Full Abstraction for Series-Parallel Pomsets*

Luca Aceto
INRIA-Sophia Antipolis
F-06560 Valbonne Cedex, France

Abstract

In this paper, we provide a behavioural characterization of the class of finite, series-parallel pomsets by showing that this simple model based on partial orders is fully-abstract with respect to the behavioural equivalence obtained by applying Abramsky's testing scenario for bisimulation equivalence, [Ab87], in all refinement contexts, [AH89]. This casts the observability of series-parallel pomsets in a purely interleaving framework. Moreover, we prove that the order structure of a series-parallel pomset is completely revealed by its set of ST-traces, [Gl90], and provide a complete axiomatization of ST-trace equivalence over the class of series-parallel pomsets.

1 Introduction

In recent years, many models of concurrent computation based upon partial orders have been proposed in the literature, e.g. *Petri Nets* [Rei85], *Event Structures* [Win80,87], *Pomsets* [Pr86] and, more recently, *Causal Trees* [DD89]. These models are based upon the idea that concurrent, communicating systems are characterized by their causal structure, i.e. by the computational events a system performs during its evolution together with the causal dependencies amongst them, and that its proper description is necessary in accounting for the nonsequential behaviour of distributed systems. The mathematical tractability of causality-based models has been investigated in the literature by providing operational and denotational semantics for process algebras, such as **CCS** [Mil80,89], **CSP** [Hoare85] and **ACP** [BK85], in terms of the above mentioned models. Partial order operational semantics for standard process algebras have been presented in e.g. [BC88], [DDM88], [DD89], and

*This work has been carried out at the University of Sussex, Falmer (U.K.). It has been supported by a grant from the United Kingdom Science and Engineering Research Council and by the Esprit BRA project CEDISYS.

denotational semantics are given in e.g. [Win82], [Go88], [Tau89]. Several notions of equivalence over the above mentioned models, which allow to abstract from the way processes evolve, have recently been proposed in the literature (the interested reader is invited to consult [GV87] and [Gl90] for a comparison among some of the proposals), thus importing in the partial ordering setting some of the abstraction techniques supported by the standard interleaving equivalences and preorders [HM85], [H88a], [BHR84].

However, not much work has been carried out in studying reasonable testing scenarios, [DH84], which justify the use of these models in giving semantics to concurrent programming languages. Notions of observability play a fundamental rôle in the study of suitable semantics for programming languages. Following Milner and Plotkin's paradigm, mathematical models for programming languages should be justified by comparing them with some natural notion of behaviourally defined equivalence between processes. Models that are in complete agreement with the coarsest equivalence over processes induced by the chosen notion of observability are called *fully abstract* in [Mil77], [Pl77], [HP79], [Sto88]. As fully abstract semantic models are the most abstract ones which are consistent with the chosen notion of observability, it is natural to try to justify the choice of a model for a language by showing that it induces exactly all the distinctions that can be made by means of some natural notion of observation.

The main aim of this paper is to provide such a behavioural justification for a simple model based on partial orders, namely the class of *series-parallel* or *N-free pomsets* [Gi84], [Pr86]. Series-parallel pomsets have been extensively studied in the literature, see e.g. [Gi84], [BC88], [Ts88], and have a pleasing algebraic and order-theoretic structure that will be exploited in the proofs of the main results of this paper. Following Gischer, the algebraic structure of the class of series-parallel pomsets will allow us to relate it to a simple process algebra, whose terms are built from a set of generators by means of the operators of sequential and parallel composition. This gives us a syntax for denoting such partially ordered structures and will allow us to give a standard LTS semantics for the resulting language. We shall define a standard notion of observational equivalence over processes by means of the bisimulation technique [Pa81], [Mil83]. The notion of observability underlying such a notion of equivalence has been thoroughly investigated in [Ab87] and is called "tightly-controlled testing" in [H88b]. Series-parallel pomsets do not give rise to a fully abstract model with respect to standard bisimulation equivalence; however, by enriching the language with a *refinement operator* like the one used in [AH89], [NEL88], [GG88], and closing bisimulation equivalence with respect to all the contexts built using this new language construct we shall be able to make series-parallel pomsets fully observable. In other words, series-parallel pomsets are fully abstract with respect to the coarsest congruence obtained by applying Abramsky's testing scenario in all refinement contexts. By relying on results from [AH89], we shall be able to provide a natural behavioural characterization of series-parallel pomsets in terms of Hennessy's *timed equivalence* [H88c]. These results will cast the observability of series-parallel pomsets in a well-known interleaving setting.

A natural notion of basic observation which is widely used in the interleaving models for concurrency is that of *trace*, [Hoare85]. Indeed, some natural models for concurrency, e.g. Hennessy's *Acceptance Trees*, [H85], and the *Failures model*, [BHR84], have been shown to be fully abstract with respect to behavioural equivalences which are intrinsically based on such a notion of observability, [Main88], [H88a]. A natural question to ask is whether series-parallel pomsets can be made fully observable by assuming a trace-based notion of observation. In this paper, we shall provide a partial answer to this question by showing that the order structure of a series-parallel pomset is totally revealed by its set of *ST-traces*, [Gl90]. ST-semantics has been recently proposed in [GV87], [Gl90] as a refinement of *split-semantics*, [H88c], in which an explicit link is required between the beginning and the end of any event. It will be shown that series-parallel pomsets give rise to a fully abstract model with respect to ST-trace equivalence over the simple process algebra considered in this paper. As a corollary of this result, we shall be able to give a complete axiomatization of ST-trace equivalence over the class of series-parallel pomsets.

We now give a brief outline of the remainder of the paper. Section 2 is devoted to a review of mostly standard material in the theory of pomsets. Two behavioural semantics for series-parallel pomsets, based on the notion of bisimulation equivalence, are presented in §3. We shall show that series-parallel pomsets are fully abstract with respect to the finer of the two behavioural semantics, which may be seen as arising by applying Abramsky's testing scenario for bisimulation equivalence in all refinement contexts. The proof of this result is algebraic in nature and relies on Gischer's axiomatization of the theory of series-parallel pomsets, [Gi84]. Section 4 is entirely devoted to providing another behavioural characterization for series-parallel pomsets. We shall prove that ST-trace equivalence coincides with equality over the class of SP pomsets, thus giving a trace-theoretic understanding of this simple model based on partial orders. We end with a conclusion and a discussion of related work.

2 Series-parallel pomsets

This section will be devoted to a brief review of some basic notions of the theory of *partially ordered multisets* (or *pomsets* in Pratt and Gischer's terminology) which will find application in the remainder of the paper. The interested reader is referred to [Gr81], [Gi84], [Pr86], for more information on pomsets and further references. The following definition introduces the main objects of study of the paper.

Definition 2.1 *1. A labelled poset $\mathbb{P}$ over a label set L is a triple $\mathbb{P} = (P, <, l)$, where*

- *P is a finite set of* events*,*
- *$<$ is a binary, transitive and acyclic relation over P, and*
- *$l : P \to L$ is a labelling function.*

Two L-labelled posets $\mathbb{P}_i = (P_i, <_i, l_i)$, $i = 1, 2$, are isomorphic, written $\mathbb{P}_1 \cong \mathbb{P}_2$, iff there exists a bijective function $h : P_1 \to P_2$ such that, for all $u, v \in P_1$, $u <_1 v$ iff $h(u) <_2 h(v)$ and $l_1(u) = l_2(h(u))$.

2. A pomset over L, $\alpha = [P, <, l]$, is an isomorphism class of L-labelled posets. For a label set L, $\mathbf{Pom[L]}$ *will denote the set of pomsets over L and will be ranged over by α, $\beta \ldots$.*

Several operations on pomsets have been defined in the above given references. Since pomsets are isomorphism classes of labelled posets, it will be convenient to define operations on them by using arbitrary representatives of the isomorphism class. For each operation it will be straightforward to establish that the result of the operation is independent of the chosen representative and such verifications will be omitted. Let $\mathbf{A}$ be a set of *observable actions* ranged over by $a, b, a' \ldots$. In the remainder of the paper we shall only need the following operations over $\mathbf{Pom[A]}$.

- *Empty pomset.* $\mathbf{1}$ will denote the isomorphism class of the $\mathbf{A}$-labelled poset $(\emptyset, \emptyset, \emptyset)$.

- *Atomic actions.* For each $a \in \mathbf{A}$, a will denote, with abuse of notation, the isomorphism class of the one element poset labelled with a. In what follows, $\mathbf{A}$ will be used to denote, with abuse of notation, the set of all such atomic pomsets.

