On Bisimilarities Induced by Orderings

S. Arun-Kumar[1*]

Department of Computer Science and Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India. email: sak@cse.iitd.ernet.in

February 27, 2008

Abstract. In this paper, we generalize the notion of bisimulations to "bisimulations induced by a pair of relations" on the underlying action set. We establish that many of the nice properties of bisimulations and bisimilarities may be thought of as actually being inherited from properties of the underlying orderings on actions. We show that many bisimulation-based orderings (including strong and weak bisimilarity) defined in the literature are instances of this generalization. We also show by an example that there are instances where the equivalence of two systems (which intuitively have the same functionality), cannot be established directly by observational equivalence.

We explore these new generalized notions by defining new preorders and equivalences based on an abstract notion of cost. By suitably translating the notion of cost from purely semantic models of concurrent systems we show that this notion may be used to translate processes expressed as labelled transition systems into labelled transition systems with cost-oriented transitions. We prove transformation theorems for both the synchronous and asynchronous products of LTSs into cost-oriented LTSs, which preserve existing bisimulation based notions that have been defined for them in the literature.

1 Introduction

In this paper we generalize the notion of bisimulations [12] (and bisimilarity) by parametrizing the notion on a pair of binary relations imposed on the set of actions. The usual notion of bisimulation is obtained as a special case when both binary relations are the identity relation on actions. With a minimal number of assumptions we show that some of the nice properties of bisimulations and bisimilarity relations may be viewed as being induced by the corresponding properties of the underlying relations on actions. In particular, we may view the properties of reflexivity, symmetry and transitivity of both strong and weak bisimilarity as being inherited from the underlying equivalence relations on the set of actions.

We develop these notions in the setting of labelled transition systems with a parallel composition operation. We restrict our attention to parallel composition operators which respect an expansion law (e.g. interleaving parallelism as in CCS and lock-step parallelism as in SCCS). This implies that processes represented as rooted labelled transition systems are closed under a product operation that represents their parallel composition. These cases provide instances of how one may define cost-oriented bisimulation relations by basing them on a “temporal” ordering or a “spatial” ordering on the underlying action sets respectively.

We explore these new generalized notions by defining new preorders and equivalences based on an abstract notion of cost. By suitably translating the notion of cost from purely semantic models of concurrent systems we show that this notion may be used to translate processes expressed as labelled transition systems into labelled transition systems with cost-oriented transitions. We prove transformation theorems for both the synchronous and asynchronous products of LTSs into cost-oriented LTSs, which preserve existing bisimulation based notions that have been defined for them in the literature.

The paper is organized as follows. In section 2 we introduce our generalized notions and prove some basic inheritance properties. Particular cases when the two underlying partial orders on the action set are the same are further

[*] corresponding author

explored in this section. In section 3 we illustrate, through an example, the need for more generalized notions than strong and weak bisimulation. In section 4 we define cost oriented bisimilarity relations. In section 5 we discuss temporal preorders and prove that they may be captured in a cost-oriented framework. Similarly in section 6.2 we discuss preorders under a synchronous parallel composition operation and show their transformation into the same cost-oriented framework. Section 7 is the conclusion.

The proofs of most of these results are quite easy (and given in the appendix for completeness). The main contribution of this paper is the articulation of a generalized viewpoint which we believe has its uses.

2 (ρ, σ)-Bisimulations

A *labelled transition system (LTS)* $\mathcal{L}$ is a 3-tuple $\langle \mathbf{P}, Act, \longrightarrow \rangle$, where $\mathbf{P}$ is a set of *process states* or *processes*, Act is a (possibly countable) set of actions and $\longrightarrow \subseteq \mathbf{P} \times Act \times \mathbf{P}$ is the *transition relation*. We use the notation $p \xrightarrow{a} q$ to denote $(p, a, q) \in \longrightarrow$ and refer to q as an a-*derivative* or an a-*successor* of p. q is a *derivative* or *successor* of p if it is an a-successor for some action a. q is *reachable* from p if either $p = q$ or q is reachable from some successor of p. A *rooted* labelled transition system is a 4-tuple $\langle \mathbf{P}, Act, \longrightarrow, p_0 \rangle$ where $\langle \mathbf{P}, Act, \longrightarrow \rangle$ is a LTS with a distinguished *initial state* $p_0 \in \mathbf{P}$. In general we will consider the set of states of such a LTS as consisting only of those states that are reachable from the initial state. The term "process" will be used to refer to a process state in a LTS, as also to the sub-LTS rooted at that state and containing all the states and transitions reachable from that given state. Since an arbitrary disjoint union of LTSs is also an LTS, we shall often refer to $\mathbf{P}$ as the set of all processes.

Even though our proofs are not specific to any particular process calculus, we often use the syntax of CCS [10] for defining processes of particular interest. In particular, $\mathbf{0}$ denotes the process which can perform no action whatsoever. Other notational conventions we use are:

- $\equiv$ for the identity relation on a set. It may be used in the context of actions, processes and also sets of processes.
- $\circ$ to denote relational composition
- $\wp(U)$ to denote the powerset of a set U.

Definition 1. Let $\mathbf{P}$ be the set of processes and let ρ and σ be binary relations on Act. A binary relation $R \subseteq \mathbf{P} \times \mathbf{P}$ is a (ρ, σ)**-induced bisimulation** or simply a (ρ, σ)**-bisimulation** if pRq implies the following conditions.

$$\forall a \in Act[p \xrightarrow{a} p' \Rightarrow \exists b, q'[a\rho b \wedge q \xrightarrow{b} q' \wedge p'Rq']] \tag{1}$$

and

$$\forall b \in Act[q \xrightarrow{b} q' \Rightarrow \exists a, p'[a\sigma b \wedge p \xrightarrow{a} p' \wedge p'Rq']] \tag{2}$$

The largest (ρ, σ)-bisimulation (under set containment) is called (ρ, σ)**-bisimilarity** and denoted $\sqsubseteq_{(\rho,\sigma)}$. A $(\equiv, \equiv)$-induced bisimulation will sometimes be called a **natural bisimulation**[1]. $\mathsf{B}_{(\rho,\sigma)}$ denotes the set of all (ρ, σ)-bisimulations.

Given binary relations ρ, ρ', σ and σ' on Act, ρ is *at aleast as fine as* (or *no coarser than*) ρ' if $\rho \subseteq \rho'$ and ρ is *finer* than ρ' if $\rho \subset \rho'$. This notion is extended pointwise to pairs of relations on Act and by abuse of notation we write $(\rho, \sigma) \subseteq (\rho', \sigma')$ to mean that (ρ, σ) is no coarser than (ρ', σ'). The following facts are proven in the same way as they are for bisimulations and the reader is referred to [10] for their proof.

Proposition 2. *Let ρ and σ be binary relations on Act and let R and S be binary relations on the set $\mathbf{P}$ of processes.*

1. *The empty relation $\emptyset$ on processes and the relation $\{\langle \mathbf{0}, \mathbf{0} \rangle\}$ are both (ρ, σ)-bisimulations.*

[1] A strong bisimulation on CCS processes is an example of a *natural bisimulation*.

2. *Arbitrary unions of (ρ,σ)-bisimulations are also (ρ,σ)-bisimulations.*
3. *Let $\mathcal{B}_{(\rho,\sigma)}$ be a function on binary relations on $\mathbf{P}$ such that, $\langle p,q\rangle \in \mathcal{B}_{(\rho,\sigma)}(R)$ iff p and q satisfy the conditions (1) and (2) of definition 1. Then*
   (a) *$\mathcal{B}_{(\rho,\sigma)}$ is monotonic i.e. $R \subseteq S$ implies $\mathcal{B}_{(\rho,\sigma)}(R) \subseteq \mathcal{B}_{(\rho,\sigma)}(S)$.*
   (b) *R is a (ρ,σ)-bisimulation iff $R \subseteq \mathcal{B}_{(\rho,\sigma)}(R)$.*
   (c) *If R is a (ρ,σ)-bisimulation then so is $\mathcal{B}_{(\rho,\sigma)}(R)$.*
   (d) *$\sqsubseteq_{(\rho,\sigma)} = \bigcup\{R \mid R \subseteq \mathcal{B}_{(\rho,\sigma)}(R)\}$ is the largest (under set containment) fixpoint of $\mathcal{B}_{(\rho,\sigma)}$.*
4. *$p \sqsubseteq_{(\rho,\sigma)} q$ iff pRq for some $R \in \mathsf{B}_{(\rho,\sigma)}$.*

$\square$

A consequence of proposition 2 is the following theorem which yields an algebraic sufficiency condition for a bisimilarity to be at least a preorder (reflexive and transitive). The notion of a semiring is taken from [7]. The last part of the theorem follows from the second part and the definition of a semiring.

Theorem 3. *Let $\wp(\mathbf{P}\times\mathbf{P})$ be the set of all binary relations on processes. Then*

1. *$\langle \mathsf{B}_{(\rho,\sigma)}, \cup, \emptyset\rangle$ is a commutative submonoid of $\langle \wp(\mathbf{P}\times\mathbf{P}), \cup, \emptyset\rangle$.*
2. *$\sqsubseteq_{(\rho,\sigma)}$ is a preorder if $\langle \mathsf{B}_{(\rho,\sigma)}, \circ, \equiv\rangle$ is a submonoid of $\langle \wp(\mathbf{P}\times\mathbf{P}), \circ, \equiv\rangle$.*
3. *$\sqsubseteq_{(\rho,\sigma)}$ is a preorder if $\langle \mathsf{B}_{(\rho,\sigma)}, \cup, \circ, \emptyset, \equiv\rangle$ is a sub-semiring of $\langle \wp(\mathbf{P}\times\mathbf{P}), \cup, \circ, \emptyset, \equiv\rangle$.*

$\square$

The following simple lemma shows that binary relations (on actions) with certain properties transmit these properties to the bisimulations and bisimilarities they induce.

