq P + \Omega \sqsubseteq P$ holds, whereas $P \sqsubseteq P + \Omega$ may not hold.*

Theorem 5.5 (Precongruence) *The operators of $\mathbf{P}[A_\perp]$ are monotonic under $\sqsubseteq$ and the relation $\sqsubseteq$ is a precongruence on $\mathbf{P}[A_\perp]$.*

See appendix for the proof. If τ were to be included in the set A of actions (with $\perp < \tau$) then livelock (or "infinite internal chatter") could be defined as the process $\mathbf{T} \overset{df}{=} \langle \{s_0\}, A_\perp, \{s_0 \xrightarrow{\tau} s_0\}, s_0 \rangle$. It is then easy to see that $\Omega \sqsubseteq \mathbf{T} \not\sqsubseteq \Omega$ and $\mathbf{0} \not\sqsubseteq \mathbf{T} \not\sqsubseteq \mathbf{0}$. Taken in conjunction with $\Omega \sqsubseteq \mathbf{0} \not\sqsubseteq \Omega$ we see that divergence, deadlock and livelock are pairwise distinct and divergence lies below both deadlock and livelock in the ordering.

6 Divergent Strong Bisimulation (DSB)

Notions of divergence originally defined by Milner ([8]) have been used by various authors ([5, 4, 13]) to define preorders on processes. In all the above cases the preorders/equivalences were obtained by abstracting away from τ actions. We reinterpret those notions in the context of undefinedness in our formulation.

Definition 6.1 *We say s* **converges** *or is* **convergent** *(and denote it by $s{\downarrow}$) if $s \overset{\perp}{\not\longrightarrow}$. If s is not convergent we say s* **may diverge** *(or is* **divergent***) and denote it by $s{\uparrow}$.*

We extend this notation to processes. In particular we have,

Fact 6.2 *$P \sqsubseteq Q$ and $P{\downarrow}$ implies $Q{\downarrow}$.*

The analogue of the preorder defined by Milner in ([8]) in our context is as follows.

Definition 6.3 (Divergent Strong Bisimulation (DSB)) *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.*

$$\forall a \in A[s \xrightarrow{a} s' \Rightarrow \exists t'[t \xrightarrow{a} t' \wedge s'\mathcal{R}t']] \tag{6.1}$$

$$s \downarrow \Rightarrow (t \downarrow \wedge \forall a \in A[t \xrightarrow{a} t' \Rightarrow \exists s'[s \xrightarrow{a} s' \wedge s'\mathcal{R}t']]) \tag{6.2}$$

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

Notice that the first clause (6.1) in definition 6.3 in our setting, implies that if $s \xrightarrow{\perp} s'$ then even if $t \overset{\perp}{\not\longrightarrow}$, we could choose $t \xrightarrow{\epsilon} t$ and we have $s'\mathcal{R}t$. That is, (6.1) is equivalent to (5.1). This also means that (6.1) is equivalent to (5.3). The clause (6.2) however, is conditional upon s being convergent.

Fact 6.4 *$\{\Omega\} \times \mathbb{P}$ is a DSB and hence $\Omega \underset{\sim}{\sqsubset} P$ for all $P \in \mathbb{P}$.*

Example 6.5 *In figure 3 let $a, b \in A$. We see that $s_0{\uparrow}$ but it is capable of performing the well-defined action $a \in A$. On the other hand $s'{\uparrow}$ too, but cannot perform any well-defined action. It follows that $\mathcal{S} = \{(s_0, t_0), (s_1, t_1)\}$, $\mathcal{S}' = \mathcal{S} \cup \{(s', t_1)\}$, $\mathcal{S}'' = \mathcal{S} \cup \{(s', t')\}$ and $\mathcal{S}''' = \mathcal{S} \cup \{(s', t_1), (s', t')\}$ are all DSBs. Hence $P_3 \underset{\sim}{\sqsubset} Q_3$.*

For any relation $\mathcal{R}$ let $\mathcal{R}^\perp = \mathcal{R} \cup \{(s, t') \mid s\mathcal{R}t, s{\uparrow}, A(s) = \emptyset, t{\downarrow}, t' \in Der(t)\}$. We refer to $\mathcal{R}^\perp$ as the $\perp$-*completion* of $\mathcal{R}$.

