INVARIANTS AND PARADIGMS OF CONCURRENCY THEORY

M KOUTNY

Rapporteur: J Harley

INVARIANTS AND PARADIGMS OF CONCURRENCY THEORY[1)]

Ryszard Janicki
Department of Computer Science and Systems
McMaster University
Hamilton, Ontario, Canada, L8S 4K1

Maciej Koutny
Computing Laboratory
The University of Newcastle upon Tyne
Newcastle upon Tyne NE1 7RU, U.K.

ABSTRACT

We introduce a new invariant semantics of concurrent systems which is a direct generalisation of the causal partial order semantics. Our new semantics overcomes some of the problems encountered when one uses causal partial orders alone. We discuss various aspects of the new invariant model. In particular, we outline how the new invariants can be generated by 1-safe inhibitor Petri nets.

---

1) Appeared in the Proceedings of the PARLE'91 Conference, Lecture Notes in Computer Science, Springer, 1991.

1 Introduction

In the development of mathematical models of concurrent behaviours, the concept of partial and total order undoubtedly occupies a central position. Interleaving models use total orders of event occurrences, while so-called 'true concurrency' models use step sequences or causal partial orders (comp. [BD87,Mi80, Ho85, Pr86]). Even more complex structures, such as failures [Ho85] or event-structures [Wi82], are in principle based on the concept of total or partial order. While interleavings and step sequences usually represent executions or observations, the causality relation represents a set of executions or observations. The lack of order between two event occurrences in the case of step sequence is interpreted as simultaneity, while in the case of causality relation is interpreted as independency. Both interleaving and true concurrency models have been developed to a high degree of sophistication providing a framework for specification and verification of concurrent systems. However, some of the behavioural aspects of concurrent behaviour are difficult to tackle in the interleaving or partial order based setting. For instance, the specification of priorities using partial orders alone is rather problematic, in particular, if the events are not instantaneous (see [La85,Ja87,JL88, BK91]). Another example are inhibitor nets (see [Pe81]) which are virtually admired by practitioners and almost completely rejected by theoreticians, in our opinion mainly because their concurrent behaviour cannot be properly defined in terms of causality based structures. We believe that problems of this kind follow from an implicit assumption that all behavioural properties of concurrent systems can be adequately modelled in terms of causality based structures. We claim that the structure of concurrency phenomenon is richer, with causality being only one of several fundamental invariants generated by sets of equivalent executions or observations. In this paper we will show how such invariants can be defined and constructed.

2 Motivation

We start by discussing two specific situations which we believe identify an inherent inability of the causal partial order semantics to properly cope with some of the aspects of the non-sequential behaviour. We will use Petri nets [Pe81,Re85] as the system model, however this does not mean that our approach is restricted to Petri nets. COSY with priorities, or TCSP with priorities could be used as well (comp. [JL88]).

The first example closely follows the discussion in [Ja87,JL88]. We consider a concurrent system *Con* comprising two sequential subsystems A and B such that: (1) A can engage in event a and after that in event b; (2) B can engage either in event b or in event c; (3) the two sequential subsystems synchronise by means of the handshake communication; and (4) the specification of *Con* includes a priority constraint stating that whenever it is possible to execute b then c must not be executed.

The priority Petri net in Figure 1 illustrates this example. We now observe that causal partial orders cannot provide a satisfactory semantical model of *Con*. We first note that in the initial state both events a and c are enabled and can be executed simultaneously (note that the priority constrained is not violated since b is not enabled in the initial state). Thus in any causality based model *Con* generates a causal partial order with one occurrence of a and one occurrence of c such that there is no causal relationship between the two event occurrences. Now, since a and c are in-