- *Sequential and parallel composition.* Let $\alpha = [P_1, <_1, l_1]$ and $\beta = [P_2, <_2, l_2]$ be pomsets on $\mathbf{A}$ and assume, wlog, that $P_1 \cap P_2 = \emptyset$. Then $\alpha;\beta$, the *sequential composition* of α and β is given by

$$\alpha;\beta = [P_1 \cup P_2, <_1 \cup <_2 \cup (P_1 \times P_2), l_1 \cup l_2]$$

  and $\alpha|\beta$, the *parallel composition* of α and β, is given by

$$\alpha|\beta = [P_1 \cup P_2, <_1 \cup <_2, l_1 \cup l_2].$$

$$
\begin{array}{llcll}
(PAR1) & x|nil & = & x & \\
(PAR2) & x|y & = & y|x & \\
(PAR3) & (x|y)|z & = & x|(y|z) & \\
 & & & & \\
(SEQ1) & x;nil & = & x & = nil;x \\
(SEQ2) & (x;y);z & = & x;(y;z) &
\end{array}
$$

Figure 1: The set of axioms E

Following Gischer [Gi84], the class SP of *series-parallel pomsets* over $\mathbf{A}$ may now be defined to be the closure of $\mathbf{A}$ and $\mathbf{1}$ with respect to the operations of sequential and parallel compositon. The definition of the class of pomsets SP has a pleasing algebraic flavour; indeed, the class of pomsets SP is in close correspondence with the set of terms $\mathcal{SP}$ built from the set of observable actions $\mathbf{A}$ by means of the operators of sequential and parallel composition. More formally, let $\mathcal{SP}$ be the set of terms generated by the syntax

$$p ::= nil \mid a \mid p;p \mid p|p,$$

where $a \in \mathbf{A}$. $\mathcal{SP}$ will be ranged over by $p, q, p' \dots$. Following Gischer, the set of terms $\mathcal{SP}$ may be interpreted as series-parallel pomsets by defining the semantic map $[\![\cdot]\!] : \mathcal{SP} \rightarrow SP$ as follows:

- $[\![nil]\!] = \mathbf{1}$,
- $[\![a]\!] = a$,
- $[\![p;q]\!] = [\![p]\!];[\![q]\!]$ and
- $[\![p|q]\!] = [\![p]\!]|[\![q]\!]$.

The following theorem, which formalizes the close connection between $\mathcal{SP}$ and SP and gives a complete axiomatization of the congruence on $\mathcal{SP}$ induced by the above given denotational semantics, has been proven in [Gi84] (Theorem 5.2, page 23). Let $=_E$ denote the least $\mathcal{SP}$-congruence which satisfies the set of axioms E in Figure 1.

Theorem 2.1 (Gischer) *For each $p, q \in \mathcal{SP}$, $[\![p]\!] = [\![q]\!]$ iff $p =_E q$.*

The algebraic characterization of the theory of SP pomsets given by the above theorem will provide the key to their behavioural characterization, which will be presented in the following section; namely, we shall give a behavioural view of the processes in $\mathcal{SP}$, based on a well-understood testing scenario familiar from the theory of bisimulation semantics, and prove that the denotational semantics for $\mathcal{SP}$ given by the map $[\![\cdot]\!]$ is fully abstract with respect to it. Following Milner and Plotkin's paradigm, this will justify the choice of SP as a denotational model for $\mathcal{SP}$ by showing that SP is the most abstract model for $\mathcal{SP}$ which is consistent with the chosen testing scenario.

We end this review of standard material on series-parallel pomsets with an order-theoretic characterization of the class of pomsets SP. It is well-known that the class of pomsets SP coincides with that of the so-called *N-free pomsets*, see e.g. [Gi84] (Theorem 3.2, pp. 14-15) and [BC88], where a more general result is proven for Event Structures. Here we only present a result from [Ts88] giving a characterization of SP pomsets in terms of their order structure which will find application in §4.

Proposition 2.1 (Tschantz) *A pomset $[P,<,l]$ is series-parallel iff the following property holds:*

$$(N)\ \forall w,x,y,z \in P\ \ w<y, w<z \text{ and } x<z \text{ imply } y<z \text{ or } w<x \text{ or } x<y.$$

In what follows, a labelled poset $\mathbb{P}$ will be said to be series-parallel (SP) iff it satisfies the above-given property (N). $\mathbf{Pos}[\mathbf{A}]$ will denote the class of SP posets labelled on $\mathbf{A}$.

3 Full-abstraction for series-parallel pomsets

This section will be entirely devoted to a discussion of a behavioural semantics for the simple language $\mathcal{SP}$ and to a proof of full abstraction of the denotational semantics given by the map $[\![\cdot]\!]$ with respect to it. Following Milner and Plotkin's approach, the behavioural view of processes we shall present, which is based on the notion of testing characterizing standard bisimulation semantics studied in [Ab87], will justify the denotational semantics in terms of series-parallel pomsets. In what follows, we shall introduce two operational semantics for the set of processes $\mathcal{SP}$ and two notions of observational equivalence for it. Relying on results from [AH89], we shall study the relationships between the two behavioural theories of processes and prove that the denotational semantics is fully abstract with respect to the finer one, the timed equivalence proposed in [H88c].

Operationally, the constructs in the language for processes $\mathcal{SP}$ will be interpreted in a fairly standard way; following Milner [Mil80,89], *nil* will be interpreted as the process that cannot perform any move. A generator $a \in \mathbf{A}$ will be interpreted as a process which is capable of performing the task represented by a and terminate in doing so. The combinators ; and

$$
\begin{array}{llll}
(1) & a \xrightarrow{a} nil & & \\
(2) & p \xrightarrow{a} p' & \text{implies} & p;q \xrightarrow{a} p';q \\
(3) & p\surd, q \xrightarrow{a} q' & \text{imply} & p;q \xrightarrow{a} q' \\
(4) & p \xrightarrow{a} p' & \text{implies} & p|q \xrightarrow{a} p'|q \\
 & & & q|p \xrightarrow{a} q|p'
\end{array}
$$

Figure 2: Axiom and rules for $\xrightarrow{a}$

$|$ will stand for sequential composition and parallel composition (without communication), respectively. Both the operational semantics for the language $\mathcal{SP}$ consist of two ingredients:

1. a termination predicate $\surd$, used in giving an operational account of the sequential composition operator, and

2. a standard LTS semantics for $\mathcal{SP}$ given using Plotkin's SOS method, [Pl81].

The termination predicate $\surd$ is the least subset of $\mathcal{SP}$ which satisfies the following axiom and rule:

- $nil \in \surd$,
- $p \in \surd$ and $q \in \surd$ imply $p;q \in \surd$ and $p|q \in \surd$.

In what follows, $p \in \surd$ will be often written as $p\surd$. Using this termination predicate we may now give the first Labelled Transition System semantics for $\mathcal{SP}$; this semantics will be based on the assumption that processes evolve by performing actions which are atomic. For each $a \in \mathbf{A}$, $\xrightarrow{a}$ will denote the least binary relation over $\mathcal{SP}$ which satisfies the axiom and rules given in Figure 2. A standard notion of observational equivalence over $\mathcal{SP}$ may now be defined by means of the bisimulation technique [Pa81], [Mil83]. The relation $\sim \subseteq \mathcal{SP}^2$ is the largest symmetric relation which satisfies, for all $p, q \in \mathcal{SP}$, $p \sim q$ if, for all $a \in \mathbf{A}$,

$$p \xrightarrow{a} p' \text{ implies } q \xrightarrow{a} q' \text{ and } p' \sim q', \text{ for some } q'.$$

The following proposition is then standard.

Proposition 3.1 *$\sim$ is a $\mathcal{SP}$-congruence.*

The testing scenario which is needed to characterize $\sim$ as a testing equivalence has been spelt out by S. Abramsky in [Ab87]; a tutorial exposition of Abramsky's testing characterization of the equivalence $\sim$ may be found in [H88b]. The main import of Abramsky's results is that, by using $\sim$ as our basic notion of equivalence, we automatically have a testing scenario justifying it; in the remainder of this section we shall behaviourally characterize the class of *SP* pomsets by means of the testing scenario presented in [Ab87]. However, as it is stated in the following proposition, there is still a mismatch between the denotational semantics for $\mathcal{SP}$ given by $[\![\cdot]\!]$ and the behavioural semantics given in terms of $\sim$. In fact, the denotational semantics is sound, but *not* complete, with respect to the behavioural one.

Proposition 3.2 (i) *For all* $p, q \in \mathcal{SP}$, $[\![p]\!] = [\![q]\!]$ *implies* $p \sim q$.

(ii) $a; a \sim a|a$, *but* $[\![a;a]\!] \neq [\![a|a]\!]$.

The import of the above proposition is that, not surprisingly, series-parallel pomsets do not give rise to a fully-abstract model with respect to standard bisimulation equivalence. The remainder of this section is devoted to showing how to define a behavioural semantics for the language $\mathcal{SP}$ with respect to which the denotational model *SP* is fully abstract. Following the system-testing approach discussed in [H88b], the discriminating power of the testing scenario which induces the equivalence $\sim$ over $\mathcal{SP}$ may be increased by enriching the language with some computationally meaningful constructs and by applying the basic tests presented in [Ab87] to processes in all language contexts built using the new combinators. In what follows, we shall apply this philosophy by enriching the language $\mathcal{SP}$ with a *refinement operator* ρ like the ones considered in e.g. [NEL88], [GG88], [AH89].