Lemma 4. (Transmission). *Let ρ, ρ', σ and σ' be binary relations on Act. Then*

1. **Monotonicity_1.** *$(\rho,\sigma) \subseteq (\rho',\sigma')$ implies $\mathsf{B}_{(\rho,\sigma)} \subseteq \mathsf{B}_{(\rho',\sigma')}$, i.e. every (ρ,σ)-bisimulation is also a (ρ',σ')-bisimulation.*
2. **Monotonicity_2.** *$(\rho,\sigma) \subseteq (\rho',\sigma')$ implies the induced bisimilarities are also similarly related[2], that is, $(\rho,\sigma) \subseteq (\rho',\sigma')$ implies $\sqsubseteq_{(\rho,\sigma)} \subseteq \sqsubseteq_{(\rho',\sigma')}$.*
3. **Reflexivity.** *If ρ and σ are both reflexive then the identity relation $\equiv$ on $\mathbf{P}$ is a (ρ,σ)-bisimulation and consequently $\sqsubseteq_{(\rho,\sigma)}$ is reflexive.*
4. **Symmetry.** *ρ and σ are both symmetric implies the converse of each (ρ,σ)-bisimulation is a (σ,ρ)-bisimulation. In addition, if $\rho = \sigma$ then $\sqsubseteq_{(\rho,\sigma)}$ is a symmetric relation.*
5. **Transitivity.** *If ρ and σ are both transitive then the relational composition of (ρ,σ)-bisimulations is another (ρ,σ)-bisimulation, and $\sqsubseteq_{(\rho,\sigma)}$ is also transitive.*
6. *If ρ and σ are both preorders or partial orders then $\sqsubseteq_{(\rho,\sigma)}$ is a preorder[3].*
7. *If ρ is an equivalence relation then so is $\sqsubseteq_{(\rho,\rho)}$.*

$\square$

When $\rho = \sigma$, the converses of the properties 1, 2, 3, 4 and 5 in lemma 4 hold, yielding an obvious characterization (theorem 6). For any binary relation ρ on Act let $\mathsf{B}_{(\rho,\rho)}$ be the family of (ρ,ρ)-bisimulations.

Lemma 5.

1. **Monotonicity.** *If $\mathsf{B}_{(\rho,\rho)} \subseteq \mathsf{B}_{(\rho',\rho')}$ then $\rho \subseteq \rho'$. Similarly, if $\sqsubseteq_{(\rho,\rho)} \subseteq \sqsubseteq_{(\rho',\rho')}$ then $\rho \subseteq \rho'$.*
2. **Reflexivity.** *Let $\equiv$ be the identity relation on processes. Then $\equiv \in \mathsf{B}_{(\rho,\rho)}$ implies ρ is reflexive. Also if $\equiv \subseteq \sqsubseteq_{(\rho,\rho)}$ then ρ must be reflexive.*
3. **Symmetry.** *If $R \in \mathsf{B}_{(\rho,\rho)}$ implies $R^{-1} \in \mathsf{B}_{(\rho,\rho)}$ then ρ must be symmetric. Similarly the symmetry of $\sqsubseteq_{(\rho,\rho)}$ implies ρ must be symmetric.*

[2] However, $(\rho,\sigma) \subset (\rho',\sigma')$ does not imply $\sqsubseteq_{(\rho,\sigma)} \subset \sqsubseteq_{(\rho',\sigma')}$.

[3] However, (ρ,σ)-bisimilarity is not necessarily a partial order if ρ and σ are both partial orders.

4. **Transitivity.** *Let $\circ$ denote relational composition. If $R, S \in \mathsf{B}_{(\rho,\rho)}$ implies $R \circ S \in \mathsf{B}_{(}\rho, \rho)$, then ρ is transitive. In other words, if $\langle \mathsf{B}_{(\rho,\rho)}, \circ \rangle$ is a semigroup then ρ is transitive. Further, if $\sqsubseteq_{(\rho,\rho)}$ is transitive then so is ρ.*

$\square$

Theorem 6. *For any binary relation ρ on Act, $\sqsubseteq_{(\rho,\rho)}$ is a preorder iff ρ is a preorder, and $\sqsubseteq_{(\rho,\rho)}$ is an equivalence iff ρ is an equivalence relation.*

$\square$

3 Example: A caching proxy server

In this section, we show by an example that it is useful to have a more generalized notion of bisimulation than the classical ones.

In most computing environments, it is fairly common to find a caching proxy server in operation. Caching proxy servers improve the performance of web-access within the network by caching most frequently accessed pages of a web-server with predominantly static content and serving them to the clients in the network. The main communication overhead is restricted to receiving header information from the web-site being accessed to determine whether the cached copy in the proxy is the latest or needs to be updated. Caching proxies also improve the performance of the web-server by reducing the number of direct accesses to the web-site from distant clients for its content.

We model greatly simplified versions of the web-server, clients and the proxy server, to show that the use of the proxy server reduces the volume of traffic between the network and the web-server while still serving up to date information to each client.

Assume all the clients in a local area network are identical in design. It will be easy to show that it suffices to model a single client which repeatedly accesses either directly or indirectly (via a proxy server) a single distant web-server with state of the form (h, a) where h is the header (timestamp) and a the content. In general, the volume of data a is much greater than that of the header (h).

The following set of actions are used in this example, written in CCS-style. Process states are written in SMALL-CAPITALS.

$gp(a)$	– *g*et *p*age a,	$op(a)$	– *o*utput *p*age a on screen
$drp()$	– *d*irect *r*equest for *p*age,	$dsp(h, a)$	– *d*irectly *s*erve *p*age
$irp()$	– *i*ndirect *r*equest for *p*age,	$isp(h, a)$	– *i*ndirectly *s*erve *p*age
$drh()$	– *d*irect *r*equest for *h*eader,	$dsh(h)$	– *d*irectly *s*erve *h*eader

We first design the Web server with the assumption that it can service only one request at a time. For a page request, it sends its current page and for a header request, it determines whether to reply with its current header or send the latest page. Sending back the header is an acknowledgement that the page has not be updated since.

$$\begin{array}{lcl}
\textsc{WebServer}(h,a) & \stackrel{\mathrm{df}}{=} & \tau.\textsc{WebServer}(h',a') + \\
 & & drp().\overline{dsp(h,a)}.\textsc{WebServer}(h,a) + \\
 & & drh(h'').\textsc{CheckUpdate}(h,a,h'') \\
\textsc{CheckUpdate}(h,a,h'') & \stackrel{\mathrm{df}}{=} & \textbf{if } h = h'' \textbf{ then } \overline{dsh(h)}.\textsc{WebServer}(h,a) \\
 & & \textbf{else } \overline{dsp(h)}.\textsc{WebServer}(h,a)
\end{array}$$

In the absence of a proxy server, a typical client DCLIENT, which accesses the web-server directly, has the following definition.

$$\textsc{DClient} \stackrel{\mathrm{df}}{=} gp(a).\overline{drp()}.dsp(h,a).\overline{op(a)}.\textsc{DClient} + \tau.\textsc{DClient}$$

With the introduction of a proxy server, the clients communicate only with the proxy and are indeed set up to do just that. In such a system the clients would look as follows. The actions involving communications of the clients with proxy server are irp and isp.

$$\textsc{IClient} \stackrel{\mathrm{df}}{=} gp(a).\overline{irp()}.isp(h,a).\overline{op(a)}.\textsc{IClient} + \tau.\textsc{IClient}$$

We may model the proxy server as follows. Again we simplify the design of the proxy server by assuming it serves only one request at a time and that it has some initial undefined content $(\bot, \bot)$ in its cache. On the first request it obtains the full page from the WEBSERVER. For each subsequent request it merely sends a request with the header as parameter.

$$\begin{array}{lcl}
\text{PROXY0}(\bot,\bot) & \stackrel{\text{df}}{=} & irp().\text{REQUESTPAGE}(\bot,\bot) + \tau.\text{PROXY0}(\bot,\bot) \\
\text{PROXY}(h_0,a_0) & \stackrel{\text{df}}{=} & irp().\text{CLIENTWAIT}(h_0,a_0) + \tau.\text{PROXY}(h_0,a_0) \\
\text{CLIENTWAIT}(h_0,a_0) & \stackrel{\text{df}}{=} & \overline{drh(h_0)}.\text{PROXYWAIT}(h_0,a_0) \\
\text{PROXYWAIT}(h_0,a_0) & \stackrel{\text{df}}{=} & dsh(h_0).\text{CACHED}(h_0,a_0) + dsp(h_0',a_0').\text{CACHED}(h_0',a_0') \\
\text{REQUESTPAGE}(h_0,a_0) & \stackrel{\text{df}}{=} & \overline{drp()}.\text{REQUESTSENT}(h_0,a_0) \\
\text{REQUESTSENT}(h_0,a_0) & \stackrel{\text{df}}{=} & dsp(h_0'',a_0'').\text{CACHED}(h_0'',a_0'') \\
\text{CACHED}(h,a) & \stackrel{\text{df}}{=} & \overline{isp(h,a)}.\text{PROXY}(h,a) \\
 & & \\
\text{CPSYSTEM} & \stackrel{\text{df}}{=} & (\text{ICLIENT}|\text{PROXY0}(\bot,\bot))\backslash\{irp(_,_), isp(_,_)\}
\end{array}$$

where the parameters "_" denote wildcard values.

It is clear that the two systems CPSYSTEM and DCLIENT are not observationally equivalent, since CPSYSTEM may perform actions such as $dsh(\_)$ which are not in the sort of DCLIENT. However they are both functionally equivalent[4] from the point of view of the user sitting at the client and the LAN administration. Moreover, the administration would claim that it is a cheaper solution.

We may compare them as follows. The runs of the two systems may be represented by the following ω-regular expressions (ignoring τ and the parameters on the actions)

DCLIENT: $(gp.\overline{drp}.dsp.\overline{op})^\omega$ CPSYSTEM: $gp.\overline{drp}.dsp.\overline{op}.(gp.\overline{drh}.(dsh + dsp).\overline{op})^\omega$

To prove that the two systems are functionally equivalent we could relate actions which produce "similar effects". In other words, define $=_\rho$ to be the smallest equivalence such that

- $\overline{drh()} =_\rho \overline{drp()}$ and
- $dsh(h) =_\rho dsp(h,a)$, for any (h,a)
- $\varepsilon =_\rho \tau$

We assume that any process p may perform the empty sequence ε, and become itself. Then we may readily see that CPSYSTEM $\sqsubseteq_{(=_\rho,=_\rho)}$ DCLIENT.