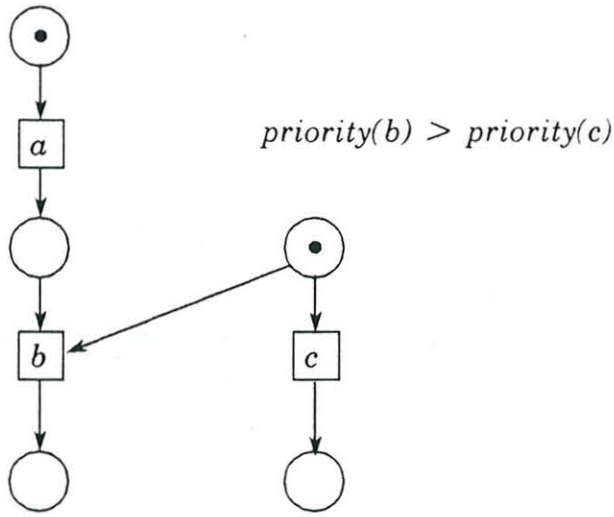

Figure 1

dependent, it should be possible to execute c followed by a, and a followed by c. Whereas the former execution sequence does not violate the priority constraint, the latter does as after executing a event b becomes enabled and c must not be executed. Note that in [BK91] it was observed that whether the simultaneous execution of a and c should be allowed is related to whether or not one can regard a as an event taking some time. If a is instantaneous then the step $\{a,c\}$ should not be allowed, and then a causal partial order semantics of *Con* can be constructed along the lines described in [BK91]. If, however, a cannot be regarded as instantaneous (possibly because it is a compound event) then one should look for an invariant model more expressive than causal partial orders to capture the behaviour of *Con*.

As the second example we consider a system which supports an error recovery mechanism. That mechanism is invoked by an occurrence of a special signalling event, *err*, which may occur simultaneously with any other event in the system. The result of an occurrence of *err* is that: (1) the error recovery procedure is called and its successful completion it signified by an occurrence of a special event *rcv*; and (2) during the error recovery no event in the system is allowed to be executed.

We again observe that the causal partial orders do not provide a satisfactory model of the system's behaviour. For it is possible to execute *err* simultaneously with some other event, say a, and then after the termination of the error recovery procedure to execute event *rcv*. In any causal partial order which might underlay such a system history, the occurrences of *err* and a must be independent, and the occurrence of *rcv* must not precede the occurrence of a. This, however, means that it is possible to execute *err* followed by a and *rcv*, violating (2).

The above two examples show that causal partial orders are not expressive enough to satisfactorily model the invariant properties of certain kinds of concurrent systems. In the rest of this paper we will outline an alternative invariant semantics which overcomes the problems highlighted in the above two examples.

The overall goal of this paper might be explained in the following way. Consider the nets of Figure 2. Two of them, PN_4 and PN_5, are nets employing inhibitor arcs. (An inhibitor arc between place p and transition t means that t can be enabled only if p is unmarked [Pe81].) We want to define an invariant semantics of these nets in such a way that the following would hold (below by a 'complete'