Definition 3.1 (i) *A* refinement map *is a function* $\rho : \mathbf{A} \to \mathcal{SP}$.

(ii) *The closure of* $\sim$ *with respect to all refinement contexts,* $\sim^\rho$, *is given by*

$$p \sim^\rho q \text{ iff, for all refinement maps } \rho,\ p\rho \sim q\rho,$$

where $p\rho$ *and* $q\rho$ *are the terms (in* $\mathcal{SP}$*) obtained by syntactically replacing* $\rho(a)$ *for each occurrence of* a *in* p *and* q*, respectively.*

By construction, $\sim^\rho$ is the largest $\mathcal{SP}$-congruence contained in $\sim$ which is preserved by all refinements of actions by processes. As pointed out before, this notion of equivalence may be seen as arising by applying Abramsky's testing scenario in all refinement contexts. We shall now show that $\sim^\rho$ is indeed the behavioural counterpart of the denotational model *SP*, i.e. that series-parallel pomsets are fully abstract with respect to the behavioural semantics induced by $\sim^\rho$. The proof of this claim proceeds in two steps. First of all, relying on work presented in [AH89], we shall give a behavioural characterization of $\sim^\rho$ in terms of Hennessy's timed equivalence [H88c], $\sim_t$. Secondly, we shall prove that the set of equations E in Figure

1 completely axiomatize $\sim_t$ over $\mathcal{SP}$. The result will then follow as $\sim_t$ and the congruence induced over $\mathcal{SP}$ by the denotational semantics have a common axiomatization.

It is easy to see that $\sim$ is strictly weaker than $\sim^\rho$. For instance, as previously remarked, $a; a \sim a|a$; however, $(a; a)\rho \not\sim (a|a)\rho$, where ρ is any refinement map such that $\rho(a) = b; c$. In fact, $(a|a)\rho = (b; c)|(b; c)$ can perform two b-moves in a row, whilst $(a; a)\rho = (b; c); (b; c)$ can not. Hence $a; a \not\sim^\rho a|a$ and this implies that $\sim$ itself is not preserved by the refinement combinator over $\mathcal{SP}$. As pointed out in e.g. [AH89], this is not at all surprising as the definition of $\sim$ is based on the assumption that processes evolve from one state to another by performing actions which are atomic. This behavioural view of processes becomes inadequate in the presence of a refinement operator like ρ and a more refined behavioural description of the processes in $\mathcal{SP}$ is needed. If actions are no longer atomic, a minimal consequence is that they have a beginning and an ending. This is exactly the intuition underlying the timed view of processes presented in [H88c]. By assuming that beginnings and endings of actions are distinct events and that they may be observed, a new behavioural description of processes may be obtained. Formally, for each $a \in \mathbf{A}$, $S(a)$ and $F(a)$ are used to denote the beginning and the termination of action a, respectively. $Ev =_{def} \{S(a), F(a) \mid a \in \mathbf{A}\}$ will be the new set of observable events and will be ranged over by e.

As pointed out in [H88c], the language for processes is not sufficiently expressive to describe a possible state a process may reach by executing the beginning of an action. To overcome this problem, a new symbol $S(a)$ for each $a \in \mathbf{A}$ is introduced into the language. $S(a)$ will denote the state in which action a is being executed but is not terminated yet. The set of *process states* $\mathcal{S}$ is the least set which satisfies:

i) $p \in \mathcal{SP}$ implies $p \in \mathcal{S}$

ii) $a \in \mathbf{A}$ implies $S(a) \in \mathcal{S}$

iii) $s \in \mathcal{S}$, $p \in \mathcal{SP}$ imply $s; p \in \mathcal{S}$

iv) $s_1, s_2 \in \mathcal{S}$ imply $s_1|s_2 \in \mathcal{S}$.

The operational semantics for process states may be defined following standard lines. For each $e \in Ev$, the transition relation $\stackrel{e}{\Longrightarrow}$ over $\mathcal{S}$ is defined as the least binary relation over $\mathcal{S}$ which satisfies the axioms and rules in Figure 3. The defining rules of $\stackrel{e}{\Longrightarrow}$ use a termination predicate on process states, $\surd_{\mathcal{S}}$, which is induced on $\mathcal{S}$ by the one previously defined on $\mathcal{SP}$; namely, $s\surd_{\mathcal{S}}$ iff $s \in \mathcal{SP}$ and $s\surd$. A standard behavioural equivalence over process states may now be defined using the notion of bisimulation. A relation $\mathcal{R} \subseteq \mathcal{S}^2$ is a *t-bisimulation* iff it is symmetric and, for each $(s_1, s_2) \in \mathcal{R}$, $e \in Ev$,

1. $a \stackrel{S(a)}{\Longrightarrow} S(a)$
2. $S(a) \stackrel{F(a)}{\Longrightarrow} nil$
3. $s \stackrel{e}{\Longrightarrow} s'$ implies $s;p \stackrel{e}{\Longrightarrow} s';p$
4. $s\surd_S$, $p \stackrel{e}{\Longrightarrow} s'$ imply $s;p \stackrel{e}{\Longrightarrow} s'$
5. $s_1 \stackrel{e}{\Longrightarrow} s_1'$ implies $s_1|s_2 \stackrel{e}{\Longrightarrow} s_1'|s_2$, $s_2|s_1 \stackrel{e}{\Longrightarrow} s_2|s_1'$

Figure 3: Axioms and rules for $\stackrel{e}{\Longrightarrow}$

$$s_1 \stackrel{e}{\Longrightarrow} s_1' \text{ implies, for some } s_2',\ s_2 \stackrel{e}{\Longrightarrow} s_2' \text{ and } (s_1', s_2') \in \mathcal{R}.$$

Let $\sim_t$ denote the maximum t-bisimulation. The following theorem from [AH89] states that $\sim_t$ gives a behavioural characterization of the relation $\sim^\rho$ defined previously by purely algebraic means.

Theorem 3.1 (AH89) *For all $p, q \in \mathcal{SP}$, $p \sim_t q$ iff $p \sim^\rho q$.*

The behavioural characterization of $\sim^\rho$ given by the above-stated theorem will be the touchstone for relating $\sim^\rho$ to the denotational semantics for $\mathcal{SP}$ in terms of series-parallel pomsets. The proof of full-abstractness of the denotational semantics with respect to $\sim^\rho$ relies on Gischer's axiomatization of the congruence induced by $[\![\cdot]\!]$ over $\mathcal{SP}$ stated in Theorem 2.1. Let us recall, for the sake of clarity, that $=_E$ denotes the least congruence over $\mathcal{SP}$ that satisfies the set of equations E in Figure 1. The key to the full-abstraction result is then provided by the following theorem, whose proof, which is rather long and involved, may be found in the full version of the paper [Ac90].

Theorem 3.2 (Equational characterization of $\sim_t$) *For all $p, q \in \mathcal{SP}$, $p \sim_t q$ iff $p =_E q$.*

The full-abstractness of series-parallel pomsets with respect to $\sim^\rho$ now follows fairly straightforwardly from the results stated above.

Theorem 3.3 (Full-abstraction for series-parallel pomsets) *For all $p, q \in \mathcal{SP}$, $[\![p]\!] = [\![q]\!]$ iff $p \sim^\rho q$.*

Proof: Assume $p, q \in \mathcal{SP}$. Then:

$$\begin{aligned} [\![p]\!] = [\![q]\!] &\iff p =_E q \quad \text{by Theorem 2.1} \\ &\iff p \sim_t q \quad \text{by Theorem 3.2} \\ &\iff p \sim^\rho q \quad \text{by Theorem 3.1.} \ \square \end{aligned}$$

We end this section with a few comments on the equational characterization of $\sim_t$ provided by Theorem 3.2. First of all, it is interesting to remark that the equational characterization of $\sim_t$, and consequently of $\sim^\rho$, is finite and does not make use of any auxiliary operator. This is not in contrast with F. Moller's results on the non-finite axiomatizability of "strong bisimulation"-like equivalences because $\sim_t$, when considered over $\mathcal{SP}$, does not satisfy his "reasonableness criterion". See [Mol89] for more details. Moreover, we can prove a stronger version of Theorem 3.2 stating that the above-given equational characterization of $\sim_t$ is also ω-complete [Mol89], i.e. complete for the open term theory.