However, a more interesting comparison involves using a relation between the costs of functionally equivalent communication actions. The internal action incurs "no cost" since each internal action occurs within the local area network and does not involve communication with any distant entity. Every visible action does have some cost associated with it. However, certain signals such as requests of all kinds are comparable to each other whereas the cost of each response depends upon the payload it carries.

Keeping the above in mind, we define $\leq$ to be the smallest preorder satisfying

- $\overline{drh(h)} \leq \overline{drp()}$ and $\overline{drp()} \leq \overline{drh(h)}$, for any header h
- $dsh(h) \leq dsp(h,a)$, for any (h,a)
- $\varepsilon \leq \tau$, and $\tau \leq \varepsilon$.

It is clear then that CPSYSTEM $\sqsubseteq_{(\leq,\leq)}$ DCLIENT.

4 Cost-based Relations

While we briefly related actions on the basis of cost in section 3, in this section we provide the notion of cost-based bisimulations a more concrete form. In the following sections we transoform some of the existing bisimilarity

[4] i.e one can show that when taken in conjunction with the web-server and hiding the communications involving it, the two systems are in fact, observationally equivalent.

relations in the literature into cost-based ones. Further, in section 3 we used bisimulations based on a single relation; the reason for considering a pair of possibly different relations would become more obvious in section 5.

Let V (ranged over by lower case initial greek letters α, β, $\ldots$) be a countable set of *visible actions*, $\tau \notin V$ a distinguished *invisible action* and $A = V \cup \{\tau\}$ the set of *actions* (ranged over by lower case initial roman letters a, b, $\ldots$). Let A^* denote the set of all finite sequences of actions (including the empty sequence ε). We write $\hat{s}$ to denote the sequence obtained from $s \in A^*$ by deleting all occurrences of τ. $\hat{s} = \varepsilon$ if s contains no visible action. Finally $|s|$ denotes the length of the sequence s. We write $s \mathrel{\hat{=}} t$ if $\hat{s} = \hat{t}$.

Let the set Act of *action-cost pairs* be defined as $Act = V_\varepsilon \times \mathbb{C}$ where $V_\varepsilon = V \cup \{\varepsilon\}$ and $\langle \mathbb{C}, \geq \rangle$ is any set of costs equipped with a reflexive total order $\geq$. For $m, n \in \mathbb{C}$, $n > m$ i.e. $n \neq m$ and $n \geq m$, implies n is a higher cost than m.

Let $\simeq$ (called *visible equivalence*) be the equivalence on Act which ignores the second component of each pair, i.e. $(a, m) \simeq (b, n)$ iff $a = b$. We may extend this relation pointwise to sequences $s \in Act^*$, and refer to *(point-wise) visibly equivalent* sequences. Clearly, two sequences of unequal lengths cannot possibly be equivalent.

The $\geq$ relation on $\mathbb{C}$ is extended to Act so that visibly equivalent action-cost pairs may be compared on cost i.e. $(a, m) \geq (b, n)$ iff $a = b$ and $m \geq n$. We then say that (a, m) is *at least as expensive as* (a, n). It is also similarly extended pointwise to visibly equivalent sequences and we say that a sequence s is *(point-wise) at least as expensive as* t if $s, t \in Act^*$ and $s \geq t$. It is clear that $\simeq$ is a strictly coarser relation than $\geq$ on Act and (by pointwise extension) Act^* whenever $|\mathbb{C}| > 1$.

In the sequel, we will identify $\langle \mathbb{C}, \geq \rangle$ with the set of natural numbers $\mathbb{N}$ under the usual $\geq$ relation. It is possible to consider the non-negative rationals or reals instead. However, while they are intuitively feasible and meaningful, they would not be relevant to the rest of this paper.

5 Temporal Preorders for Interleaving

In this section we revisit the preorders of [1] and [3] and provide fresh characterizations. The setting is a LTS $\langle \mathbf{P}, A, \rightarrow \rangle$, where $A = V \cup \{\tau\}$ is the set of actions as mentioned in the beginning of section 4. We further assume that $V = \Lambda \cup \bar{\Lambda}$, where Λ and $\bar{\Lambda}$ are disjoint and in bijection under the operation $\bar{\ }$. All the processes we mention in this section are assumed to be drawn from this structure. In general, transition systems for processes in an interleaving model of concurrency such as CCS use a product construction on LTS's such as the following.

Definition 7. Given rooted LTSs $\mathcal{L} = \langle \mathbf{P}, Act, \longrightarrow, p_0 \rangle$ and $\mathcal{M} = \langle \mathbf{Q}, Act, \longrightarrow, q_0 \rangle$, their **inteleaving** is a third LTS $\mathcal{N} = \langle \mathbf{R}, Act, \longrightarrow, r_0 \rangle$ and denoted $\mathcal{N} = \mathcal{L} | \mathcal{M}$ such that

- $r_0 = p_0 | q_0$
- $\mathbf{R} \subseteq \{p|q : p \in \mathbf{P}, q \in \mathbf{Q}\}$ consists of the set of states reachable from r_0, where for each $r = p|q$ for $p \in \mathbf{P}$, $q \in \mathbf{Q}$, its successors are such that
  - $p \xrightarrow{a} p' \Rightarrow p|q \xrightarrow{a} p'|q$ for each $a \in A$,
  - $q \xrightarrow{a} q' \Rightarrow p|q \xrightarrow{a} p|q'$ for each $a \in A$, and
  - $p \xrightarrow{\alpha} p' \wedge q \xrightarrow{\bar{\alpha}} q' \Rightarrow p|q \xrightarrow{\tau} p'|q'$ for each $\alpha \in V$.

Definition 8. Let $\preceq$ be the relation on A^* generated by the inequations $s \preceq s$ and $\tau s \preceq s$, i.e. $\preceq$ is closed under reflexivity, transitivity and substitution under catenation contexts. It is clear that $\tau \preceq \varepsilon$, $s\tau \preceq s$ for all s and that $\preceq$ is antisymmetric. Hence $\preceq$ is a partial order on A^*. For $s, t \in A^*$, let $s \mathrel{\preceq\!\!\cdot}\ t$ if $\hat{s} = \hat{t}$ and $|s| \geq |t|$ and $s \doteq t$ if $s \mathrel{\preceq\!\!\cdot}\ t$ and $t \mathrel{\preceq\!\!\cdot}\ s$.

Some easy consequences of definition 8 are listed below.

Lemma 9. *Let $s, t \in A^*$. Then*

1. *s and t are comparable (by any of the preorders/equivalences defined above) only if $\hat{s} = \hat{t}$.*
2. *$s = t$ iff $s \preceq t$ and $t \preceq s$.*
3. *$\preceq\!\cdot$ is coarser than $\preceq$ (i.e. $s \preceq t$ implies $s \preceq\!\cdot\; t$ but not the converse).*
4. *$\widehat{=}$ is a strictly coarser relation than $\doteq$, which in turn is coarser than $=$.*
5. *For any $a \in V$, $a \preceq\; t$ implies $t = a$ and $a \preceq\!\cdot\; t$ implies $t = a$,*
6. *$\tau^i \preceq \tau^j$ iff $i \geq j$ and $\tau^i \preceq\!\cdot\; \tau^j$ iff $i \geq j$.*

$\square$

Definition 10. For $s, t \in A^*$ and $a \in A$, the transitions $p \xrightarrow{s} p'$ and $p \xRightarrow{s} p'$ are defined by induction on the length of s as follows

- $p \xrightarrow{\varepsilon} p$ for all p,
- $p \xrightarrow{s} p'$ for $s = ta$ iff $\exists p'' : p \xrightarrow{t} p'' \xrightarrow{a} p'$,
- $p \xRightarrow{\varepsilon} p'$ iff $\exists m \geq 0 : p \xrightarrow{\tau^m} p'$,
- $p \xRightarrow{a} p'$ iff $\exists p'', p''' : p \xRightarrow{\varepsilon} p'' \xrightarrow{a} p''' \xRightarrow{\varepsilon} p'$, and
- $p \xRightarrow{s} p'$ for $s = ta$ iff $\exists p'' : p \xRightarrow{t} p'' \xRightarrow{a} p'$.

We will be particularly interested in the set of *extended* actions defined by $EA = \{u \in A^* \mid |\hat{u}| \leq 1\}$, viz. the set of sequences which contain at most one visible action. Any extended action $u \in EA$ is of the form τ^i or $\tau^i \alpha \tau^j$, where $\alpha \in V$ and i, j are natural numbers.

In view of definition 10, it is possible to view an LTS $\langle \mathbf{P}, A, \rightarrow \rangle$ as one of the form $\langle \mathbf{P}, A^*, \rightarrow \rangle$ or $\langle \mathbf{P}, EA, \rightarrow \rangle$, where the action set depends upon our viewpoint.

We now have a variety of distinct preorders (and equivalences) on sequences of actions. Consider the following definitions taken from the literature [10], [1], [3].

Definition 11. A binary relation R on the states of a LTS $\langle \mathbb{P}, A, \rightarrow \rangle$, is

1. a *strong bisimulation* if pRq implies for every $a \in A$, (a) $p \xrightarrow{a} p' \Rightarrow \exists q' : q \xrightarrow{a} q' \wedge p'Rq'$, and (b) $q \xrightarrow{a} q' \Rightarrow \exists p' : p \xrightarrow{a} p' \wedge p'Rq'$.
2. an *efficiency prebisimulation* if pRq implies for every $u, v \in EA$, (a) $p \xrightarrow{u} p' \Rightarrow \exists v, q' : u \preceq v \wedge q \xrightarrow{v} q' \wedge p'Rq'$, and (b) $q \xrightarrow{v} q' \Rightarrow \exists u, p' : u \preceq v \wedge p \xrightarrow{u} p' \wedge p'Rq'$.
3. an *elaboration* if pRq implies for every $a \in A$, (a) $p \xrightarrow{a} p' \Rightarrow \exists q' : q \xRightarrow{\hat{a}} q' \wedge p'Rq'$, and (b) $q \xrightarrow{a} q' \Rightarrow \exists p' : p \xRightarrow{a} p' \wedge p'Rq'$.
4. a *weak bisimulation* if pRq implies for every $a \in A$ (a) $p \xrightarrow{a} p' \Rightarrow \exists q' : q \xRightarrow{\hat{a}} q' \wedge p'Rq'$ and (b) $q \xrightarrow{a} q' \Rightarrow \exists p' : p \xRightarrow{\hat{a}} p' \wedge p'Rq'$.