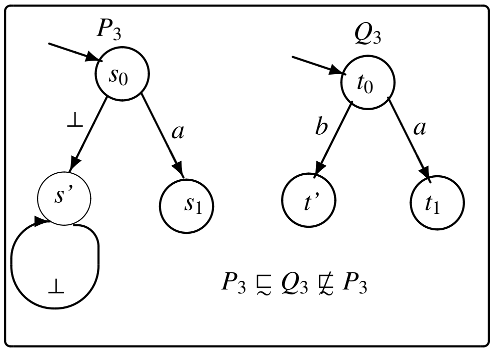
P_3
Q_3
s_0
t_0
$\perp$
a
b
a
s'
s_1
t'
t_1
$\perp$
$P_3 \underset{\sim}{\sqsubseteq} Q_3 \underset{\sim}{\not\sqsubseteq} P_3$

Figure 3: A pair of processes comparable under divergent strong bisimulation

Example 6.6 *From figure 3 and example 6.5 we see that $\mathcal{S}'''$ and $\mathcal{S}^{\perp}$ are both LSBs. They are both DSBs as well.*

Lemma 6.7

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 6.8 $\underset{\sim}{\sqsubseteq} = \sqsubseteq$ *and* $\underset{\sim}{\sqsubseteq\!\!\!\sqsupseteq} = \sqsubseteq\!\!\!\sqsupseteq$.

Proof: Follows from lemma 6.7. QED

Hence every DSB may be $\perp$-completed to yield a LSB. By theorem 6.8, in order to to prove $P \sqsubseteq Q$, it suffices to construct a DSB between their derivatives ignoring pairs of states (s, t) where $s\uparrow$ and $A(s) = \emptyset$ unless s is the target of a well-defined transition.

7 Modal Characterisation for Image-finite Processes

In [8] Milner makes a case for defining equivalences of agents extrinsically using properties drawn from a simple modal language. In [10] he has also defined what it means for a logic to characterise a behavioural equivalence relation viz. that two processes are behaviourally equivalent if and only if they satisfy the same properties defined by the logic. In [7] the definition was generalised to the logical characterisation of behavioural *preorders*, in terms of containment of properties. We give a definition below that subsumes both Milner's definition for behavioural equivalences and the one in [7].

Throughout this section we restrict ourseleves to only image-fintie processes. Besides the obvious reasonability of this constraint, there is also a technical reason for this restriction which is beyond the scope of this paper (see §3.1 of [3]).

Definition 7.1 (Logical characterisation of a behavioural preorder) *Let $(\mathcal{L}, \models^X)$ be a logic consisting of a language $\mathcal{L}$ and a relation $\models^X \subseteq \mathbb{P}_{IF} \times \mathcal{L}$. $(\mathcal{L}, \models^X)$* **characterises**

a behavioural preorder $\preceq$ over $\mathbb{P}$ if for any $P, Q \in \mathbb{P}$, $P \preceq Q$ iff $\mathcal{L}_X(P) \subseteq \mathcal{L}_X(Q)$, where $\mathcal{L}_X(P) = \{\phi \in \mathcal{L} \mid P \models^X \phi\}$. $P \subseteq_X Q$ iff $\mathcal{L}_X(P) \subseteq \mathcal{L}_X(Q)$.

In [8] Milner then proceeds to treat his notion of divergence within Hennessy-Milner Logic (HML) and shows that the relation of affirmation which he defines, characterises behavioural equivalence. That is, two processes are behaviourally equivalent if and only if they affirm exactly the same set of properties from the modal language.

In [3] the authors have generalised Hennessy-Milner Logic (HML) to a parameterised form called PHML corresponding to the parameterisation of bisimulations and bisimilarities in [1]. In general however, PHML defined over a set of "observables" (which may or may not be the set of "actions") is not a modal logic unless certain conditions are met. Rather, it is a negation-free logic which reduces to a modal logic whenever the parameters ρ and σ satisfy the conditions (see theorem 4.1 in [1])

- ρ and σ are both equivalences and so $\sqsubseteq_{(\rho,\sigma)}$ is an equivalence, or else
- the preorders ρ and σ are not equivalences, and
  - either $\rho = \sigma$ in which case the induced bisimilarity relation $\sqsubseteq_{(\rho,\rho)}$ is a preorder,
  - or $\rho = \sigma^{-1}$ in which case the bisimilarity relation $\sqsubseteq_{(\rho,\sigma)}$ is an equivalence.