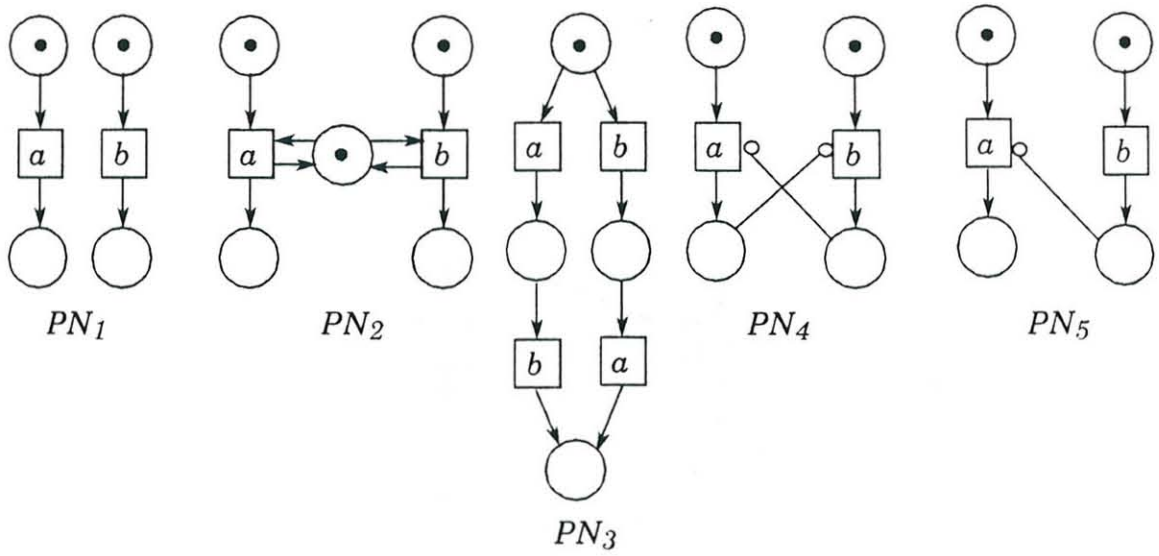

Figure 2

history or execution of net P_i we mean one which involves exactly one occurrence of a and one occurrence of b).

(1) Different nets generate different complete concurrent histories.

(2) Each net except PN_3 generates one complete concurrent history.

(3) In each case a concurrent history is defined on the same level of abstraction as the causal partial order.

Taking into account only complete executions (or observations) expressed in terms of step sequences, we might define the semantics of the nets in the following way. Let $\sigma_1,\sigma_2,\sigma_3$ be the step sequences $\sigma_1=\{a\}\{b\}$, $\sigma_2=\{b\}\{a\}$ and $\sigma_3=\{a,b\}$. Then:

$Steps(PN_1)=\{\sigma_1,\sigma_2,\sigma_3\}$

$Steps(PN_2)=\{\sigma_1,\sigma_2\}$

$Steps(PN_3)=\{\sigma_1,\sigma_2\}$

$Steps(PN_4)=\{\sigma_3\}$

$Steps(PN_5)=\{\sigma_1,\sigma_3\}$.

Step sequences cannot distinguish between PN_2 and PN_3, and do not tell us that each of PN_1, PN_2, PN_4, PN_5 generates in fact only one complete concurrent history. That each of PN_1, PN_4 and PN_5 generates only one complete concurrent history is intuitively obvious (no conflict occurs in these nets). However, this may be not so clear in the case of PN_2. Moreover, one might ask why at all should we distinguish between PN_2 and PN_3. To show that making such a distinction may in some cases be appropriate we consider a program statement:

$a: x:=x+1 \;\&\; b: x:=x+3$

where "&" denotes the *commutativity* operator (see [LH82]) which means that the instructions a and b may be performed in any order but never simultaneously. We believe that this statement should generate *one* concurrent history comprising two essentially equivalent executions, σ_1 and σ_2, rather than two different concurrent histories, one comprising σ_1, the other σ_2. Thus, since PN_2 seems to be a natural implementation of the commutativity operator, it should also generate one complete concurrent history. On the other hand, PN_3 is clearly a net generating two different com-

plete histories. Thus each PN_i ($i=1,2,4,5$) should generate exactly one complete concurrent history H_i, where: $H_1=\{\sigma_1,\sigma_2,\sigma_3\}$, $H_2=\{\sigma_1,\sigma_2\}$, $H_4=\{\sigma_3\}$ and $H_5=\{\sigma_1,\sigma_3\}$; while PN_3 should generate two complete concurrent histories: $H_{31}=\{\sigma_1\}$ and $H_{32}=\{\sigma_2\}$. A question which one might now ask is whether we could define these histories in a more structured and compact way, for example, by using causality-like relations? There are only three causal relationships involving one occurrence of a and one occurrence of b, namely:

c_1 = *a and b are independent*

c_2 = *a precedes b*

c_3 = *a follows b.*

Clearly, c_1 characterises history H_1, c_2 characterises H_{31} and c_3 characterises H_{32}. This means that none of H_2, H_4 and H_5 can be characterised by a suitable causal relationship. To solve the problem we then observe that causality and independency can be characterised in the following way:

If a and b are two events involved in a concurrent history H then

a <u>*precedes*</u> *b* *if in all executions belonging to H, a precedes b.*

a <u>*follows*</u> *b* *if in all executions belonging to H, a follows b.*

If a neither precedes nor follows b, then a and b are <u>*independent*</u>.