Definition 11 and the proof of the following characterization (Proposition 12) may be found in [1, 3] or are easy consequences of propositions in them. Furthermore, it has been shown that the largest efficiency prebisimulation and the largest elaboration are both preorders. We will refer to them in general as efficiency-based preorders.

Proposition 12. *A binary relation R on the states of a LTS, is*

1. *a* strong bisimulation *iff pRq implies for every $s \in A^*$, (a) $p \xrightarrow{s} p' \Rightarrow \exists q' : q \xrightarrow{s} q' \wedge p'Rq'$ and (b) $q \xrightarrow{s} q' \Rightarrow \exists p' : p \xrightarrow{s} p' \wedge p'Rq'$.*
2. *an efficiency prebisimulation iff pRq implies for every $s, t \in A^*$, (a) $p \xrightarrow{s} p' \Rightarrow \exists q', t : s \preceq t \wedge q \xrightarrow{t} q' \wedge p'Rq'$, and (b) $q \xrightarrow{t} q' \Rightarrow \exists p', s : s \preceq t \wedge p \xrightarrow{s} p' \wedge p'Rq'$.*
3. *an elaboration iff pRq implies for every $s, t \in A^*$, (a) $p \xrightarrow{s} p' \Rightarrow \exists q', t : s \,\widehat{=}\, t \wedge q \xrightarrow{t} q' \wedge p'Rq'$, and (b) $q \xrightarrow{t} q' \Rightarrow \exists p', s : s \preceq t \wedge p \xrightarrow{s} p' \wedge p'Rq'$.*
4. *a* weak bisimulation *iff pRq implies for every $s, t \in A^*$, (a) $p \xrightarrow{s} p' \Rightarrow \exists q', t : \hat{s} = \hat{t} \wedge q \xrightarrow{t} q' \wedge p'Rq'$, and (b) $q \xrightarrow{t} q' \Rightarrow \exists p', s : \hat{s} = \hat{t} \wedge p \xrightarrow{s} p' \wedge p'Rq'$.*

□

It is obvious from proposition 12 that the above-mentioned bisimulations are induced by the various preorders and equivalences defined earlier, on the set of actions. In other words, the following facts are easy consequences of proposition 12.

Facts 13.

1. *A strong bisimulation is exactly a natural bisimulation.*
2. *An efficiency prebisimulation is exactly a* $(\preceq, \preceq)$*-bisimulation.*
3. *An elaboration is exactly a* $(\,\widehat{=}\,, \preceq)$*-bisimulation.*
4. *A weak bisimulation is exactly a* $(\,\widehat{=}\,, \,\widehat{=}\,)$*-bisimulation.*

□

It is interesting to note that even though $\preceq$ is a partial order, the bisimilarity $\sqsubseteq_{(\preceq,\preceq)}$ is only a preorder. However, we shall see in the sequel that there exist further characterizations of the efficiency preorders and we will be using these characterizations to relate the preorders of the previous section to the preorders defined here.

Lemma 14. *Let* $\alpha \in V$ *be any visible action and* i*,* j*,* m*,* n *be naturals. Then*

1. $\tau^i\alpha\tau^j \preceq \tau^m\alpha\tau^n$ *iff* $i \geq m$ *and* $j \geq n$*.*
2. $\tau^i\alpha\tau^j \preceq\!\cdot\; \tau^m\alpha\tau^n$ *iff* $i + j \geq m + n$*.*
3. $\tau^i\alpha\tau^j \doteq \tau^m\alpha\tau^n$ *iff* $i + j = m + n$*.*

□

Lemma 15. *Let* R *be a binary relation on process states. The following are equivalent for all* $\langle p, q\rangle \in R$*.*

1. *For all* $a \in A$*,* $p \xrightarrow{a} p' \Rightarrow \exists q' : (q \xrightarrow{\hat{a}} q' \vee q \xrightarrow{a} q') \wedge p'Rq'$*.*
2. *For all* $u \in EA$*,* $p \xrightarrow{u} p' \Rightarrow \exists v, q' : u \preceq v \wedge q \xrightarrow{v} q' \wedge p'Rq'$*.*
3. *For all* $u \in EA$*,* $p \xrightarrow{u} p' \Rightarrow \exists v, q' : u \preceq\!\cdot\; v \wedge q \xrightarrow{v} q' \wedge p'Rq'$*.*

□

Lemma 15 (whose proof is outlined in the appendix) and lemma 17 are used in the characterization theorem 19. In particular, we draw the reader's attention to the statements 2 and 3 in lemma 15. We notice that within the scope of the universal quantifier "*For all* u", it is possible to substitute a coarser preorder viz. $\preceq\!\cdot$ to achieve the same effects as $\preceq$.

Corollary 16. *Each of the following statements is equivalent to each of the statements in lemma 15, for all* $\langle p, q\rangle \in R$*, where* R *is a binary relation on processes.*

1. *For all* $s \in A^*$*,* $p \xrightarrow{s} p' \Rightarrow \exists t, q' : s \preceq t \wedge q \xrightarrow{t} q' \wedge p'Rq'$*.*
2. *For all* $s \in A^*$*,* $p \xrightarrow{s} p' \Rightarrow \exists t, q' : s \preceq\!\cdot\, t \wedge q \xrightarrow{t} q' \wedge p'Rq'$*.*

□

Lemma 17. *Let* R *be a binary relation on process states. The following are equivalent for all* $\langle p, q\rangle \in R$*.*

1. *For all* $a \in A$*,* $q \xrightarrow{a} q' \Rightarrow \exists p' : p \xRightarrow{a} p' \wedge p'Rq'$*.*
2. *For all* $v \in EA$*,* $q \xrightarrow{v} q' \Rightarrow \exists u, p' : u \preceq v \wedge p \xrightarrow{u} p' \wedge p'Rq'$*.*
3. *For all* $v \in EA$*,* $q \xrightarrow{v} q' \Rightarrow \exists u, p' : u \preceq\!\cdot\; v \wedge p \xrightarrow{u} p' \wedge p'Rq'$*.*
4. *For all* $t \in A^*$*,* $q \xrightarrow{t} q' \Rightarrow \exists s, p' : s \preceq t \wedge p \xrightarrow{s} p' \wedge p'Rq'$*.*
5. *For all* $t \in A^*$*,* $q \xrightarrow{t} q' \Rightarrow \exists s, p' : s \preceq\!\cdot\, t \wedge p \xrightarrow{s} q' \wedge p'Rq'$*.*

□

For the sake of completeness, we state without proof the following lemma (see also [3]).

Lemma 18. *Let R be a binary relation on process states. The following are equivalent for all $\langle p, q\rangle \in R$.*

1. *For all $a \in A$, $p \xrightarrow{a} p' \Rightarrow \exists q' : q \xRightarrow{\hat{a}} q' \wedge p'Rq'$.*
2. *For all $u \in EA$, $p \xrightarrow{u} p' \Rightarrow \exists v, q' : u \mathrel{\hat{=}} v \wedge q \xrightarrow{v} q' \wedge p'Rq'$.*
3. *For all $s \in A^*$, $p \xrightarrow{s} p' \Rightarrow \exists t, q' : s \mathrel{\hat{=}} t \wedge q \xrightarrow{t} q' \wedge p'Rq'$.*

$\square$

We then have the following characterization (theorem 19) of the efficiency-based preorders. This theorem follows from lemma 15, corollary 16 and lemma 17.

Theorem 19. **The characterization theorem.** *A binary relation R on the states of an LTS $\langle \mathbf{P}, A, \rightarrow\rangle$ is*

1. *an efficiency prebisimulation iff every $\langle p, q\rangle \in R$ satisfies any of the statements in lemma 15, corollary 16 and any of the statements in lemma 17.*
2. *an elaboration iff every $\langle p, q\rangle \in R$ satisfies any of the statements in lemma 18 and any of the statements in lemma 17.*

$\square$

More succinctly we have the following corollary.

Corollary 20. *A binary relation R on the states of an LTS $\langle \mathbf{P}, A, \rightarrow\rangle$ is*

1. *an efficiency prebisimulation iff it is a $(\preceq\!\cdot\,, \preceq\!\cdot\,)$-bisimulation,*
2. *an elaboration iff it is a $(\,\hat{=}\,, \preceq\!\cdot\,)$-bisimulation,*

$\square$

Note that even though $(\preceq, \preceq) \subset (\preceq\!\cdot\,, \preceq\!\cdot\,)$, the respective bisimilarities induced by them are the same. A similar remark applies to the bisimilarities induced by the preorders $(\,\hat{=}\,, \preceq)$ and $(\,\hat{=}\,, \preceq\!\cdot\,)$.

5.1 Transforming the Preorders

Let $h : EA \rightarrow Act$ be the function defined by

$$h(\tau^i) = (\varepsilon, i) \qquad\qquad h(\tau^i\alpha\tau^j) = (\alpha, i + j)$$

h induces an equivalence (denoted $\doteq$) on EA. Our choice of h is motivated by corollary 20 and the following consideration, stated without proof.

Lemma 21. *$EA/\doteq$ is isomorphic to Act. In other words, for all $u, v \in EA$, $u \doteq v$ iff $h(u) = h(v)$.*

$\square$

We may use this function to transform an LTS $\mathcal{L} = \langle \mathbf{P}, \mathbf{EA}, \rightarrow\rangle$ into an LTS $h(\mathcal{L}) = \langle \mathbf{P}, \mathbf{Act}, \rightarrow\rangle$ such that for any $p, p' \in \mathbf{P}$, and $u \in EA$, $p \xrightarrow{u} p'$ if and only if $p \xrightarrow{h(u)} p'$. We then have the following transformation theorem.