In the present case, we have $\sqsubseteq = \sqsubseteq_{(\leq,\leq)}$ since it is induced by the preorder (actually partial order) $\leq \subseteq A_{\perp,\epsilon} \times A_{\perp,\epsilon}$. Hence PHML [3] applied to our present context ($\rho = \leq = \sigma$) directly yields a modal logic. We define the resulting modal logic $\mathcal{L}_{(\leq,\leq)}$ (see [3] definition 5) discarding superscripts on the modal operators as being redundant in the present case.

Definition 7.2 (Syntax of PHML.) *The language $\mathcal{L}_{(\leq,\leq)}$ is defined by the following BNF over the set of "observables" $A_{\perp,\epsilon}$*[5]*.*

$$\phi \ ::= \ \mathbf{tt} \mid \mathbf{ff} \mid \langle a\rangle\phi \mid [a]\phi \mid \bigwedge_{i\in I} \phi_i \mid \bigvee_{i\in I} \phi_i \tag{7.1}$$

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

In particular, we also have $\bigwedge_{i\in\emptyset} \phi_i \equiv \mathbf{tt}$ and $\bigvee_{i\in\emptyset} \phi_i \equiv \mathbf{ff}$. In the sequel we omit the subscript $_{(\leq,\leq)}$ as being understood.

Definition 7.3 (Satisfaction) *$\models^S \subseteq \mathbb{P}_{IF} \times \mathcal{L}$ is the smallest (infix) relation defined by induction on the structure of formulae for any process P and any action $a \in A_{\perp,\epsilon}$.*

$P \models^S \mathbf{tt}$ *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* $\forall i \in I[P \models^S \phi_i]$

$P \models^S \mathbf{ff}$ *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* $\exists i \in I[P \models^S \phi_i]$

[5]For the present, we are assuming that every action in $A_{\perp,\epsilon}$ including the undefined action $\perp$ is observable; we relax this later.

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,*

We may directly apply the results in [3] (definition 5 and theorem 3) to our present formulation of $\sqsubseteq$ to yield a logical characterisation of the preorder $\sqsubseteq$.

Theorem 7.4 (Logical characterisation of $\sqsubseteq$) *$P \sqsubseteq Q$ if and only if $\mathcal{L}_S(P) \subseteq \mathcal{L}_S(Q)$.*

Proof: Directly follows from theorem 3 in [3] for image-finite processes. QED

Despite the fact that our view and formulation of divergence differ from those of Milner's in [8], it is instructive in the light of the characterisation given in Theorem 6.8 to construct a notion of affirmation *à la* Milner, as it allows us to dispense with modalities involving the undefined action. It is important to note however, that the affirmation semantics we present below is only akin to Milner's in form rather than in actual meaning, since

- our notion of divergence (using an undefined action) is quite distinct from Milner's, and
- there is no place for formulae such as $\langle\bot\rangle\phi$, $[\bot]\phi$, $\langle\epsilon\rangle\phi$ or $[\epsilon]\phi$ in Milner's logical language.

Definition 7.5 (Affirmation) *$\models^A \subseteq \mathbb{P} \times \mathcal{L}$ is the smallest (infix) relation defined by induction on the structure of formulae for any process P and any action $a \in A_{\bot,\epsilon}$.*

$P \models^A \mathbf{tt}$ *for each* $P \in \mathbb{P}$	$P \models^A \mathbf{ff}$ *for no* $P \in \mathbb{P}$
$P \models^A \langle a\rangle\phi$ *iff*	$P \models^A [a]\phi$ *iff*
$\exists P'[P \xrightarrow{a} P' \wedge P' \models^A \phi]$	$P{\downarrow} \wedge \forall P'[P \xrightarrow{a} P' \Rightarrow P' \models^A \phi]$
$P \models^A \bigwedge_{i\in I} \phi_i$ *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.*

The similarities in the two semantic formulations are as follows.