We shall now present a proof of the ω-completeness of the set of equations E with respect to $\sim_t$. Let $\mathtt{Var}$ be a countable set of *variables* ranged over by x, y, z. $\mathcal{SP}(\mathtt{Var})$ will denote the set of expressions built by adding the clause

- $x \in \mathtt{Var}$ implies $x \in \mathcal{SP}(\mathtt{Var})$

to the formation rules for $\mathcal{SP}$. $\mathcal{SP}(\mathtt{Var})$ will be ranged over by $t, t', t_1, \ldots$. The equivalence $\sim_t$ can now be extended to $\mathcal{SP}(\mathtt{Var})$ in the standard way as follows:

Definition 3.2 *Let $t, t' \in \mathcal{SP}(\mathtt{Var})$. Then $t \sim_t t'$ iff for all closed substitutions $\sigma : \mathtt{Var} \to \mathcal{SP}$, $t\sigma \sim_t t'\sigma$. An equational theory EQ over the signature of $\mathcal{SP}$ is then called* ω-complete *with respect to $\sim_t$ iff for all open terms $t, t' \in \mathcal{SP}(\mathtt{Var})$, $t \sim_t t'$ iff $EQ \vdash t = t'$. $EQ \vdash t = t'$ will also be written as $t =_{EQ} t'$.*

We shall now prove that the set of axioms E presented in Figure 1 is indeed ω-complete with respect to $\sim_t$ over $\mathcal{SP}(\mathtt{Var})$. In the proof we shall make use of a novel technique for proving the ω-completeness of a set of equations developed by J.F. Groote in [Gro90]. For the sake of clarity, we shall now briefly outline Groote's proof-technique for showing the ω-completeness of a set of equations. Assume that t and t' are open terms in $\mathcal{SP}(\mathtt{Var})$ and $t \sim_t t'$, i.e., by Theorem 3.2, $t\sigma =_E t'\sigma$ for all closed substitutions σ. The application of Groote's technique requires the isolation of a closed substitution $\rho : \mathtt{Var} \to \mathcal{SP}$, mapping each variable occurring in t and t' to a distinguished closed term representing this variable, and of a translation map $R : \mathcal{SP} \to \mathcal{SP}(\mathtt{Var})$, which replaces each subterm representing a variable by the variable itself. This pair of functions is required to satisfy the following conditions:

(1) $t =_E R(\rho(t))$ and $t' =_E R(\rho(t'))$,

(2) for each $\odot \in \{;, |\}$ and $p_1, p_2, q_1, q_2 \in \mathcal{SP}$, $R(p_1 \odot p_2) =_{E'} R(q_1 \odot q_2)$, where $E' = E \cup \{R(p_i) = R(q_i) \mid i = 1, 2\}$, and

(3) for each axiom $t_1 = t_2$ in E and closed substitution σ, $R(\sigma(t_1)) =_E R(\sigma(t_2))$.

Having found such a pair of maps ρ and R satisfying conditions (1)-(3) above, we could then obtain the ω-completeness of E with respect to $\sim_t$ by applying the following instance of Theorem 3.1 from [Gro90], page 317.

Theorem 3.4 *If for each $t, t' \in \mathcal{SP}(\mathtt{Var})$ such that $t\sigma =_E t'\sigma$, for all closed substitutions σ, there exist a closed substitution $\rho : \mathtt{Var} \to \mathcal{SP}$ and a map $R : \mathcal{SP} \to \mathcal{SP}(\mathtt{Var})$ satisfying (1)-(3) above then E is ω-complete.*

We shall now apply the technique described above to prove that E is indeed ω-complete with respect to $\sim_t$ over $\mathcal{SP}$.

Theorem 3.5 **(ω-Completeness)** *For each $t, t' \in \mathcal{SP}(\mathtt{Var})$, $t \sim_t t'$ iff $t =_E t'$.*

Proof: Let $t, t' \in \mathcal{SP}(\mathtt{Var})$ be such that $t \sim_t t'$. By Theorem 3.4, in order to prove that E is ω-complete, it is sufficient to find $\rho : \mathtt{Var} \to \mathcal{SP}$ and $R : \mathcal{SP} \to \mathcal{SP}(\mathtt{Var})$ satisfying conditions (1)-(3) above. Define $\rho : \mathtt{Var} \to \mathcal{SP}$ by $\rho(x) = a_x \in \mathbf{A}$, where, for each $x, y \in \mathtt{Var}$,

- a_x does not occur in t and t', and
- $a_x = a_y$ implies $x = y$. (Note that such a map can be found because $\mathbf{A}$ is infinite)

The translation map $R : \mathcal{SP} \to \mathcal{SP}(\mathtt{Var})$ is defined by induction on the structure of $p \in \mathcal{SP}$ as follows:

- $R(nil) = nil$,
- $R(a) = \begin{cases} x & \text{if } a = a_x \\ a & \text{otherwise,} \end{cases}$
- $R(p \odot q) = R(p) \odot R(q)$, for $\odot \in \{;, |\}$.

We are now left to prove that ρ and R satisfy conditions (1)-(3). We examine each of the conditions in turn.

(1) We prove that, for all $\bar{t} \in \mathcal{SP}(\mathtt{Var})$ not containing actions of the form a_x, $\bar{t} =_E R(\rho(\bar{t}))$. The proof is by structural induction on $\bar{t}$. We only examine two of the cases.

  - $\bar{t} = a$. Then $R(\rho(a)) = R(a) = a$ because $a \neq a_x$, for all x. The claim now follows by the reflexivity of $=_E$.

- $\bar{t} = t_1; t_2$. Then we have that

$$\begin{array}{rcll} R(\rho(t_1;t_2)) & = & R(\rho(t_1);\rho(t_2)) & \\ & = & R(\rho(t_1));R(\rho(t_2)) & \\ & =_E & t_1;t_2 & \text{by induction.} \end{array}$$

(2) Let $\odot \in \{;,|\}$ and $p_1, p_2, q_1, q_2 \in \mathcal{SP}$. Then, letting $E' = E \cup \{R(p_i) = R(q_i) \mid i = 1,2\}$, we have that

$$\begin{array}{rcl} R(p_1 \odot p_2) & = & R(p_1) \odot R(p_2) \\ & =_{E'} & R(q_1) \odot R(q_2) \\ & = & R(q_1 \odot q_2). \end{array}$$

(3) Let $t_1 = t_2$ be an equation in E and σ be a closed substitution. Then it is easy to see that $R(\sigma(t_1)) =_E R(\sigma(t_2))$. For instance,

$$\begin{array}{rcll} R(\sigma((x|y)|z)) & = & R((\sigma(x)|\sigma(y))|\sigma(z)) & \\ & = & (R(\sigma(x))|R(\sigma(y)))|R(\sigma(z)) & \\ & =_E & R(\sigma(x))|(R(\sigma(x))|R(\sigma(y))) & \text{by (PAR1)} \\ & = & R(\sigma(x|(y|z))). & \end{array}$$

As ρ and R satisfy conditions (1)-(3), by Theorem 3.4 we have that E is indeed ω-complete. □

4 Series-parallel pomsets and ST-traces

In the previous section we showed that series-parallel pomsets are fully-abstract with respect to the equivalence obtained by applying Abramsky's testing scenario for bisimulation in all refinement contexts. The observability of SP pomsets was then cast in a well-known interleaving setting. The aim of this section is to investigate to what extent the model SP can be made fully observable by assuming a *trace-based* basic notion of observation. It will be shown that the causal structure of an N-free pomset is totally revealed by its set of *ST-traces* [Gl90], i.e. that SP pomsets are fully abstract with respect to ST-trace equivalence over the set of processes $\mathcal{SP}$. ST-semantics has recently been proposed in [GV87], [Gl90] as a refinement of the timed behavioural view of processes outlined in §3. This more refined view of processes is obtained by requiring a link between the beginning and the end of any event; this allows one to express that a start-action $S(a)$ and an end-action $F(a)$ represent the beginning and the end of the same occurrence of action a. Notions of ST-bisimulation and

ST-trace equivalence have been proposed and studied in [GV87], [Gl90] for Petri Nets and Event Structures, respectively, and the interested reader is invited to consult these references for more details on ST-semantics.

In what follows we shall mainly work with labelled posets rather than pomsets; this will make the technical development slightly less cumbersome. All the results will be lifted to pomsets and the process language $\mathcal{SP}$ in a straightforward way. Our first aim is to give the class of labelled SP posets the structure of a labelled transition system following the intuitions underlying the timed view of processes described in [H88c] and §3. In order to provide an LTS semantics for the class of $\mathbf{A}$-labelled posets, we shall have to extend the class of labelled posets in order to express those intermediate stages in the evolution of a process in which some actions have started but have not yet terminated.

Definition 4.1 *Let* $\mathbf{A_S} = \mathbf{A} \cup \{S(a) \mid a \in \mathbf{A}\}$. *An* $\mathbf{A_S}$*-labelled poset* $\mathbb{P} = (P, <, l)$ *is* sensible *iff, for all* $u \in P$, $l(u) = S(a)$, *for some* $a \in \mathbf{A}$, *implies* u *is minimal in* $\mathbb{P}$.