Theorem 22. **The Transformation Theorem.** *Let $\mathcal{L} = \langle \mathbf{P}, EA, \rightarrow\rangle$ be a LTS. A binary relation $R \subseteq \mathbf{P} \times \mathbf{P}$ is*

1. *a weak bisimulation in $\mathcal{L}$ iff it is a $(\,\hat{-}\,, \,\hat{-}\,)$-bisimulation in $h(\mathcal{L})$,*
2. *an elaboration in $\mathcal{L}$ iff it is a $(\,\hat{-}\,, \geq)$-bisimulation in $h(\mathcal{L})$ and*
3. *an efficiency prebisimulation in $\mathcal{L}$ iff it is a $(\geq, \geq)$-bisimulation in $h(\mathcal{L})$.*

$\square$

Originally, the efficiency-based preorders were defined for languages such as CCS. The transformation theorem enables us to treat the number of invisible actions as a form of cost and reason about performance in an abstract fashion. Even in the original works ([2, 3]) the invisible action represented a form of abstract unit cost viz. the cost of synchronization, or the cost of idling, without actually quantifying it. The realization that in any extended action $\tau^i\alpha\tau^j$ what matters is only the sum $i + j$ and not the individual quantities i and j allows a quantification on the cost that yields a more intuitive mechanism via the transformation than was originally proposed in [2, 3].

6 Spatial Preorders for Synchronous Systems

In this section we first introduce multisets and then give the intended action structure using them. Our motivation for using multisets rather than the group structure proposed by Milner [9] stems from the fact that in hardware among the factors which often contribute to cost are

- the number of connections in the layout of a chip, and
- the heat disspated or the power consumed are directly related to the number of operations being performed simultaneously.

We relate our notions to Milner's group-based action structure ([9]) in our transformation theorem. One of the major disadvantages of Milner's action structure is that it does not contain sufficient information for comparison of synchronous hardware on various criteria like the ones stated above. We may for example, directly relate the heat dissipated to the number of signals passing through various wires which form the connections in the circuit. It is therefore more appropriate to allow for more information to be gleaned from the action structure. We define a synchronous version (spatial version) of the efficiency preorders previously defined for the temporal preorders and show that they satisfy some very nice properties. We also show that Milner's bisimilarity relation may be recovered from our spatial preorders by suitably mapping the multi-set action structure to Milner's group structure.

In this context it is important to realise that while most of our results on multisets are very general, we shall be mostly interested in finite multisets. Our view of a multiset of signals is that all the signals in the multiset occur simultaneously. It is unreasonable therefore to assume that a finite (circuit) structure is capable of communicating an infinite number of signals in a single instant of time. The power requirement for this would be infinite. Also it would enable an infinite amount of computation to be performed in a finite time — a proposition that is impossible under any reasonable model of computation.

6.1 The Multiset action structure

Definition 23. A **multiset** A on an universe U is a function $A : U \to \mathbb{N}$. A is called **finite** if it is almost everywhere 0. For any, $a \in U$, $A(a)$ is the **multiplicity** of the element a. We say $a \notin A$ if $A(a) = 0$. The set of all elements of A with non-zero multiplicities is called the **support** of the multiset and denoted $\S A$. The **empty** multiset, denoted $\emptyset$, is one which has no elements.

Notation: A finite multiset A is often written by enumerating its elements — the number of copies of each element equals its multiplicity in A. A multiset A with $A(a) = 3$ and $A(b) = 2$ and $\forall x \in U \setminus \{a, b\} : A(x) = 0$, is a finite multiset and may be written either by enumeration (e.g. $[a, a, a, b, b]$) or more simply as a^3b^2. Its support is the set $\S A = \{a, b\}$.

Let A, B and C be multisets on an universe U. We then define the following operations and relations on multisets.

Union:	$C = A \cup B$ where $\forall x \in U : C(x) = max(A(x), B(x))$
Intersection:	$C = A \cap B$ where $\forall x \in U : C(x) = min(A(x), B(x))$
Sum:	$C = A + B$ where $\forall x \in U : C(x) = A(x) + B(x)$
Difference:	$C = A - B$ where $\forall x \in U : C(x) = max(0, A(x) - B(x))$

Submultiset:	$A \subseteq B$	if $\forall x \in U : A(x) \leq B(x)$
Proper submultiset:	$A \subset B$	if $A \subseteq B$ and $A \neq B$
Cardinality:	$\mid A \mid \; = \; \mid \S A \mid$	
Size:	$\parallel A \parallel \; = \; \sum\{A(x) : x \in U\}$	

Definition 24. Let
- Λ be a countable set of names,

- $\overline{\Lambda}$ the set of **co-names** disjoint from Λ and in bijection with it,
- $\mathsf{Vis} = \Lambda \cup \overline{\Lambda}$ the set of **visible signals**,
- $\mathsf{1}$ the distinguished **invisible** signal not occurring in Vis and
- $\mathsf{Sig} = \mathsf{Vis} \cup \{\mathsf{1}\}$ the set of all **signals**.

The bijection between Λ and $\overline{\Lambda}$ may be extended to a complement operation $\overline{} : \mathsf{Sig} \to \mathsf{Sig}$ such that $\overline{\mathsf{1}} = \mathsf{1}$ and for every $\alpha \in \mathsf{Sig}$, $\overline{\overline{\alpha}} = \alpha$, and α and $\overline{\alpha}$ are a **complementary pair**.

Definition 25. An **action** A is a finite multiset of signals satisfying the *exclusivity* property: $\forall \alpha \in \mathsf{Vis} : \mathrm{A}(\alpha) = 0 \vee \mathrm{A}(\overline{\alpha}) = 0$. ACT_0 is the set of all actions. The **visible content** $\underline{\mathrm{A}}$ of an action A is the action $\underline{\mathrm{A}} \subseteq \mathrm{A}$ such that $\underline{\mathrm{A}}(\mathsf{1}) = 0$ and $\underline{\mathrm{A}}(\alpha) = \mathrm{A}(\alpha)$ for all $\alpha \in \mathsf{Vis}$. An action is said to be **invisible** iff $\underline{\mathrm{A}} = \emptyset$.

The reader may wonder why we have specified an exclusivity condition in our definition of an action. Our notion of an action here represents a collection of signals that occur simultaneously in time (often this means at the leading edge or the trailing edge of a clock). In this calculus (as in many others) naming is used to specify connections between various ports. A wire or a direct connection is represented by a pair of complementary signals. Such a synchronized pair of actions yields a single indivisible signal viz. $\mathsf{1}$. Hence in the specification of open systems it is desirable to specify only whatever signals are visible to an external observer who might experiment with a component. In cases where a signal needs to be fanned out we will use a form of renaming which allows an output signal to be split into two or more signals and then synchronize with appropriately named complementary input signals. Hence our definition of an action includes the exclusivity condition as a pre-requisite.

Note: An invisible action A is often denoted by the natural number i, where $i = \mathrm{A}(\mathsf{1})$. The empty action $\emptyset$ is denoted $\mathsf{0}$. The set of invisible actions is thus identified with the set $\mathbb{N}$ of natural numbers. An action $\mathrm{A} = \underline{\mathrm{A}} + i$ is sometimes written $\mathrm{A} = \langle \underline{\mathrm{A}}, i \rangle$. Further, for any $\mathrm{A} = \langle \underline{\mathrm{A}}, i \rangle$, $\mathrm{A} \neq \emptyset$ implies $\underline{\mathrm{A}} \neq \emptyset \vee i \neq 0$. $\mathrm{ACT} = \mathrm{ACT}_0 - \{\mathsf{0}\}$ is the set of **nonempty** actions. $\underline{\mathrm{ACT}} = \{\langle \underline{\mathrm{A}}, 0 \rangle | \mathrm{A} \in \mathrm{ACT}\}$ is the set of **pure** visible actions.

Definition 26. The following operations are defined on actions.

- $\overline{\mathrm{A}}$ the *complement* of action A is the point-wise complement of A, i.e. $\overline{\mathrm{A}}(a) = \mathrm{A}(\overline{a})$ for all $a \in \mathsf{Sig}$. For any action A, $\underline{\overline{\mathrm{A}}}$ denotes the visible content of $\overline{\mathrm{A}}$.
- The *composition* of two actions $\mathrm{A} = \langle \underline{\mathrm{A}}, i \rangle$ and $\mathrm{B} = \langle \underline{\mathrm{B}}, j \rangle$ is another action denoted $\mathrm{A} \otimes \mathrm{B}$ and defined as

$$\mathrm{A} \otimes \mathrm{B} = \langle (\underline{\mathrm{A}} - \underline{\overline{\mathrm{B}}}) + (\underline{\mathrm{B}} - \underline{\overline{\mathrm{A}}}), \parallel \underline{\mathrm{A}} \cap \underline{\overline{\mathrm{B}}} \parallel + i + j \rangle$$

The following facts are easy consequences of the definitions above.

Facts 27. *Let* $\mathrm{A} = \langle \underline{\mathrm{A}}, i \rangle$, $\mathrm{B} = \langle \underline{\mathrm{B}}, j \rangle$ *and* $\mathrm{C} = \langle \underline{\mathrm{C}}, k \rangle$ *be actions.*

1. $\forall \alpha \in V : (\mathrm{A} \otimes \mathrm{B})(\alpha) = ((\mathrm{A} + \mathrm{B}) - (\overline{\mathrm{A}} + \overline{\mathrm{B}}))(\alpha)$.
2. $\underline{\mathrm{A} \otimes \mathrm{B}} = (\mathrm{A} + \mathrm{B}) - (\overline{\mathrm{A}} + \overline{\mathrm{B}})$.
3. $\mathrm{A} \otimes \mathrm{B} = \mathrm{B} \otimes \mathrm{A}$.
4. $((\mathrm{A} \otimes \mathrm{B}) \otimes \mathrm{C})(\alpha) = ((\mathrm{A} + \mathrm{B} + \mathrm{C}) - (\overline{\mathrm{A}} + \overline{\mathrm{B}} + \overline{\mathrm{C}}))(\alpha)$*, for all* $\alpha \in V$.
5. $(\mathrm{A} \otimes \mathrm{B})(1) = \|\underline{\mathrm{A}} \cap \underline{\overline{\mathrm{B}}}\| + i + j$.
6. $((\mathrm{A} \otimes \mathrm{B}) \otimes \mathrm{C})(1) = i + j + k + \sum_{\alpha \in \Lambda} min(\mathrm{A}(\alpha) + \mathrm{B}(\alpha) + \mathrm{C}(\alpha), \mathrm{A}(\overline{\alpha}) + \mathrm{B}(\overline{\alpha}) + \mathrm{C}(\overline{\alpha}))$
7. $\otimes$ *is associative.*
8. ACT_0 *is an abelian monoid under* $\otimes$*, with* $\mathsf{0}$ *as the identity element.*

Note. Most of the parts in the proposition above may be proven using the condition of exclusivity on actions. In particular, if we remove the finiteness condition the set of multisets over Sig will still remain an abelian monoid under $\otimes$.