1. All the formulae $[\epsilon]\mathbf{ff}$, $\langle\bot\rangle\mathbf{ff}$, $\langle\epsilon\rangle\mathbf{ff}$, and $\langle a\rangle\mathbf{ff}$ for any $a \in A$ never hold for any process and hence are equivalent to $\mathbf{ff}$ in both semantics.
2. The two semantics yield identical meanings (actually process models) for $\langle\bot\rangle\phi$.
3. $\langle\epsilon\rangle\phi$ is equivalent to ϕ in both semantics.

As may be expected, the differences mostly stem from the treatment of negation (which is actually hidden in the logic) and the condition $P{\downarrow}$ in the definition of $P \models^A [a]\phi$.

1. $P \models^S [\bot]\mathbf{tt}$ always holds whereas $P \models^A [\bot]\mathbf{tt}$ holds only when $P{\downarrow}$.
2. $P \models^S [\epsilon]\mathbf{tt}$ holds for every process whereas $P \models^A [\epsilon]\mathbf{tt}$ holds again only if $P{\downarrow}$.
3. For any $a \in A$, $P \models^S [a]\mathbf{tt}$ holds for all processes whereas $P \models^A [a]\mathbf{tt}$ holds only if $P{\downarrow}$.
4. $P \models^A [\bot]\phi$ if and only if $P{\downarrow}$, whereas $P \models^S [\bot]\phi$ if and only if $P{\downarrow}$ or every $\bot$-successor of P satisfies ϕ.
5. $P \models^S [\epsilon]\phi$ if and only if $P \models \phi$, (regardless of whether it converges or diverges) whereas $P \models^A [\epsilon]\phi$ if and only if $P{\downarrow}$ and $P \models^A \phi$.

The question of considering modalities $\langle\perp\rangle$ and $[\perp]$ and whether $\perp$ is "observable" is a troublesome one and it would be preferable to avoid using it in the modal logic. We consider the modal language $\mathcal{L}^{-\perp} \subset \mathcal{L}$ which excludes the modal prefixes $\langle\perp\rangle$ and $[\perp]$. However we do retain the prefixes $\langle\epsilon\rangle$ and $[\epsilon]$. For any process P, $\mathcal{L}_S^{-\perp}(P)$ (respectively $\mathcal{L}_A^{-\perp}(P)$) denotes the set of formulae in $\mathcal{L}^{-\perp}$ that P satisfies (respectively affirms).

Definition 7.6

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 7.7 (Characterisation of $\subseteq_S^{-\perp}$ and $\subseteq_A^{-\perp}$) $\boxed{\subseteq_S^{-\perp} = \sqsubseteq = \subseteq_A^{-\perp}}$ *i.e. $\mathcal{L}^{-\perp}$ characterises the preorder $\sqsubseteq$.*

Proof: Follows from the claims below (whose proofs are given in the appendix).

Claim (1)	$\Omega \subseteq_S^{-\perp} P$ for all P	*Claim* (2)	$\Omega \subseteq_A^{-\perp} P$ for all P
Claim (3)	$\subseteq_S^{-\perp} \subseteq \sqsubseteq$	*Claim* (4)	$\subseteq_A^{-\perp} \subseteq \sqsubseteq$
Claim (5)	$\sqsubseteq \subseteq \subseteq_S^{-\perp}$	*Claim* (6)	$\sqsubseteq \subseteq \subseteq_A^{-\perp}$

QED

By obtaining the logical characterisation entirely based on the action set A_ϵ, we have shown that the $\perp$ modalities ($\langle\perp\rangle$ and $[\perp]$) do not influence the discrimination power of the modal logic. Hence they may be dispensed with altogether. It is also possible to provide a proof that independently shows that $\underset{\sim}{\sqsubseteq}$ is characterised by $\subseteq_S^{-\perp}$. However that is unnecessary since $\sqsubseteq = \underset{\sim}{\sqsubseteq}$ by theorem 6.8.