$\mathbf{Pos}[\mathbf{A_S}]$ *will denote the class of sensible, series-parallel* $\mathbf{A_S}$*-labelled posets.*

Note that each $\mathbb{P} \in \mathbf{Pos}[\mathbf{A}]$ is a sensible, series-parallel $\mathbf{A_S}$-labelled poset. Intuitively, $\mathbf{A_S}$-labelled posets $(P, <, l)$ in which $l(u) \in \mathbf{A}$, for all $u \in P$, are the model-theoretic counterpart of the processes in $\mathcal{SP}$ and those with at least a minimal element labelled $S(a)$, for some $a \in \mathbf{A}$, correspond to *proper states* in $\mathcal{S}$, i.e. states in which some actions will have started, but have not yet terminated. The following definition introduces the transition relations over $\mathbf{Pos}[\mathbf{A_S}]$.

Definition 4.2 (Transition relations for posets) *Let* $\mathbb{P} = (P, <, l) \in \mathbf{Pos}[\mathbf{A_S}]$. *Then:*

(i) $\mathbb{P} \xrightarrow{\langle S(a), u\rangle} \mathbb{P}'$ *iff*

- (a) *u is minimal in* $\mathbb{P}$,
- (b) $l(u) = a$, *and*
- (c) $\mathbb{P}' = (P, <, l')$, *where, for each* $v \in P$,

$$l'(v) = \begin{cases} S(a) & \textit{if } u = v \\ l(u) & \textit{otherwise.} \end{cases}$$

(ii) $\mathbb{P} \xrightarrow{\langle F(a), u\rangle} \mathbb{P}_1$ *iff*

- (a) *u is minimal in* $\mathbb{P}$,
- (b) $l(u) = S(a)$, *and*

(c) $\mathbb{P}_1 = (P_1, <_1, l_1)$, *where* $P_1 = P - \{u\}$ *and* $<_1$, l_1 *are the restrictions of* $<$ *and* l *to* P_1, *respectively.*

The following fact, whose proof follows easily from the definition of the transition relations $\xrightarrow{\langle e,u\rangle}$, $e \in Ev$, and is thus omitted, states that $\mathbf{Pos}[\mathbf{A_S}]$ is indeed closed under derivation.

Fact 4.1 (Closure under derivation) *Let* $\mathbb{P} \in \mathbf{Pos}[\mathbf{A_S}]$ *and* $e \in Ev$. *Then* $\mathbb{P} \xrightarrow{\langle e,u\rangle} \mathbb{P}'$ *implies* $\mathbb{P}' \in \mathbf{Pos}[\mathbf{A_S}]$.

Using the above-given operational semantics for $\mathbf{Pos}[\mathbf{A_S}]$, it is now possible to define a natural notion of complete trace of a poset $\mathbb{P} \in \mathbf{Pos}[\mathbf{A_S}]$. Intuitively, a complete trace γ of a poset $\mathbb{P} \in \mathbf{Pos}[\mathbf{A_S}]$ records a possible linear history of the evolution of the process denoted by $\mathbb{P}$, i.e. the set of events the process involves in together with their relative order of execution. In what follows we shall only be interested in this notion and the ones derived from it for SP posets $\mathbb{P} \in \mathbf{Pos}[\mathbf{A}]$.

Definition 4.3 (Complete traces) *Let* $\mathbb{P} = (P, <, l) \in \mathbf{Pos}[\mathbf{A}]$. *A sequence* $\gamma = \langle e_1, u_1\rangle \cdots \langle e_k, u_k\rangle \in (Ev \times P)^*$, $k \geq 0$, *is a* complete trace *of* $\mathbb{P}$ *iff there exist* $\mathbb{P}_0, \ldots, \mathbb{P}_k$ *in* $\mathbf{Pos}[\mathbf{A_S}]$ *such that*

(i) $\mathbb{P}_0 = \mathbb{P}$, $\mathbb{P}_k = (\emptyset, \emptyset, \emptyset)$ *and*

(ii) $\mathbb{P}_i \xrightarrow{\langle e_{i+1}, u_{i+1}\rangle} \mathbb{P}_{i+1}$, *for all* $i < k$.

$CT(\mathbb{P})$ will denote the set of complete traces of $\mathbb{P}$. The projection maps will be homomorphically extended to strings over $(Ev \times P)^*$, i.e. for $\gamma = \langle e_1, u_1\rangle \cdots \langle e_k, u_k\rangle$, $\pi_1(\gamma) = e_1 \cdots e_k$ and $\pi_2(\gamma) = u_1 \cdots u_k$.

It is easy to see that if $\gamma = \langle e_1, u_1\rangle \cdots \langle e_k, u_k\rangle$ is a complete trace of $\mathbb{P} = (P, <, l) \in \mathbf{Pos}[\mathbf{A}]$ then $P = \{u_1, \ldots, u_k\}$, $k = 2m$ where $|P| = m$ and, for all $u \in P$, there exist unique i, j such that $1 \leq i < j \leq k$, $\langle e_i, u_i\rangle = \langle S(a), u\rangle$ and $\langle e_j, u_j\rangle = \langle F(a), u\rangle$, with $a = l(u)$. By using the above notion of complete trace it is now possible to define two key notions of trace equivalence over $\mathbf{Pos}[\mathbf{A}]$. The first one, which is based on the operational intuition underlying the timed operational semantics defined in §3, is (complete) *split trace* equivalence [Va88], [Gl90]. Split trace equivalence is just standard interleaving trace equivalence, [Hoare85], but based on interleavings of beginnings and endings. *ST-trace* equivalence, [Gl90], will then be defined as a refinement of split trace equivalence by requiring that beginnings and endings of the same occurrence of an action $a \in \mathbf{A}$ are explicitly connected in a complete trace.

Definition 4.4 (Split and ST-trace equivalence) *Let* $\mathbb{P}, \mathbb{P}_1, \mathbb{P}_2 \in \mathbf{Pos}[\mathbf{A}]$.

(i) *A sequence $\sigma \in Ev^*$ is a* split trace *of $\mathbb{P}$ iff there exists $\gamma \in CT(\mathbb{P})$ such that $\pi_1(\gamma) = \sigma$. $S(\mathbb{P})$ will denote the set of split traces of $\mathbb{P}$. Then $\mathbb{P}_1$ and $\mathbb{P}_2$ are split trace equivalent, $\mathbb{P}_1 \approx_{2t} \mathbb{P}_2$, iff $S(\mathbb{P}_1) = S(\mathbb{P}_2)$.*

(ii) *$\mathbb{P}_1$ and $\mathbb{P}_2$ are* ST-trace *equivalent, $\mathbb{P}_1 \approx_{ST} \mathbb{P}_2$, iff the following conditions hold:*

- **(a)** *for each $\gamma_1 = \langle e_1, u_1 \rangle \cdots \langle e_k, u_k \rangle \in CT(\mathbb{P}_1)$ there exists $\gamma_2 = \langle f_1, v_1 \rangle \cdots \langle f_h, v_h \rangle \in CT(\mathbb{P}_2)$ such that*
  - *$h = k$, $\pi_1(\gamma_1) = \pi_1(\gamma_2)$, and*
  - *for all $1 \leq i < j \leq k$, $u_i = u_j$ iff $v_i = v_j$ (ST-condition);*
- **(b)** *viceversa, with the rôles of $\mathbb{P}_1$ and $\mathbb{P}_2$ interchanged.*

The following fact states two basic properties of the above-given notions of equivalence over **Pos**[**A**]. The first justifies our choice of working with labelled posets rather than pomsets by showing that the notions of equivalence given above may be consistently lifted to pomsets. The second states that $\approx_{ST}$ is at least as strong as split trace equivalence. For the sake of clarity, we recall that isomorphism between labelled posets is denoted by $\cong$ (see Definition 2.1).

Fact 4.2 *Let $\mathbb{P}_1$, $\mathbb{P}_2 \in$ **Pos**[**A**]. Then:*

(i) *$\mathbb{P}_1 \cong \mathbb{P}_2$ implies $\mathbb{P}_1 \approx_{2t} \mathbb{P}_2$ and $\mathbb{P}_1 \approx_{ST} \mathbb{P}_2$.*

(ii) *$\mathbb{P}_1 \approx_{ST} \mathbb{P}_2$ implies $\mathbb{P}_1 \approx_{2t} \mathbb{P}_2$.*

In the light of the statement above, $\approx_{2t}$ and $\approx_{ST}$ may be now extended to SP in a rather straightforward way.

Definition 4.5 *Let $\alpha = [P_1, <_1, l_1], \beta = [P_2, <_2, l_2] \in SP$. Then $\alpha \approx_{2t} \beta$ ($\alpha \approx_{ST} \beta$) iff $(P_1, <_1, l_1) \approx_{2t} (P_2, <_2, l_2)$ ($(P_1, <_1, l_1) \approx_{ST} (P_2, <_2, l_2)$).*

The remainder of this section will be entirely devoted to showing that ST-trace equivalence coincides with isomorphism over **Pos**[**A**] (and thus with equality over SP). This implies that SP pomsets can be made fully observable by assuming a trace-like notion of observation, albeit one in which beginnings and endings of the same occurrence of an action are explicitly linked. The following standard example shows that $\approx_{ST}$ does not coincide with isomorphism over general labelled posets and pomsets.