6.2 Spatial Preorders

Definition 28. Given an LTS of the form $\mathcal{L} = \langle \mathbf{P}, \textsc{Act}, \longrightarrow, p_0 \rangle$ each process of such an LTS is called a **synchronous** process. The **synchronous product** of two rooted LTSs $\mathcal{L} = \langle \mathbf{P}, \textsc{Act}, \longrightarrow, p_0 \rangle$ and $\mathcal{M} = \langle \mathbf{Q}, \textsc{Act}, \longrightarrow, q_0 \rangle$ is a third LTS $\mathcal{N} = \langle \mathbf{R}, \textsc{Act}, \longrightarrow, r_0 \rangle$ and denoted $\mathcal{N} = \mathcal{L} \otimes \mathcal{M}$ such that

- $r_0 = p_0 \otimes q_0$
- $\mathbf{R} \subseteq \{p \otimes q : p \in \mathbf{P}, q \in \mathbf{Q}\}$ consists of the set of states reachable from r_0, where for each $r = p \otimes q$ for $p \in \mathbf{P}$, $q \in \mathbf{Q}$, its successors are such that
  $p \xrightarrow{\textsc{a}} p' \wedge q \xrightarrow{\textsc{b}} q' \Rightarrow p \otimes q \xrightarrow{\textsc{a}\otimes\textsc{b}} p' \otimes q'$ for each $\textsc{a}, \textsc{b} \in \textsc{Act}$.

In analogy to the preorders defined on processes in section 5 where comparison relations were defined over sequences of actions (the temporal ordering relation) here we compare actions according to their internal structure (spatial).

Definition 29. Let $\textsc{a} = \langle \underline{\textsc{a}}, i \rangle$ and $\textsc{b} = \langle \underline{\textsc{b}}, j \rangle$ be two actions. We say that $\textsc{a}$ is *at least as* **wide** as $\textsc{b}$ denoted $\textsc{a} \trianglelefteq \textsc{b}$ iff $\underline{\textsc{a}} = \underline{\textsc{b}}$ and $i \geq j$. On the other hand $\textsc{a}$ is **visibly equivalent** to $\textsc{b}$ and denoted $\textsc{a} \bowtie \textsc{b}$ if $\underline{\textsc{a}} = \underline{\textsc{b}}$

Facts 30.

1. *$\trianglelefteq$ is a partial order on* $\textsc{Act}$.
2. *$\bowtie$ is an equivalence relation on* $\textsc{Act}$.
3. *Both $\bowtie$ and $\trianglelefteq$ are preserved under $\otimes$, i.e. for any three actions* $\textsc{a}$, $\textsc{b}$ *and* $\textsc{c}$, *and* $\diamond \in \{\trianglelefteq, \bowtie\}$, $\textsc{a} \diamond \textsc{b}$ *implies* $\textsc{a} \otimes \textsc{c} \diamond \textsc{b} \otimes \textsc{c}$ *and* $\textsc{c} \otimes \textsc{a} \diamond \textsc{c} \otimes \textsc{b}$

As in the case of bisimulations induced by the temporal ordering on sequences of actions, we obtain bisimulations based on the spatial orderings $\trianglelefteq$ and $\bowtie$.

Definition 31. A binary relation R on the states of a LTS $\langle \mathbf{P}, \textsc{Act}, \rightarrow \rangle$, is said to be a

1. a *fine strong bisimulation* if pRq implies for every $\textsc{a} \in \textsc{Act}$, (a) $p \xrightarrow{\textsc{a}} p' \Rightarrow \exists q' : q \xrightarrow{\textsc{a}} q' \wedge p'Rq'$ and (b) $q \xrightarrow{\textsc{a}} q' \Rightarrow \exists p' : p \xrightarrow{\textsc{a}} p' \wedge p'Rq'$.
2. a *fine efficiency prebisimulation* if pRq implies for every $\textsc{a} \in \textsc{Act}$, (a) $p \xrightarrow{\textsc{a}} p' \Rightarrow \exists \textsc{b}, q' : \textsc{a} \trianglelefteq \textsc{b} \wedge q \xrightarrow{\textsc{b}} q' \wedge p'Rq'$ and (b) $q \xrightarrow{\textsc{b}} q' \Rightarrow \exists \textsc{a}, p' : \textsc{a} \trianglelefteq \textsc{b} \wedge p \xrightarrow{\textsc{a}} p' \wedge p'Rq'$
3. a *fine elaboration* if pRq implies for every $\textsc{a} \in \textsc{Act}$, (a) $p \xrightarrow{\textsc{a}} p' \Rightarrow \exists \textsc{b}, q' : \underline{\textsc{a}} = \underline{\textsc{b}} \wedge q \xrightarrow{\textsc{b}} q' \wedge p'Rq'$ and (b) $q \xrightarrow{\textsc{b}} q' \Rightarrow \exists \textsc{a}, p' : \textsc{a} \trianglelefteq \textsc{b} \wedge p \xrightarrow{\textsc{a}} p' \wedge p'Rq'$
4. a *fine weak bisimulation* if pRq implies for every $\textsc{a}, \textsc{b} \in \textsc{Act}$, (a) $p \xrightarrow{\textsc{a}} p' \Rightarrow \exists \textsc{b}, q' : \underline{\textsc{a}} = \underline{\textsc{b}} \wedge q \xrightarrow{\textsc{b}} q' \wedge p'Rq'$, and (b) $q \xrightarrow{\textsc{b}} q' \Rightarrow \exists \textsc{a}, p' : \underline{\textsc{a}} = \underline{\textsc{b}} \wedge p \xrightarrow{\textsc{a}} p' \wedge p'Rq'$

Facts 32. *For LTSs on* $\textsc{Act}$

1. *A fine strong bisimulation is exactly a natural bisimulation.*
2. *A fine efficiency prebisimulation is exactly a* $(\trianglelefteq, \trianglelefteq)$*-bisimulation.*
3. *A fine elaboration is exactly a* $(\bowtie, \trianglelefteq)$*-bisimulation.*
4. *A fine weak bisimulation is exactly a* $(\bowtie, \bowtie)$*-bisimulation.*

The case of these spatial preorders/equivalences is much simpler than that of the temporal ones seen before. Notice that

- The representation of each action $\textsc{a}$ in the form $\langle \underline{\textsc{a}}, i \rangle$ is adequate as a transformation of an LTS.
- $\textsc{Act}/\bowtie$ is an abelian group under $\otimes$ and is isomorphic to the group structure of actions in [9].
- The "strong bisimulation" relation defined in [9] is exactly a $(\bowtie, \bowtie)$-bisimulation.

Theorem 33. The Transformation Theorem. *Let $Act = \underline{\text{ACT}} \times \mathbb{N}$ be the set of action-cost pairs and let g : $\text{ACT} \to Act$ be the function $g(\langle \underline{\text{A}}, i \rangle) = (\underline{\text{A}}, i)$ which extended in an obvious fashion transforms any LTS $\mathcal{L} = \langle \mathbf{P}, \text{ACT}, \to \rangle$ into the LTS $g(\mathcal{L}) = \langle \mathbf{P}, Act, \to \rangle$. A binary relation $R \subseteq \mathbf{P} \times \mathbf{P}$ is*

1. *a fine weak bisimulation in $\mathcal{L}$ iff it is a $(\simeq, \simeq)$-bisimulation in $g(\mathcal{L})$,*
2. *a fine elaboration in $\mathcal{L}$ iff it is a $(\simeq, \geq)$-prebisimulation in $g(\mathcal{L})$ and*
3. *a fine efficiency prebisimulation in $\mathcal{L}$ iff it is a $(\geq, \geq)$-prebisimulation in $g(\mathcal{L})$.*

□

7 Conclusion

In the foregoing we have generalized the notion of a bisimulation to one parametrized by a pair of relations. We have shown that the commonly accepted properties of bisimilarities are in fact inherited from the underlying relations on actions.

Using these notions we have defined cost-oriented bisimulations and shown that some of the bisimilarity relations already available in the literature may be mirrored in cost-oriented bisimulation relations by fairly simple transformations on the labelled transition systems.

Our example also indicates that for open systems, weak-bisimilarity (and indeed other extensional equivalence notions that are coarser than it and rely on the *identity* of names of actions) may not be useful enough. Other bisimulation-based relations with more concrete forms of cost may be found in the works of Lüttgen, Vogler, Jenner and Corradini ([8, 4–6]), but they do not raise such a question.

Both our example (section 3) and the later sections indicate that to obtain bisimilarity relations that are intuitively useful, the two parameters ρ and σ should be related. Symmetry considerations (lemma 4.4 and 5.3) for bisimilarities that are equivalences also bear this out.

We believe that cost-based preordering relations have genuine uses in combining semantical issues with issues of complexity in reactive systems which are usually characterized by infinite behaviour. These transformations should allow verification and comparison of systems within a single framework.

In particular in section 4 we have chosen cost to be a natural number. By allowing addition on the naturals, we open up the possibility of considering a cost relation on sequences that is coarser than the point-wise relations we have considered in the latter sections of this paper. For instance, a sequence s could be considered *at least as expensive* as another visibly equivalent sequence t (denoted $s \succeq t$) if the sum of the costs on the actions of s is no less than the corresponding sum in t (even though their costs are not related point-wise). In languages like CCS and CSP one also needs to fix the cost of synchronization as a function of the costs of the individual actions that participate in it. We do believe that such considerations would be useful in many practical systems – for example, finite-state open systems (represented as process graphs) whose runs start from some initial state and progress eventually into a strongly connected component.

However, some problems which have eluded obvious solutions are the following:

1. Theorem 3 only gives a sufficient condition for a (ρ, σ)-bisimilarity to be a preorder. A ncessary condition on the semiring of (ρ, σ)-bisimulations would be desirable, when $\rho \neq \sigma$. More specifically, whereas necessity of reflexivity in the underlying pair of relations may be proven fairly easily, transitivity eludes us.
2. From a purely verification perspective, there exist efficient algorithms, notably that of Paige and Tarjan [11], which partition the state space into equivalence classes and enable the computation of natural bisimilarity. It is not clear what the right generalization for the computation of (ρ, σ)-bisimilarities is.
3. It is also not clear at the moment, what the right generalization would be, to capture other cost-based preorders and equivalences such as those of [8, 4–6].