8 Conclusion

We have expanded the notion of process to include agents which are capable of performing an undefined action. By stipulating that it lies below all other actions, we have been able to define a totally undefined process Ω which lies below all other processes. In particular, $\Omega \sqsubseteq \mathbf{0} \not\sqsubseteq \Omega$. We may also include a distinguished silent action τ in the action set A. Since all actions in A are mutually incomparable, deadlock ($\mathbf{0}$) and livelock ($\mathbf{T}$) will not be equated by the relation $\sqsubseteq\!\sqsupseteq$.

We have not explicitly addressed the question of recursion. However, it is easy to see that guarded recursion equations made up of only well-defined actions, will yield unique fixpoints as solutions. It is also reasonably clear that if $\tau \in A$, then the two equations $X \Leftarrow X$ and $X \Leftarrow \tau.X$ would yield different (least) solutions, Ω and $\mathbf{T}$ respectively, with $\Omega \sqsubseteq \mathbf{T} \not\sqsubseteq \Omega$.

We have followed the usual process algebra practice that the notions of "observability" and "distinguishability" of processes are best decided by a formal logic that can express the properties of interest. To ensure that, it is necessary that the logic be expressive enough to characterise the appropriate behavioural relations that are of interest to us. However, there is a lack of clarity in the notion of an "undefined action" – whether it is "observable" or "visible" and if so what actually may be observed in a run that contains it etc. But we have shown that its presence does not unduly affect the theory of process behaviours (theorem 6.8). In fact, the syntactic rendering of Milner's notions of divergence and affirmation ([8]) apply directly in our case too (e.g. divergent strong bisimulations and the affirmation semantics) and yield results that are consistent with our own notions of the differences between divergence, deadlock and livelock.

For processes to be closed under parallel composition it is necessary to impose the strictness condition (4.1) [6]. Otherwise any interleaving of undefined and well-defined actions would violate the irrecoverability constraint 3.1.

As an aside, it is interesting that a purely syntactic rendering of convergence and divergence in our framework seems to yield results similar to those obtained by Milner in [8].

The algorithm for on-the-fly-computation of (ρ, σ)-bisimulations for finite A given in [1] (section 6) may be directly applied to compute LSBs with $\rho = \sigma = \leq$ and the complexity remains the same ($O(n^2|A_\perp|^2)$ time and $O(n + |A_\perp|^2)$ space, where n is the number of states in the product LTS) since comparisons under $\leq$ take constant time. Some small improvements are possible if we choose instead to construct DSBs (lemma 6.7) by saving on $\perp$-completions.

Future work. The relation $\sqsubseteq$ could be used as a refinement relation which allows a progression from a totally undefined process to one which satisfies a certain specification (expressed in terms of the properties or the behaviours that need to be satisfied).

The fact that the algebra is closed (upto $\sim$) under the various parallel composition operators opens up the possibility that a formal specification language could potentially use more than one parallel composition operator. The challenge in designing such a language would then rest on defining a suitable syntax and a structural operational semantics that is faithful to our model. We conjecture that it may be useful in hardware-software codesign.

9 Appendix

Proof of lemma 3.4

- $\underline{\text{Reflexivity}}$. Trivial from definition 3.2.
- $\underline{\text{Transitivity}}$. Assume $u \leq v \leq w$. If $u \in A^*$ then clearly $u = v = w$ and it follows that $u \leq w$. Otherwise $u = x\perp$ for some $x \in A^*$, which implies by clause 2 of definition 3.2 that $v = x.y\perp$ for some $y \in A^*$. Likewise since $v \leq w$, $w = (x.y).z\perp = x.(y.z)\perp$ for some $z \in A^*$. Hence $u \leq w$.
- $\underline{\text{Antisymmetry}}$. Assume $u \leq v \leq u$. If $u \in A^*$, by clause 1 of definition 3.2 it trivially follows that $u = v$. If not then, $u = x\perp$ for some $x \in A^*$ and by clause 2 of definition 3.2 $v = xy\perp$ for some $y \in A^*$. However $v \leq u$ implies that $xy\perp \leq x\perp$ which is possible only if $y = \varepsilon$. Hence $u = v$.