By following the above pattern we now can introduce three new invariant relations, called *commutativity*, *synchronisation* and *weak causality*, in the following way:

a <u>*comm*</u> *b* *if in all executions belonging to H, either a precedes b or b precedes a*

a <u>*synch*</u> *b* *if in all executions belonging to H, a is simultaneous with b*

a <u>*wc*</u> *b* *if in all executions belonging to H, a precedes or is simultaneous with b.*

One now may observe that *a* <u>*comm*</u> *b* characterises H_2; *a* <u>*synch*</u> *b* characterises H_4; and *a* <u>*wc*</u> *b* characterises H_5. The new invariant relations can be used to distinguish between the five nets of Figure 2, but it is not at all clear yet whether they would work in the general case. One might also ask several other questions, such as: How can one define commutativity, synchronisation and weak causality in the general case? What is their relationship to the causality relation as well as their mutual relationship? Are there other relations of this kind? These are examples of questions we will try to answer in this paper. As we already mentioned, the distinction between the concurrent histories generated by PN_2 and PN_3 *may or may not be* desirable, depending on the intended interpretation of the nets. Another question which seems to be interesting in this context is whether there is a formal mechanism which, when switched on makes PN_2 and PN_3 semantically different, and when switched off makes them semantically identical. It is then worth observing that under the assumption that for every allowed concurrent history: *the existence of the executions in the opposite orders implies the existence of a simultaneous execution*, PN_2 and PN_3 become equivalent as H_2 is no longer a valid history (we have to decompose it onto H_{31} and H_{32}). We will call such rules *paradigms*, and show how they can be defined and used.

3 The Model

The model we are going to develop is a three-level model: Systems-Invariants-Observations, and we will proceed from the bottom (i.e. the observation level) to the top of the hierarchy. In this paper we will focus on the invariant level. We will provide only the most basic results concerning the observation level (for more details see [JK90,JK90a]), while the system level will be considered in an informal manner at the end of this paper.

3.1 Observations

We define observation as an abstract model of execution. More precisely, by an observation we will mean a special report supplied by an observer who can perceive the evolution of a concurrent system. Such an observer has to fill in a (possibly infinite) matrix with rows and columns being indexed by event occurrences. The observer is supposed to fill in the entire matrix except the diagonal using only three symbols: $\rightarrow$, $\leftarrow$ and $\leftrightarrow$, with $\rightarrow$ denoting precedence, $\leftarrow$ following, and $\leftrightarrow$ simultaneity. (How the observer makes his judgement is beyond our interest.) Together with a natural interpretation of the precedence relation this means that observations can be represented by *partially ordered sets* of event occurrences, where ordering represents *precedence*, and incomparability represents *simultaneity*.

A *partially ordered set* (or *poset*) is a pair $po=(X,R)$, where X is a non-empty set and $R\subseteq X\times X$ is an irreflexive ($\neg aRa$) and transitive ($aRb \wedge bRc \Rightarrow aRc$) relation. We say po is *total* if for all different $a,b\in X$, aRb or bRa. We also denote: $dom(po)=X$, $\rightarrow_{po}=R$, $\leftarrow_{po}=R^{-1}$ and $\leftrightarrow_{po}=\{(a,b)\in X\times X \mid a\neq b \wedge \neg aRb \wedge \neg bRa\}$. Not all partial orders may be interpreted as possible observations. The additional properties we require are that: (1) the observer perceives only a single thread of time, and can only observe a finite number of events in a finite period of time and that (2) an event can last only for a finite period of time. It can be shown that (1) and (2) lead to the following definition of an observation of a concurrent history (see [JK90,JK90a] for details).