References

1. S. Arun-Kumar and M. Hennessy. An efficiency preorder for processes. In *Theoretical Aspects of Computer Software, Sendai 1991*, number 526 in Lecture Notes in Computer Science, pages 152–175. Springer-Verlag, 1991.
2. S. Arun-Kumar and M. Hennessy. An efficiency preorder for processes. *Acta Informatica*, 29:737–760, 1992.
3. S. Arun-Kumar and V. Natarajan. Conformance: A precongruence close to bisimilarity. In *STRICT, Berlin 1995*, number 526 in Workshops in Computing Series, pages 55–68. Springer-Verlag, 1995.
4. F. Corradini, R. Gorrieri, and M. Rocetti. Performance preorder and competitive equivalence. *Acta Informatica*, 34:805–835, 1997.
5. F. Corradini, W. Vogler, and L. Jenner. Comparing the worst-case efficiency of asynchronous systems with PAFAS. *Acta Informatica*, 38:735–792, 2002.
6. L. Jenner and W. Vogler. Comparing the efficiency of asynchronous systems. Technical Report 1998-3, Universitat Augsburg, December 1998.
7. W. Kuich and A. Salomaa. *Semirings, Automata, Languages. Volume 5: EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1986.
8. G. Luettgen and W. Vogler. A faster then relation for asynchronous processes. In *Proceedings CONCUR 2001, LNCS 2154, Springer-Verlag*, pages 262 – 276, 2001.
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. R. Paige and R.E. Tarjan. Three partition refinement algorithms. *SIAM Journal of Computing*, 16(6):973–989, December 1987.
12. D. M. R. Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI Conference on Theoretical Computer Science*, volume 104, pages 167–183. Lecture Notes in Computer Science, 1981.

8 Appendix: Proofs of relevant lemmas and theorems

Proofs of Proposition 2

1. Let $R_i, i \in I$ for any indexing set I, be (ρ, σ)-bisimulations and let $R = \bigcup_{i \in I} R_i$. The result follows immediately from the fact that any $\langle p, q \rangle \in R$ belongs to R_i, for some $i \in I$ and $R_i \subseteq R$ and satisfies the conditions 1 and 2
2. Let $\mathcal{B}_{(\rho,\sigma)}$ be a function on binary relations on $\mathbb{P}$.
   (a) Assume $R \subseteq S \subseteq \mathbb{P} \times \mathbb{P}$. Consider any $\langle p, q \rangle \in \mathcal{B}_{(\rho,\sigma)}(R)$. Then for all $a \in Act$, $p \xrightarrow{a} p'$ implies $\exists b, q' : a \rho b \wedge q \xrightarrow{b} q' \wedge p' R q'$ which implies $p' S q'$. Similarly for all $b \in Act$, $q \xrightarrow{b} q'$ implies $\exists a, p' : a \sigma b \wedge p \xrightarrow{a} p' \wedge p' R q'$ which implies $p' S q'$. Hence $\langle p, q \rangle \in \mathcal{B}(S)$.
   (b) ($\Longrightarrow$) Let R be a (ρ, σ)-bisimulation and $\langle p, q \rangle \in R$. Since p and q satisfy the conditions of conditions *rho* and *sigma* it follows that $\langle p, q \rangle \in \mathcal{B}_{(\rho,\sigma)}(R)$. Hence $R \subseteq \mathcal{B}_{(\rho,\sigma)}(R)$.
   ($\Longleftarrow$) Assume $R \subseteq \mathcal{B}_{(\rho,\sigma)}(R)$. Any $\langle p, q \rangle \in R$ also belongs to $\mathcal{B}_{(\rho,\sigma)}(R)$ and satisfies the conditions 1 and 2. Hence R is a (ρ, σ)-bisimulation.
   (c) Since R is a (ρ, σ)-bisimulation it follows that $R \subseteq \mathcal{B}_{(\rho,\sigma)}(R)$ by fact 2.3.3b. Since every $\langle p, q \rangle \in \mathcal{B}_{(\rho,\sigma)}(R)$ satisfies the conditions 1 and 2 and it follows that $\mathcal{B}_{(\rho,\sigma)}(R)$ is (ρ, σ)-bisimulation.
   (d) $\underline{\sqsupset}_{(\rho,\sigma)} = \bigcup \mathcal{R}$ where $\mathcal{R} = \{R \mid R \subseteq \mathcal{B}_{(\rho,\sigma)}(R)\}$. By fact 2.2 $\underline{\sqsupset}_{(\rho,\sigma)}$ is also a (ρ, σ)-bisimulation. Hence by fact 2.3.3b, $\underline{\sqsupset}_{(}\rho, \sigma) \subseteq \mathcal{B}_{(\rho,\sigma)}(\underline{\sqsupset}_{(\rho,\sigma)})$ and by fact 2.3.3c, $\mathcal{B}_{(\rho,\sigma)}(\underline{\sqsupset}_{(}\rho, \sigma)) \subseteq \underline{\sqsupset}_{(\rho,\sigma)}$, showing that $\underline{\sqsupset}_{(\rho,\sigma)}$ is a fixpoint of $\mathcal{B}_{(\rho,\sigma)}$.
   (e) Let $\underline{\sqsupset}$ be any fixpoint of $\mathcal{B}_{(\rho,\sigma)}$ i.e. let $\underline{\sqsupset} = \mathcal{B}_{(\rho,\sigma)}(\underline{\sqsupset})$. Then clearly $\underline{\sqsupset} \subseteq \mathcal{B}(\underline{\sqsupset})$ too and hence $\underline{\sqsupset}$ is a $\rho, \sigma)$-bisimulation and is hence a member of $\mathcal{R}$ which implies $\underline{\sqsupset} \subseteq \underline{\sqsupset}_{(\rho,\sigma)}$.

$\square$

Proof of Lemma 4.5

Proof. Let R and S be (ρ,σ)-bisimulations and let $T = R \circ S$ and let $\langle p, r\rangle \in T$. Then for some q, $pRqSr$. Suppose $p \xrightarrow{a} p'$. Then $\exists b, q' \ : \ a\rho b \wedge q \xrightarrow{b} q' \wedge p'Rq'$. $q \xrightarrow{b} q'$ implies $\exists c, r' \ : \ b\rho c \wedge r \xrightarrow{c} r' \wedge q'Sr'$. Since ρ is a transitive relation $a\rho b\rho c$ implies $a\rho c$ and the conclusion follows from $p \xrightarrow{a} p' \Longrightarrow \exists c,\ r' \ : \ a\rho c \wedge r \xrightarrow{c} r' \wedge p'Tr'$. Similarly if $r \xrightarrow{c} r'$ then $\exists b,\ q' \ : \ b\sigma c \wedge q \xrightarrow{b} q' \wedge q'Sr'$ and $q \xrightarrow{b} q'$ implies $\exists a,\ p' \ : \ a\sigma b \wedge p \xrightarrow{a} p' \wedge p'Rq'$ and since σ is transitive we have $r \xrightarrow{c} r' \Rightarrow \exists a, p' : a\sigma c \wedge p \xrightarrow{a} p' \wedge p'Rr'$. Hence T is a (ρ,σ)-bisimulation. Also as a consequence, we get that $\sqsubseteq_{(\rho,\sigma)}$ is closed under relational composition and hence is a transitive relation.

$\square$

Even though our proofs are not specific to any particular process calculus, we often use the syntax of CCS for defining processes of particular interest in our proofs. In particular, in the proofs of lemma 5 we use the following process definitions to avoid defining them repeatedly:

$$p = a.\mathbf{0} \qquad\qquad q = b.\mathbf{0}$$

Proof of Lemma 5.1.

Proof. If $\rho \not\subseteq \rho'$, there exist $a, b \in Act$ such that $(a, b) \in \rho - \rho'$. This implies there exists no (ρ', ρ')-bisimulation containing the pair $\langle p, q\rangle$.

1. Assume $\mathsf{B}_{(\rho,\rho)} \subseteq \mathsf{B}_{(\rho',\rho')}$. $R = \{\langle p, q\rangle, \langle \mathbf{0}, \mathbf{0}\rangle\}$ is a (ρ,ρ)-bisimulation. However since $(a, b) \notin \rho'$, R is not a (ρ', ρ')-bisimulation, which is a contradiction.
2. Assume $\sqsubseteq_{(\rho,\rho)} \subseteq \sqsubseteq_{(\rho',\rho')}$. Hence $p \sqsubseteq_{(\rho,\rho)} q$ implies $p \sqsubseteq_{(\rho',\rho')} q$, which is a contradiction.

Proof of Lemma 5.2.

Proof. If ρ is not reflexive, there must be some action $a \in Act$, that is not related to itself via ρ. Then $\equiv$ is not a (ρ,ρ)-bisimulation since $\langle p, p\rangle \in \equiv$. In fact, then there exists no (ρ,ρ)-bisimulation containing $\langle p, p\rangle$. This contradicts both $\equiv \in \mathsf{B}_{(\rho,\rho)}$ and $\equiv \subseteq \sqsubseteq_{(\rho,\rho)}$. Hence ρ must be reflexive.

Proof of Lemma 5.3.

Proof. If ρ is not symmetric, there must some a, b such that $a\rho b$ but $(b, a) \notin \rho$. Now $R = \{\langle p, q\rangle, \langle \mathbf{0}, \mathbf{0}\rangle\}$ is a (ρ,ρ)-bisimulation implies so is R^{-1}. But this is impossible since there can be no (ρ,ρ)-bisimulation containing $\langle q, p\rangle$. Similarly for th next part, it follows that since $\sqsubseteq_{(\rho,\rho)}$ is a symmetric relation both R and R^{-1} must be contained in $\sqsubseteq_{(\rho,\rho)}$, which is again impossible.

Proof of Lemma 5.4.