Proof of Theorem 5.5

It is easy to see that

1. $\{(a.P, a.Q) \mid a \in A, P \sqsubseteq Q\}$ is a LSB.
2. $\{(\sum_{i \in I} P_i, \sum_{i \in I} Q_i) \mid \forall i \in I[P_i \sqsubseteq Q_i]\}$ is a LSB.

[6] e.g. $P|\Omega \sqsupseteq \Omega|P \sqsupseteq P + \Omega$

Proof of lemma 6.7 part 1

$\mathcal{R} \subseteq \mathcal{R}^{\perp}$ always. If $\mathcal{R} = \mathcal{R}^{\perp}$ there is nothing to prove. Hence suppose $s\mathcal{R}^{\perp}t'$ and $(s, t') \notin \mathcal{R}$. Then since $s\uparrow$ and $A(s) = \emptyset$, clearly $s \xrightarrow{\perp} s$ is the only action that s can perform and for some t we have $s\mathcal{R}t$ and $t\downarrow$ and for some $x \in A^*$, $t \xrightarrow{x} t'$. It is clear that (s, t') satisfies the clause (6.1) since $s \xrightarrow{\perp} s$, $t' \xrightarrow{\epsilon} t'$ and $s\mathcal{R}^{\perp}t'$. Further (s, t') vacuously satisfies the clause (6.2). Hence $\mathcal{R}^{\perp}$ is a DSB.

Proof of lemma 6.7 part 2

Let $\mathcal{R}$ be a DSB. We have already seen that the clause (6.1) is equivalent to clause (5.1). To prove that clause (6.2) implies clause (5.4), consider any pair $(s, t) \in \mathcal{R}^{\perp}$.

If $s\downarrow$ (so $s \overset{\perp}{\not\longrightarrow}$) we have $t\downarrow$ and hence $t \overset{\perp}{\not\longrightarrow}$. If $t \xrightarrow{b} t'$ for some $b \in A$ and state t' then there does exist s' with $s \xrightarrow{b} s'$, $s'\mathcal{R}t'$ and since $b \geq b$ it follows that clause (5.4) holds.

If on the other hand if $s\uparrow$ and $A(s) = \emptyset$ then $s \xrightarrow{\perp} s$ is the only action that s can perform. In such a case if $t\uparrow$ too, there is nothing to prove. If $t\downarrow$ then we have the following cases.

Case $t \xrightarrow{\epsilon} t$. Since $\perp \leq \epsilon$ and $s\mathcal{R}t$ clause (5.4) holds.

Case $t \xrightarrow{b} t'$, for some $b \in A$. Then $s \xrightarrow{\perp} s$ and $\perp \leq b$ and $(s, t') \in \mathcal{R}^{\perp}$.

Proof of claims (1) and (2) of theorem 7.7

1. We need to show that $\Omega \models^S \phi$ implies for any process Q, $Q \models^S \phi$. We proceed by induction on the structure of formulae ϕ. The basis cases $\phi \equiv \mathbf{tt}$ and $\phi \equiv \mathbf{ff}$ are trivial. The cases of conjunction and disjunction also follow easily from the induction hypothesis.
   Case $\phi \equiv \langle a\rangle\psi, a \in A_\epsilon$. If $\Omega \xrightarrow{a}$ then clearly $a = \epsilon$ and $\Omega \xrightarrow{\epsilon} \Omega$ and $\Omega \models^S \psi$. Clearly by the induction hypothesis for every Q we have $Q \models^S \psi$ and $Q \xrightarrow{\epsilon} P$ and therefore $Q \models^S \phi$.
   Case $\phi \equiv [b]\psi, b \in A_\epsilon$. $\Omega \models^S [b]\psi$ implies for every $a \leq b$ and P' such that $\Omega \xrightarrow{b} P'$, $P' \models^S \psi$. Since $\Omega \xrightarrow{\perp} \Omega$ and $\Omega \xrightarrow{\epsilon} \Omega$ are the only possibilities, a can only be either $\perp$ or ϵ and P' can only be Ω. By the induction hypothesis in each case we have for every Q, $Q \models^S \psi$. If $Q \xrightarrow{\perp} Q'$ then $Q' = \Omega$ and hence $Q' \models^S \psi$. On the other hand $Q \xrightarrow{\epsilon} Q' = Q$ and by induction hypothesis since $\Omega \models^S \psi$ we have $Q \models^S \psi$.
2. The only changes in the proof that $\Omega \models^A \phi$ implies for any process Q, $Q \models^A \phi$ occur in the above two cases. Since $\Omega \xrightarrow{\epsilon} \Omega$ is the only transition possible besides $\Omega \xrightarrow{\perp} \Omega$, the case $\phi \equiv \langle a\rangle\psi, a \in A_\epsilon$ reduces to $\phi \equiv \langle\epsilon\rangle\psi$ and it follows trivially that $\Omega \models^A \psi$ and since $Q \xrightarrow{\epsilon} Q$, $Q \models^A \psi$. The case $\phi \equiv [b]\psi, b \in A_\epsilon$ is trivial because $\Omega\uparrow$ and hence $\Omega \not\models^A [b]\psi, b \in A_\epsilon$.