Observation is an initially finite interval order of event occurrences.

Note that a poset is initially finite if for every $a\in dom(po)$, the set $\{b\in dom(po) \mid \neg a\rightarrow_{po}b\}$ is finite, and that a poset po is an *interval order* if $(a\rightarrow_{po}b \wedge c\rightarrow_{po}d) \Rightarrow (a\rightarrow_{po}d \vee c\rightarrow_{po}b)$. The definition of interval order is taken from [Fi70], however the origin of this concept can be traced back to Wiener's 1914 paper [Wn14], where he considered interval orders as a way to analyse temporal events, each event occurring over some finite time span. The main characterisation of interval orders is given below.

Theorem 1 [Fi70]
A countable poset po is an interval order if and only if there are ϕ, $\rho : dom(po) \rightarrow Reals$ such that $\rho(a)>0$ for all a, and if $a,b\in dom(po)$ then: $a\rightarrow_{po}b \Leftrightarrow \phi(a)+\rho(a)<\phi(b)$. □

The above result was strengthened in [JK90a] by showing that we can additionally require that ϕ is injective. The general properties of interval orders and their applications to the measurement theory were discussed in [Fi85], while the application of interval orders to model observations of concurrent histories was discussed in [JK90,JK90a].

A *step sequence* is an initially finite poset *po* such that $(a \leftrightarrow_{po} b \wedge b \leftrightarrow_{po} c \wedge a \neq c) \Rightarrow a \leftrightarrow_{po} c$, while an *interleaving sequence* is an initially finite total order. Let Obs, Obs_{step}, Obs_{itl} denote respectively the sets of observations, step sequences and interleaving sequences. We have $Obs_{itl} \subsetneq Obs_{step} \subsetneq Obs$, and throughout the rest of this paper, o (with an index, if necessary) will usually range over Obs.

3.2 Invariants and Histories

A description of a concurrent system solely in terms of the observations it may generate is unsatisfactory for many reasons. In fact, any argument made in favour of causal partial orders existing in the literature (see, for instance, [BD85]), can also be used to support the introduction of the new invariants. We will first focus on the relationship between different observations of a concurrent history, where a concurrent history is essentially an invariant or a *set of invariants* satisfied by all its observations. It will be shown that the familiar causality relation is just *one* of many possible invariant relations. There are, of course, different ways in which an invariant might be defined for a given set of observations, depending on the specific kind of properties of the system one is interested in. In this paper we restrict ourselves to invariants which seem to be the most basic ones.

A *report set* is a non-empty set Δ of observations such that $dom(o_1) = dom(o_2)$ for all $o_1, o_2 \in \Delta$. We will denote by $dom(\Delta)$ the common domain of the observations in Δ. Note that a report set may be considered as the first approximation of a concurrent history.

Let Δ be a report set with the domain Σ. A *simple (binary) relational invariant* of Δ, is a relation $I \subseteq \Sigma \times \Sigma$ which can be characterised by:

$(a,b) \in I :\Leftrightarrow a \neq b \wedge \forall o \in \Delta.\ \Phi(a,b,o)$,

where $\Phi(a,b,o)$ is any formula derived from the following grammar:

$\Phi ::= true \mid false \mid a \rightarrow_o b \mid a \leftarrow_o b \mid a \leftrightarrow_o b \mid \neg\Phi \mid \Phi \vee \Phi \mid \Phi \wedge \Phi$.