Example 4.1 *Let α and β denote the following pomsets:*

- *$\alpha = [\![(a;b)|(a;b)]\!]$ and*

- $\beta = [P, <, l]$, *where* $P = \{1,2,3,4\}$, $1 < 2$, $3 < 4$ *and* $1 < 4$, $l(1) = l(3) = a$ *and* $l(2) = l(4) = b$. *This pomset is just Gischer's* $N(a,b,a,b)$, *[Gi84].*

Then $\alpha \approx_{ST} \beta$, *but obviously* $\alpha \neq \beta$. *Note that* β *is not a series-parallel pomset.*

The following lemmas, which analyze basic properties of the transition relations $\xrightarrow{\langle e,u\rangle}$, $e \in Ev$, will be useful in the proof of the main result of this section. The following lemma states that sequences of start-moves are made up of independent transitions; such transitions may then be performed in any order without influencing the resulting target state. A similar property holds for end-moves.

Lemma 4.1 (Commuting start and end moves) *Let* $I\!P = (P, <, l) \in \mathbf{Pos}[\mathbf{A_S}]$. *Then the following properties hold.*

(i) $I\!P \xrightarrow{\langle S(a),u\rangle} I\!P' \xrightarrow{\langle S(b),v\rangle} I\!P''$ *implies* $I\!P \xrightarrow{\langle S(b),v\rangle} I\!P_1 \xrightarrow{\langle S(a),u\rangle} I\!P''$, *for some* $I\!P_1 \in \mathbf{Pos}[\mathbf{A_S}]$.

(ii) $I\!P \xrightarrow{\langle F(a),u\rangle} I\!P' \xrightarrow{\langle F(b),v\rangle} I\!P''$ *implies* $I\!P \xrightarrow{\langle F(b),v\rangle} I\!P_1 \xrightarrow{\langle F(a),u\rangle} I\!P''$, *for some* $I\!P_1 \in \mathbf{Pos}[\mathbf{A_S}]$.

The following lemma states that end-moves and start-moves corresponding to events which are not causally related may be performed in any order without influencing the resulting target state.

Lemma 4.2 *Let* $I\!P = (P, <, l) \in \mathbf{Pos}[\mathbf{A_S}]$. *Then* $I\!P \xrightarrow{\langle F(a),u\rangle} I\!P' \xrightarrow{\langle S(b),v\rangle} I\!P''$ *and* $u \not< v$ *imply* $I\!P \xrightarrow{\langle S(b),v\rangle} I\!P_1 \xrightarrow{\langle F(a),u\rangle} I\!P''$, *for some* $I\!P_1 \in \mathbf{Pos}[\mathbf{A_S}]$.

The following result presents a basic consistency requirement on the complete traces of a poset $I\!P = (P, <, l) \in \mathbf{Pos}[\mathbf{A}]$; namely that, for each $v \in P$, the end of each event $u < v$ must precede the start of v in every linear history of $I\!P$.

Lemma 4.3 *Let* $I\!P = (P, <, l) \in \mathbf{Pos}[\mathbf{A}]$. *Assume that* $u, v \in P$ *and* $u < v$. *Then* $\gamma = \langle e_1, u_1\rangle \cdots \langle e_k, u_k\rangle \in CT(I\!P)$, $\langle F(a), u\rangle = \langle e_i, u_i\rangle$ *and* $\langle S(b), v\rangle = \langle e_j, u_j\rangle$, *with* $l(u) = a$ *and* $l(v) = b$, *imply* $i < j$.

We now have all the technical material which is needed to prove the main theorem of this section, namely that $\approx_{ST}$ coincides with isomorphism over $\mathbf{Pos}[\mathbf{A}]$.

Theorem 4.1 (ST-trace equivalence = isomorphism over Pos[A])
Let $I\!P_i = (P_i, <_i, l_i) \in \mathbf{Pos}[\mathbf{A}]$, $i = 1, 2$. *Then* $I\!P_1 \cong I\!P_2$ *iff* $I\!P_1 \approx_{ST} I\!P_2$.

Proof: The "only if" implication follows by Fact 4.2. We shall now concentrate on the proof of the "if" implication. Assume that $I\!P_1, I\!P_2 \in \mathbf{Pos}[\mathbf{A}]$ and that $I\!P_1 \approx_{ST} I\!P_2$; we will show that $I\!P_1 \cong I\!P_2$. The proof proceeds in two steps:

1. first of all, we shall show that $\mathbb{P}_i$, $i = 1, 2$, may be recovered from a particular $\gamma_i \in CT(\mathbb{P}_i)$;

2. secondly, we shall construct an isomorphism between $\mathbb{P}_1$ and $\mathbb{P}_2$ by making use of the information on the order structure of the posets obtained in the previous step and the fact that $\mathbb{P}_1 \approx_{ST} \mathbb{P}_2$.

Following [Ts88], let $\ll$ be the ordering relation over $CT(\mathbb{P}_1)$ obtained by lexicographically extending the one over $Ev \times P_1$ given by

$$\langle S(a), u\rangle \ll \langle F(b), v\rangle, \text{ for all } a, b \in \mathbf{A} \text{ and } u, v \in P_1.$$

Let $\gamma_1 = \langle e_1, u_1\rangle \cdots \langle e_k, u_k\rangle$, $k \geq 0$, be minimal in $CT(\mathbb{P}_1)$ with respect to $\ll$. Then there exist $\mathbb{P}_1, \ldots, \mathbb{P}_{k-1}$ such that

$$\Gamma = \mathbb{P} \xrightarrow{\langle e_1, u_1\rangle} \mathbb{P}_1 \xrightarrow{\langle e_2, u_2\rangle} \cdots \mathbb{P}_{k-1} \xrightarrow{\langle e_k, u_k\rangle} (\emptyset, \emptyset, \emptyset).$$

As previously remarked, $P_1 = \{u_1, \ldots, u_k\}$; we shall now show that the ordering relation $<_1$ and the labelling function l_1 of P_1 may be recovered from γ_1. By the definition of the transition relations over $\mathbf{Pos}[\mathbf{A_S}]$, it is easy to see that, for each $u \in P_1$, $l_1(u) = a$ iff $\langle S(a), u\rangle = \langle e_i, u_i\rangle$ for some $1 \leq i \leq k$. We shall now concentrate on showing how $<_1$ may be recovered from γ_1. As we are dealing with finite partial orders, $<_1$ is completely determined by the *covering relation* over $\mathbb{P}_1$; for all $u, v \in P_1$, u is covered by v iff $u <_1 x <_1 v$, for no $x \in P_1$. Intuitively, γ_1 begins with a block of start-moves followed alternately by blocks of end-moves and blocks of start-moves and then ends with a final block of end-moves. Then the events in P_1 appearing in the first block of start-moves will correspond to the minimal elements in $\mathbb{P}_1$, those appearing in the last block to the maximal elements of $\mathbb{P}_1$ and, for each intervening block of end-moves followed by a block of start-moves, the events appearing in the block of end-moves will correspond to events in $\mathbb{P}_1$ covered by those appearing in the block of start-moves. We shall thus be able to recover from γ_1 the covering relation in $\mathbb{P}_1$; this is sufficient to recover $<_1$.

Let $\prec_1$ be the relation over P_1 such that, for all $u, v \in P_1$, $u \prec_1 v$ iff

(FS) there exists a subword $\langle e_h, u_h\rangle \cdots \langle e_{h+r}, u_{h+r}\rangle\langle e_{h+r+1}, u_{h+r+1}\rangle \cdots \langle e_l, u_l\rangle$ of γ_1, with $h < l$ and $r \geq 0$, such that

(i) $u_h = u$ and $u_l = v$,

(ii) for all $i \leq r$, $e_{h+i} = F(a_i)$ for some $a_i \in \mathbf{A}$, and

(iii) for all $h + r + 1 \leq j \leq l$, $e_j = S(b_j)$ for some $b_j \in \mathbf{A}$.