Proof of claims (3) and (4) of theorem 7.7

3. Assume $P \subseteq_S^{-\perp} Q$, i.e. $\mathcal{L}_S^{-\perp}(P) \subseteq \mathcal{L}_S^{-\perp}(Q)$.
   (a) If $P \xrightarrow{\perp} P'$ then $P' \sim \Omega$ and it is clear from claim (1) that $P' \sqsubseteq Q'$ for

any $b \geq \perp$ and $Q \xrightarrow{b} Q'$ (notice that $\epsilon \geq \perp$ and $Q \xrightarrow{\epsilon} Q$ always holds). Suppose $P \xrightarrow{a} P'$ for some $a \in A_\epsilon$. Then $P \models^S \langle a \rangle \bigwedge \mathcal{L}_S^{-\perp}(P')$. Hence $Q \models^S \langle a \rangle \bigwedge \mathcal{L}_S^{-\perp}(P')$ which implies $Q \xrightarrow{b} Q'$ for some $b \geq a$. But $a \neq \perp$ implies $b = a$ and hence there exists Q' such that $Q' \models^S \bigwedge \mathcal{L}_S^{-\perp}(P')$.

   (b) Assume $Q \xrightarrow{b} Q''$. We need to show that there exists $a \leq b$ and P'' such that $P \xrightarrow{a} P''$ and $P'' \subseteq_S^{-\perp} Q''$. Assume the contrary i.e. for every $a \leq b$ and P'', $P \xrightarrow{a} P''$ implies $P'' \not\subseteq_S^{-\perp} Q''$.
   Let $\Psi = \{\bigwedge \mathcal{L}_S^{-\perp}(P'') \mid P \xrightarrow{a} P'', a \leq b\}$. We have $P \models^S [b] \bigvee \Psi$. Since $P \subseteq_S^{-\perp} Q$ we have $Q \models^S [b] \bigvee \Psi$. Since $Q \xrightarrow{b} Q''$, there must be some $a \leq b$ and some Q' such that $Q \xrightarrow{a} Q'$ and $Q' \models^S \bigvee \Psi$ which in turn implies that $Q' \models^S \bigwedge \mathcal{L}_S^{-\perp}(P')$ for some $P' \in P \xrightarrow{a'}$ _ for some $a' \leq b$, which is a contradiction.

4. In the proof of part 3a, replace all occurrences of $\models^S$ by $\models^A$ to obtain the proof for clause (5.3) of definition 5.1. To prove clause (5.4) of definition 5.1 we proceed as follows. If $P\uparrow$ then there is nothing to prove. So we assume that $P\downarrow$ (i.e. $P \stackrel{\perp}{\not\longrightarrow}$) and $Q \xrightarrow{b} Q''$. We need to show that there exist $a \leq b$ (in this case $a = b$) and P'' such that $P \xrightarrow{b} P''$ and $P'' \subseteq_A^{-\perp} Q''$. Assume the contrary i.e. for every P'', $P \xrightarrow{b} P''$ implies $P'' \not\subseteq_A^{-\perp} Q''$. Let $\Psi = \{\bigwedge \mathcal{L}_A^{-\perp}(P'') \mid P \xrightarrow{b} P''\}$. We have $P \models^A [b] \bigvee \Psi$ and hence $Q \models^A [b] \bigvee \Psi$. Since $Q \xrightarrow{b} Q''$, there must be some Q' such that $Q' \models^A \bigvee \Psi$ which in turn implies that $Q' \models \bigwedge \mathcal{L}_A^{-\perp}(P')$ for some $P' \in P \xrightarrow{b}$ _ which is a contradiction.