Proof.
1. Assume $\mathsf{B}_{(\rho,\rho)}$ is closed under relational composition, but that ρ itself is not transitive. Then for some actions $a, b, c \in Act$, we have $a\rho b$ and $b\rho c$ but not $a\rho c$. Consider the processes p and q as in the proof of Lemma 1 and a process $r \equiv c.\mathbf{0}$. The relations $R = \{\langle p, q\rangle, \langle \mathbf{0}, \mathbf{0}\rangle\}$ and $S = \{\langle q, r\rangle, \langle \mathbf{0}, \mathbf{0}\rangle\}$ are (ρ,ρ)-bisimulations, But their composition $R \circ S = \{\langle p, r\rangle, \langle \mathbf{0}, \mathbf{0}\rangle\}$ is not.
2. In fact, it may be easily shown that there exists no (ρ,ρ)-bisimulation containing the pair $\langle p, r\rangle$. Hence if $\sqsubseteq_{(\rho,\rho)}$ is a transitive relation, then $p \sqsubseteq_{(\rho,\rho)} q \sqsubseteq_{(\rho,\rho)} r$ implies $p \sqsubseteq_{(\rho,\rho)} r$ which is impossible.

Proof of lemma 15.

We prove the following cycle of implications, viz. $(15.3) \Rightarrow (15.1) \Rightarrow (15.2) \Rightarrow (15.3)$.

$(15.3) \Rightarrow (15.1)$ follows by taking $u = a$ and $v = a$ or $v = \hat{a}$. In either case $u \preceq v$. Similarly $(15.2) \Rightarrow (15.3)$ is obvious since $u \preceq v \Rightarrow u \preceq\cdot\, v$. We proceed with the proof of $(15.1) \Rightarrow (15.2)$. Here we need to consider two cases

- *Case* $\hat{u} = \varepsilon$. Then $u = \tau^i$ for some $i \geq 0$ and we proceed by induction on i. If $i = 0$, $u = \varepsilon$ and we may choose $v = \varepsilon$ too and there is nothing to prove. If $i = m+1$ then for some p_1 we have that one of the folowing diagrams holds with $v = \tau^k$ or $v = \tau^{k+1}$ for some $k \leq m$.

$$\begin{array}{ccccc} p & \xrightarrow{\tau^m} & p_1 & \xrightarrow{\tau} & p' \\ | & & | & & | \\ R & & R & & R \\ | & & | & & | \\ q & \xrightarrow{\tau^k} & q_1 & \xrightarrow{\tau} & q' \end{array} \qquad \textit{or} \qquad \begin{array}{ccccc} p & \xrightarrow{\tau^m} & p_1 & \xrightarrow{\tau} & p' \\ | & & | & & | \\ R & & R & & R \\ | & & | & & | \\ q & \xrightarrow{\tau^k} & q_1 & = & q' \end{array}$$

- *Case* $\hat{u} = \alpha \in V$. Then $u = \tau^i \alpha \tau^j$ for some $i, j \geq 0$. The proof for this case uses that of the previous case and is summarized in the following diagram.

$$\begin{array}{ccccccc} p & \xrightarrow{\tau^i} & p_1 & \xrightarrow{\alpha} & p_2 & \xrightarrow{\tau^j} & p' \\ | & & | & & | & & | \\ R & & R & & R & & R \\ | & & | & & | & & | \\ q & \xrightarrow{\tau^k} & q_1 & \xrightarrow{\alpha} & q_2 & \xrightarrow{\tau^m} & q' \end{array}$$

  Hence $v = \tau^k \alpha \tau^m$ where $0 \leq k \leq i$ and $0 \leq m \leq j$.

Proof of corollary 16.

By restricting our attention to EA, it is obvious that $(16.1) \Rightarrow (15.2)$ and $(16.2) \Rightarrow (15.3)$. The rest of this proof may be used to show both $(15.2) \Rightarrow (16.1)$ and $(15.3) \Rightarrow (16.2)$. Let $p \xrightarrow{s} p'$. We simply divide up s into blocks of extended actions, say $s = u_1 \ldots u_k$. For each $i = 1, \ldots, k$ it is possible to find a v_i such that $u_i \preceq v_i$ and the following diagram holds.

$$\begin{array}{ccccccccc} p & \xrightarrow{u_1} & p_1 & \xrightarrow{u_2} & \cdots & \xrightarrow{u_{k-1}} & p_{k-1} & \xrightarrow{u_k} & p_k = p' \\ | & & | & & & & | & & | \\ R & & R & & R & & R & & R \\ | & & | & & & & | & & | \\ q & \xrightarrow{v_1} & q_1 & \xrightarrow{v_2} & \cdots & \xrightarrow{v_{k-1}} & q_{k-1} & \xrightarrow{v_k} & q_k = q' \end{array}$$

Taking $t = v_1 \ldots v_k$, we have $s \preceq t$ and $s \preceq\!\cdot\, t$.

Proof of theorem 22

$(\Rightarrow)$. It is easy to see from the definition of h that for any $u, v \in EA$,

$$u \,\widehat{=}\, v \text{ iff } h(u) \simeq h(v) \tag{3}$$

and

$$u \preceq\!\cdot\, v \text{ iff } h(u) \geq h(v) \tag{4}$$

We know that R is

1. a weak bisimulation if and only if it is a ($\widehat{=}$, $\widehat{=}$)-bisimulation,
2. an elaboration if and only if it is a ($\widehat{=}$, $\preceq\!\cdot$)-bisimulation (ref corollary 20), and
3. an efficiency prebisimulation if and only if it is a ($\preceq\!\cdot$, $\preceq\!\cdot$)-bisimulation (ref corollary 20).

Let R be any of the above induced bisimulations. Let $\langle p, q \rangle \in R$. Then for all $u, v \in EA$ we have from (3) that

1. $p \xrightarrow{u} p' \Rightarrow \exists v, q' : u \,\widehat{=}\, v \wedge q \xrightarrow{v} q' \wedge p' R q'$ implies $p \xrightarrow{h(u)} p' \Rightarrow \exists v, q' : h(u) \simeq h(v) \wedge q \xrightarrow{h(v)} q' \wedge p' R q'$,

2. $q \xrightarrow{v} q' \Rightarrow \exists u, p' : u \mathrel{\widehat{=}} v \wedge p \xrightarrow{u} p' \wedge p'Rq'$ implies $q \xrightarrow{h(v)} q' \Rightarrow \exists u, p' : h(u) \simeq h(v) \wedge p \xrightarrow{h(v)} p' \wedge p'Rq'$

and from (4) that

1. $p \xrightarrow{u} p' \Rightarrow \exists v, q' : u \preceq\!\cdot\, v \wedge q \xrightarrow{v} q' \wedge p'Rq'$ implies $p \xrightarrow{h(u)} p' \Rightarrow \exists v, q' : h(u) \geq h(v) \wedge q \xrightarrow{h(v)} q' \wedge p'Rq'$
2. $q \xrightarrow{v} q' \Rightarrow \exists u, p' : u \preceq\!\cdot\, v \wedge p \xrightarrow{v} p' \wedge p'Rq'$ implies $q \xrightarrow{h(v)} q' \Rightarrow \exists u, p' : h(u) \geq h(v) \wedge p \xrightarrow{h(v)} p' \wedge p'Rq'$

By choosing two of the appropriate clauses the result follows for each of the bisimulations in question.

($\Longleftarrow$) From the definition of h we have, for all $m \in \mathbb{N}$,

$$p \xrightarrow{(\varepsilon,m)} p' \text{ in } h(\mathcal{L}) \text{ iff } p \xrightarrow{\tau^m} p' \text{ in } \mathcal{L} \tag{5}$$

and for all $\alpha \in V$, $m \in \mathbb{N}$,

$$p \xrightarrow{(\alpha,m)} p' \text{ in } h(\mathcal{L}) \text{ iff } \exists i, j \geq 0 : m = i + j \wedge p \xrightarrow{\tau^i \alpha \tau^j} p' \text{ in } \mathcal{L} \tag{6}$$

For each $(a, m) \in Act$, let $h^{-1}(a, m) = \{u \in EA \mid h(u) = (a, m)\}$, where $a \in V_\varepsilon$ and $m \in \mathbb{N}$. We then have that

$$(a, m) \geq (a, n) \text{ iff } \forall u \in h^{-1}(a, m) \forall v \in h^{-1}(a, n) : u \preceq\!\cdot\, v \tag{7}$$

We then have the following implications for all $\langle p, q \rangle \in \mathbb{P}$, $a \in V_\varepsilon$, $m, n \in \mathbb{N}$.

1. $p \xrightarrow{(a,m)} p' \Rightarrow \exists n, q' : q \xrightarrow{(a,n)} q' \wedge p'Rq'$ in $h(\mathcal{L})$ implies for any $u \in h^{-1}(a, m)$,
   $p \xrightarrow{u} p' \Rightarrow \exists v \in h^{-1}(a, n), q' : u \mathrel{\widehat{=}} v \wedge q \xrightarrow{v} q' \wedge p'Rq'$ in $\mathcal{L}$.
2. $q \xrightarrow{(a,n)} q' \Rightarrow \exists m, p' : p \xrightarrow{(a,m)} q' \wedge p'Rq'$ in $h(\mathcal{L})$ implies for any $v \in h^{-1}(a, n)$,
   $q \xrightarrow{v} q' \Rightarrow \exists u \in h^{-1}(a, m), q' : u \mathrel{\widehat{=}} v \wedge p \xrightarrow{u} p' \wedge p'Rq'$ in $\mathcal{L}$.
3. $p \xrightarrow{(a,m)} p' \Rightarrow \exists n, q' : m \geq n \wedge q \xrightarrow{(a,n)} q' \wedge p'Rq'$ in $h(\mathcal{L})$ implies for any $u \in h^{-1}(a, m)$,
   $p \xrightarrow{u} p' \Rightarrow \exists v \in h^{-1}(a, n), q' : u \preceq\!\cdot\, v \wedge q \xrightarrow{v} q' \wedge p'Rq'$ in $\mathcal{L}$.
4. $q \xrightarrow{(a,n)} q' \Rightarrow \exists m, p' : m \geq n \wedge p \xrightarrow{(a,m)} q' \wedge p'Rq'$ in $h(\mathcal{L})$ implies for any $v \in h^{-1}(a, n)$,
   $q \xrightarrow{v} q' \Rightarrow \exists u \in h^{-1}(a, m), q' : u \preceq\!\cdot\, v \wedge p \xrightarrow{u} p' \wedge p'Rq'$ in $\mathcal{L}$.

From the above implications the result follows.