Let $\prec_1^+$ denote the transitive closure of $\prec_1$. We claim that

$$u \prec_1^+ v \iff u <_1 v. \tag{1}$$

- We prove, first of all, that $u \prec_1^+ v$ implies $u <_1 v$, i.e. that $\prec_1^+$ is *sound* with respect to $<_1$. Assume that $u, v \in P_1$ and $u \prec_1 v$. Then there exists a subsequence of Γ

$$I\!P_h \xrightarrow{\langle F(a_{h+1}), u_{h+1}\rangle} I\!P_{h+1} \xrightarrow{\langle F(a_{h+2}), u_{h+2}\rangle} \cdots I\!P_{h+r+1} \xrightarrow{\langle S(b_{h+r+2}), u_{h+r+2}\rangle} \cdots \xrightarrow{\langle S(b_{l+1}), u_{l+1}\rangle} I\!P_{l+1},$$

with $h < l$, $r \geq 0$, such that $u_{h+1} = u$, $u_{l+1} = v$ and

$$\langle F(a_{h+1}), u_{h+1}\rangle \cdots \langle F(a_{h+r+1}), u_{h+r+1}\rangle \langle S(b_{h+r+2}), u_{h+r+2}\rangle \cdots \langle S(b_{l+1}), u_{l+1}\rangle$$

satisfies the property (FS). Assume, towards a contradiction, that $u \not<_1 v$. Then, by repeatedly applying lemma 4.1, we have that, for some $I\!P'$ and $I\!P''$,

$$I\!P_h \xrightarrow{\omega_1} I\!P' \xrightarrow{\langle F(a_{h+1}), u\rangle} I\!P_{h+r+1} \xrightarrow{\langle S(b_{l+1}), v\rangle} I\!P'' \xrightarrow{\omega_2} I\!P_{l+1},$$

with $\omega_1 = \langle F(a_{h+2}), u_{h+2}\rangle \cdots \langle F(a_{h+r+1}), u_{h+r+1}\rangle$ and $\omega_2 = \langle S(b_{h+r+2}), u_{h+r+2}\rangle \cdots \langle S(b_l), u_l\rangle$ As $u \not<_1 v$, by lemma 4.2 there exists $\bar{I\!P}$ such that

$$I\!P' \xrightarrow{\langle S(b_{l+1}), v\rangle} \bar{I\!P} \xrightarrow{\langle F(a_{h+1}), u\rangle} I\!P''.$$

Thus $\gamma = \langle e_1, u_1\rangle \cdots \langle e_h, u_h\rangle \omega_1 \langle S(b_{l+1}), v\rangle \langle F(a_{h+1}), u\rangle \omega_2 \langle e_{l+2}, u_{l+2}\rangle \cdots \langle e_k, u_k\rangle \in CT(I\!P_1)$ and $\gamma \ll \gamma_1$. However, this contradicts the minimality of γ_1 in $CT(I\!P_1)$ with respect to $\ll$. Hence $u \prec_1 v$ implies $u <_1 v$; by transitivity, $u \prec_1^+ v$ implies $u <_1 v$.

- We now prove that $u <_1 v$ implies $u \prec_1^+ v$, i.e. that $\prec_1^+$ is *complete* with respect to $<_1$. The proof of this fact will depend on the assumption that $I\!P_1$ is a series-parallel poset. Assume that u is covered by v in $I\!P_1$. Then, by lemma 4.3, $\langle F(a), u\rangle = \langle e_h, u_h\rangle$ and $\langle S(b), v\rangle = \langle e_l, u_l\rangle$, for some h, l such that $h < l$. If the subword of γ

$$\langle e_h, u_h\rangle \cdots \langle e_l, u_l\rangle$$

has the (FS) property then we have that $u \prec_1 v$. Otherwise, there exist h_1 and h_2, with $h < h_1 < h_2 < l$, such that $e_{h_1} = S(a_{h_1})$, $e_{h_2} = F(a_{h_2})$, for some $a_{h_1}, a_{h_2} \in \mathbf{A}$, and $\langle e_h, u_h\rangle \cdots \langle e_{h_1}, u_{h_1}\rangle$ and $\langle e_{h_2}, u_{h_2}\rangle \cdots \langle e_l, u_l\rangle$ have the (FS) property. By the definition of $\prec_1$, we have that $u = u_h \prec_1 u_{h_1}$ and $u_{h_2} \prec_1 u_l = v$. By the soundness of $\prec_1$ with respect to $<_1$, we have that $u <_1 u_{h_1}$ and $u_{h_2} <_1 v$. Hence we have that

$$u <_1 v, \; u <_1 u_{h_1} \text{ and } u_{h_2} <_1 v.$$

As $I\!P_1$ is a series-parallel poset, by applying the (N) property in proposition 2.1 we derive that

(a) $u_{h_1} <_1 v$ or

(b) $u <_1 u_{h_2}$ or

(c) $u_{h_2} <_1 u_{h_1}$.

We examine each possibility in turn, showing that each of them leads to a contradiction. If (a) holds then we have that $u <_1 u_{h_1} <_1 v$, contradicting the hypothesis that u is covered by v in $\mathbb{P}_1$. Similarly, if (b) holds. If (c) holds then $u_{h_2} <_1 u_{h_1}$, but this contradicts the assumption that $h_1 < h_2$, i.e. that the start of event u_{h_1} occurs before the end of event u_{h_2} in γ_1. Hence we have shown that if u is covered by v in $\mathbb{P}_1$ then $u \prec_1 v$; by transitivity, $u <_1 v$ implies $u \prec_1^+ v$.

Thus we have shown that $u <_1 v$ iff $u \prec_1^+ v$, for all $u, v \in P_1$. As $\mathbb{P}_1 \approx_{ST} \mathbb{P}_2$, there exists $\gamma_2 = \langle f_1, v_1\rangle \cdots \langle f_n, v_n\rangle \in CT(\mathbb{P}_2)$ such that:

(i) $k = n$, $\pi_1(\gamma_1) = \pi_1(\gamma_2)$, and

(ii) for all $1 \leq i < j \leq k$, $u_i = u_j$ iff $v_i = v_j$.

Again, we have that $P_2 = \{v_1, \ldots, v_n\}$. We may now define $\prec_2$ from γ_2 as we did for $\prec_1$ from γ_1 and, as γ_2 is also minimal in $CT(\mathbb{P}_2)$, by symmetry and (1) we obtain that $x <_2 y$ iff $x \prec_2^+ y$, for all $x, y \in P_2$. Let us now define $\phi : P_1 \to P_2$ by $\phi(u) = x$ iff there exists $1 \leq i \leq k$ such that $u_i = u$ and $v_i = x$. Then, ϕ is a well-defined function by clause (ii) above and it is label-preserving by clause (i) and the definition of the transition relations. It is easy to see that ϕ is also bijective by clause (ii). Moreover, by construction, ϕ is such that $u \prec_1 v$ iff $\phi(u) \prec_2 \phi(v)$, for all $u, v \in P_1$. Hence, by claim (1) and transitivity, we have that $u <_1 v$ iff $\phi_1 <_2 \phi(v)$, for all $u, v \in P_1$. Thus $\mathbb{P}_1 \cong \mathbb{P}_2$. □

The following result is an immediate corollary of the above theorem.

Corollary 4.1 *Let $\alpha, \beta \in SP$. Then $\alpha = \beta$ iff $\alpha \approx_{ST} \beta$.*

ST-trace equivalence can be inherited by $\mathcal{SP}$ via the semantic map $[\![\cdot]\!]$ in a straightforward way; for each $p, q \in \mathcal{SP}$, we write $p \approx_{ST} q$ iff $[\![p]\!] \approx_{ST} [\![q]\!]$. By using the results presented in §3 and the above theorem and corollary, it is now possible to provide a complete axiomatization of ST-trace equivalence over $\mathcal{SP}$. Moreover, as stated by the following theorem, $\approx_{ST}$ gives yet another characterization of the largest congruence over $\mathcal{SP}$ which is preserved by refinement and is contained in $\sim$.

Theorem 4.2 *For all $p, q \in \mathcal{SP}$, $p \approx_{ST} q$ iff $p \sim^{\rho} q$ iff $p =_E q$.*

Proof: The claim follows by the above corollary, Theorem 2.1 and theorem 3.3. □

ST-trace equivalence, $\approx_{ST}$, could be defined directly on the language SP without much difficulty; however, the proof of the main result of this section has been greatly simplified by working with labelled posets rather than with terms in SP.

5 Conclusions

In this paper, we have presented a behavioural characterization of the class of series-parallel pomsets, [Gi84], based on a natural interleaving testing scenario. This has been obtained by showing that the model of series-parallel pomsets is fully-abstract with respect to the behavioural equivalence obtained by applying Abramsky's testing scenario for bisimulation equivalence, [Ab87], in all refinement contexts, [AH89], [GG88], [NEL88]. Following Milner and Plotkin's paradigm, this result justifies the use of this simple mathematical model based on partial orders in giving semantics to the basic process algebra studied in this paper. Moreover, we have shown that identity over the class of SP pomsets coincides with ST-trace equivalence, [Gl90]. Thus SP pomsets can be made fully abstract by assuming a trace-based notion of observation, albeit one in which beginnings and ends of the same occurrence of an action are explicitly linked. This retrievability result has allowed us to give a complete axiomatic characterization of ST-trace equivalence over the class of SP pomsets. A natural question to ask is whether SP pomsets are completely characterized by their set of *split traces*, see [Va88], [Gl90] and §4. The following conjecture naturally suggests itself.