Proof of claims (5) and (6) of theorem 7.7

5. Assume $P \sqsubseteq Q$. To prove that $P \subseteq_S^{-\perp} Q$ i.e. that $\mathcal{L}_S^{-\perp}(P) \subseteq \mathcal{L}_S^{-\perp}(Q)$ we prove that $P \models^S \phi$ implies $Q \models^S \phi$ by induction on the structure of ϕ. The case $\phi \equiv \mathbf{tt}$ is trivial and so are the cases of conjunction and disjunction. So we proceed directly to the modal operators.
   (a) $\underline{\text{Case } \phi \equiv \langle a \rangle \psi, a \in A_\epsilon}$. If $P \models^S \langle a \rangle \psi$ then for some P' and $b \geq a$ we have $P \xrightarrow{b} P'$ and $P' \models^S \psi$. But $a \in A_\epsilon$ and since $A_{\perp,\epsilon}$ is a flat cpo, we have $b = a$ and hence $P \xrightarrow{a} P'$ and $P' \models^S \psi$. Since $P \sqsubseteq Q$, there must be Q' such that $Q \xrightarrow{a} Q'$ and $P' \sqsubseteq Q'$. By the induction hypothesis we obtain $Q \models^S \langle a \rangle \psi \equiv \phi$.
   (b) $\underline{\text{Case } \phi \equiv [b]\psi, b \in A_\epsilon}$. If $P \models^S [b]\psi$ then for each $a \leq b$ and $P \xrightarrow{a} P'$ we have $P' \models^S \psi$. Since $P \sqsubseteq Q$, for each a such that $P \xrightarrow{a} P'$ there must be $a' \geq a$ and Q' with $Q \xrightarrow{a'} Q'$ such that $P' \sqsubseteq Q'$. By the induction hypothesis for each such Q' we have $P' \models^S \psi$ implies $Q' \models^S \psi$. Since $b \in A_\epsilon$, the only values of a that are possible are $a = \perp$ and $a = b$. If $a = \perp$

then $P' \sim \Omega$ and we have from claim (2) that $P' \subseteq_S^{-\perp} Q'$. On the other hand if $a = b$ then $a' = b$ and again we have by induction hypothesis that $P' \subseteq_S^{-\perp} Q'$. From all of this we get that $Q \models^S [b]\psi$.

6. Assume $P \sqsubseteq Q$. We prove that $P \models^A \phi$ implies $Q \models^A \phi$ by induction on the structure of ϕ. Again the only non-trivial cases are the formulae with modal prefixes.
   (a) <u>Case $\phi \equiv \langle a\rangle\psi, a \in A_\epsilon$</u>. If $P \models^A \langle a\rangle\psi$ then for some P' $P \xrightarrow{a} P'$ and $P' \models^A \psi$. The rest follows easily as in part 5a.
   (b) <u>Case $\phi \equiv [b]\psi, b \in A_\epsilon$</u> If $P \models^A [b]\psi$ then $P\!\downarrow$ and by fact 6.2 $Q\!\downarrow$ too. For each $P' \in P \xrightarrow{b}$ _ we have $P' \models^A \psi$ and for each such P' there exists $Q' \in Q \xrightarrow{b}$ _ such that $P' \sqsubseteq Q'$. By the induction hypothesis $P' \models^A \psi$ implies $Q' \models^A \psi$. Hence $Q \models^A [b]\psi \equiv \phi$.

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] R. De Nicola and M.C.B. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1983.

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

[6] Astrid Kiehn and S. Arun-Kumar. Amortised bisimulations. In *Formal Techniques for Networked and Distributed Systems*, volume 3731, pages 320–334. Lecture Notes in Computer Science, 2005.

[7] 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.

[8] 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.

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

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

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

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

[13] 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.