Conjecture: For all $\alpha, \beta \in SP$, $\alpha = \beta$ iff $\alpha \approx_{2t} \beta$.

All the author's attempts to prove or disprove the above conjecture have so far failed. It is interesting to note that the validity of the above conjecture would have some striking consequences. First of all, it would imply that, for all $p, q \in SP$, $p \sim_t q$ iff $p \approx_{2t} q$, i.e. that timed-bisimulation and split trace equivalence coincide over SP. As it is well-known, this result is *not* true of standard strong bisimulation and trace equivalence because the processes in SP are not *deterministic*, [Mil89], [Va88]. Moreover, by following the proof of the results presented in [AH89], it would be possible to show that equality between SP pomsets is the largest congruence contained in standard interleaving trace equivalence which is preserved by refinement.

The work presented in this paper may be seen as an embryonic attempt at defining a natural testing scenario which justifies the use of partial order semantics without assuming any notion of "causal observation". We have shown that such a testing scenario does exist for the simple model considered in the paper; however, as work by R. van Glabbeek on ST-bisimulation semantics shows, [Gl90], a notion of system testing based on the refinement operator does not suffice to reveal the full-distinguishing power of partial order semantics. The search for a testing scenario which justifies models like *Event Structures* and *Causal Trees* seems to be a very interesting topic for future research.

We end this conclusion with a brief discussion of related work. Precursors of the work presented in §3 are [Gi84], [Ts88], where language equivalence for pomsets and series-parallel pomsets are studied in detail, and recent papers in the literature studying notions of equivalence for concurrent systems which are perserved by refinement of actions [AH89], [GG88], [NEL88], [Gl90]. In all these references, the authors present semantic theories for processes which support refinement of actions. The reference [GG88] gives a good survey of the work in this area; [NEL88] gives a natural fully abstract model for a language incorporating a refinement operator and [AH89] characterizes the largest congruence contained in bisimulation equivalence which is preserved by refinement over a simple process algebra and gives a finite, complete axiomatization for it. In [Gl90], the author studies notions of ST-bisimulation and ST-trace equivalence over prime Event Structures [Win87] and proves that they are both preserved by refinement.

Retrievability results like the one presented in §4 for SP pomsets have been shown in, e.g., [Va88]. There the author shows that deterministic Event Structures are characterized, up to isomorphism, by their set of step-sequences. A similar result is shown for split-traces; this implies that the causal structure of a deterministic concurrent system can be reconstructed by observers which are capable of observing the beginning and the end of events.

Acknowledgement

I should like to thank my supervisor Matthew Hennessy for his guidance, continuous support and the joint work which has led to the development of the main ideas in this paper.

6 References

[Ab87] S. Abramsky, Observation Equivalence as a Testing Equivalence, TCS 53, pp. 225-241, 1987

[Ac90] L. Aceto, Full Abstraction for Series-Parallel Pomsets, Computer Science Report 1/90, University of Sussex, March 1990

[**AH89**] L. Aceto and M. Hennessy, Towards Action-refinement in Process Algebras, Proc. 4^{th} LICS, pp. 138-145, IEEE Computer Society Press, 1989 (Full version to appear in Information and Computation)

[**BC88**] G. Boudol and I. Castellani, Concurrency and Atomicity, TCS 59, pp. 25-84, 1988

[**BHR84**] S. D. Brookes, C. A. R. Hoare and A. W. Roscoe, A Theory of Communicating Sequential Processes, JACM 31,3, pp. 560-599, 1984

[**BK85**] J. A. Bergstra and J. W. Klop, Algebra of Communicating Processes with Abstraction, TCS 37, 1, pp. 77-121, 1985

[**DD89**] P. Darondeau and P. Degano, Causal Trees, Proc. ICALP 89, LNCS 372, pp. 234-248, Springer-Verlag, 1989

[**DDM88**] P. Degano, R. De Nicola and U. Montanari, Partial Ordering Semantics for CCS, Technical Report TR-3/88, Università di Pisa, 1988

[**DH84**] R. De Nicola and M. Hennessy, Testing Equivalences for Processes, TCS 34,1, pp. 83-134, 1984

[**Gi84**] J. L. Gischer, *Partial Orders and the Axiomatic Theory of Shuffle*, Ph. D. Thesis, Stanford University, 1984

[**GG88**] R. van Glabbeek and U. Goltz, Equivalence Notions for Concurrent Systems and Refinement of Actions, Proc. 14^{th} MFCS, LNCS 379, pp. 237-248, Springer-Verlag, 1988

[**Gl90**] R. van Glabbeek, The Refinement Theorem for ST-bisimulation, to appear in Proc. IFIP Working Group, Sea of Galilee, 1990

[**Go88**] U. Goltz, On Representing CCS Programs by Finite Petri Nets, Arbeitspapiere der GMD 290, February 1988

[**Gr81**] J. Grabowski, On Partial Languages, Fundamenta Informaticae IV.2, pp. 427-498, 1981

[**Gro90**] J.F. Groote, A New Strategy for Proving ω-Completeness applied to Process Algebra, Proceedings CONCUR '90, LNCS 458, pp. 314-331, Springer-Verlag, 1990

[**GV87**] R. van Glabbeek and F. Vaandrager, Petri Net Models for Algebraic Theories of Concurrency, Proc. PARLE Conference 1987, LNCS 259, Springer-Verlag, 1987

[H85] M. Hennessy, Acceptance Trees, JACM 32,4, pp. 896-928, 1985

[H88a] M. Hennessy, *Algebraic Theory of Processes*, MIT Press, 1988

[H88b] M. Hennessy, Observing Processes, Proc. REX Workshop 88, LNCS 354, pp. 173-200, Springer-Verlag, 1988

[H88c] M. Hennessy, Axiomatising Finite Concurrent Processes, SIAM Journal on Computing, October 1988

[HM85] M. Hennessy and R. Milner, Algebraic Laws for Nondeterminism and Concurrency, JACM 32,1, pp. 137-161, 1985

[Hoare85] C. A. R. Hoare, *Communicating Sequential Processes*, Prentice-Hall, 1985

[HP79] M. Hennessy and G. Plotkin, Full Abstraction for a Simple Parallel Programming Language, Proc. MFCS, Lecture Notes in Computer Science vol. 74, Springer-Verlag, 1979

[Main88] M. Main, Trace, Failure and Testing Equivalences for Communicating Systems, International Journal of Parallel Programming 16, pp. 383-401, 1988

[Mil77] R. Milner, Fully Abstract Models of Typed Lambda-Calculi, TCS 4, pp. 1-22, 1977

[Mil80] R. Milner, *A Calculus of Communicating Systems*, Lecture Notes in Computer Science Vol. 92, Springer-Verlag, 1980

[Mil83] R. Milner, Calculi for Synchrony and Asynchrony, TCS 25, pp. 267-310, 1983

[Mil89] R. Milner, *Communication and Concurrency*, Prentice-Hall, 1989

[Mol89] F. Moller, *Axioms for Concurrency*, Ph. D. Thesis, University of Edinburgh, 1989

[NEL88] M. Nielsen, U. Engberg and K. S. Larsen, Fully Abstract Models for a Process Language with Refinement, Proc. REX Workshop 88, LNCS 354, pp. 523-548, Springer-Verlag, 1988

[Pa81] D. Park, Concurrency and Automata on Infinite Sequences, Lecture Notes in Computer Science vol. 104, Springer-Verlag, 1981

[Pl77] G. Plotkin, LCF Considered as a Programming Language, TCS 5, pp. 223-255, 1977

[Pl81] G. Plotkin, A Structural Approach to Operational Semantics, Report DAIMI FN-19, Computer Science Dept. , Aarhus University, 1981

[Pr86] V. Pratt, Modelling Concurrency with Partial Orders, International Journal of Parallel Programming 15, pp. 33-71, 1986

[Rei85] W. Reisig, *Petri Nets*, EATCS Monographs on TCS, Springer-Verlag, 1985

[Sto88] A. Stoughton, *Fully Abstract Models of Programming Languages*, Research Notes in TCS, Pitman-Wiley, 1988

[Tau89] D. Taubner, *The Finite Representation of Abstract Programs by Automata and Petri Nets*, LNCS 369, Springer-Verlag, 1989

[Ts88] S. Tschantz, Languages Under Concatenation and Shuffling, Vanderbilt University Report, 1988

[Va88] F. Vaandrager, Determinism $\longrightarrow$ (Event Structure Isomorphism = Step Sequence Equivalence), Report CS-R8839, CWI Amsterdam, October 1988

[Win80] G. Winskel, *Events in Computation*, Ph. D. Thesis, University of Edinburgh, 1980

[Win82] G. Winskel, Event Structure Semantics for CCS and Related Languages, Proc. ICALP 82, LNCS, Springer-Verlag, 1982

[Win87] G. Winskel, Event Structures, Advances in Petri Nets 1986, LNCS 255, pp. 325-392, Springer-Verlag, 1987