Some of the basic terms of the above grammar are redundant, e.g., $a \leftarrow_o b$ is equivalent to $\neg(a \rightarrow_o b \vee a \leftrightarrow_o b)$. However, this does not cause any problems, while increases readability. Let $SRI(\Delta)$ denote the set of all simple (binary) relational invariants of Δ, and let $\rightarrow_\Delta$, $\leftarrow_\Delta$, $\leftrightarrow_\Delta$, $\rightleftharpoons_\Delta$, $\nearrow_\Delta$, $\nwarrow_\Delta$ be binary relations on Σ such that for all $a,b \in \Sigma$,

$a \rightarrow_\Delta b \ :\Leftrightarrow\ a \neq b \wedge \forall o \in \Delta.\ a \rightarrow_o b$

$a \leftarrow_\Delta b \ :\Leftrightarrow\ a \neq b \wedge \forall o \in \Delta.\ a \leftarrow_o b$

$a \leftrightarrow_\Delta b \ :\Leftrightarrow\ a \neq b \wedge \forall o \in \Delta.\ a \leftrightarrow_o b$

$a \rightleftharpoons_\Delta b \ :\Leftrightarrow\ a \neq b \wedge \forall o \in \Delta.\ a \rightarrow_o b \vee a \leftarrow_o b$

$a \nearrow_\Delta b \ :\Leftrightarrow\ a \neq b \wedge \forall o \in \Delta.\ a \rightarrow_o b \vee a \leftrightarrow_o b$

$a \nwarrow_\Delta b \ :\Leftrightarrow\ a \neq b \wedge \forall o \in \Delta.\ a \leftarrow_o b \vee a \leftrightarrow_o b$.

The relations $\rightarrow_\Delta$, $\leftarrow_\Delta$ are called *causalities*, $\rightleftharpoons_\Delta$ *commutativity*, $\leftrightarrow_\Delta$ *synchronisation*, and $\nearrow_\Delta$, $\nwarrow_\Delta$ *weak causalities*. In the sequel we will use $\rightarrow, \leftarrow, \leftrightarrow, \rightleftharpoons, \nearrow, \nwarrow$ to denote mappings, called *invariants*, which for every report set return respectively $\rightarrow_\Delta, \leftarrow_\Delta, \leftrightarrow_\Delta, \rightleftharpoons_\Delta, \nearrow_\Delta, \nwarrow_\Delta$. The set of all invariants will be denoted by SRI.

Proposition 2

For every report set Δ, $SRI(\Delta) = \{\varnothing, \rightarrow_\Delta, \leftarrow_\Delta, \leftrightarrow_\Delta, \rightleftharpoons_\Delta, \nearrow_\Delta, \nwarrow_\Delta, \Sigma \times \Sigma - id_\Sigma\}$, and there is Δ such that $SRI(\Delta)$ consists of eight different relations. □

Proposition 3

$\leftarrow_\Delta = (\rightarrow_\Delta)^{-1}$, $\nwarrow_\Delta = (\nearrow_\Delta)^{-1}$, $\rightarrow_\Delta = \nearrow_\Delta \cap \rightleftharpoons_\Delta$ and $\leftrightarrow_\Delta = \nearrow_\Delta \cap \nwarrow_\Delta$. □

Due to the symmetry present in $SRI(\Delta)$ one can in fact consider only four non-trivial invariants, namely $\rightarrow_\Delta$, $\leftrightarrow_\Delta$, $\rightleftharpoons_\Delta$ and $\nearrow_\Delta$. Furthermore, $\rightarrow_\Delta$ and $\leftrightarrow_\Delta$ may be expressed in terms of $\nearrow_\Delta$ and $\rightleftharpoons_\Delta$, so it seems reasonable to try to find a possibly smallest sets of invariants from which all the relations in $SRI(\Delta)$ could be generated.

A *signature* of a non-empty family F of report sets is a set of invariants $S \subseteq SRI$ such that for all $\Delta, \Delta_o \in F$ we have:
$(dom(\Delta) = dom(\Delta_o) \wedge \forall I \in S.\ I(\Delta) = I(\Delta_o)) \Rightarrow (\forall I \in SRI.\ I(\Delta) = I(\Delta_o))$.
A signature is *universal* if F is the family of all report sets. Moreover, a signature S of F is *minimal* if (1) no proper subset of S is a signature of F, and (2) for every $J \in S$ and every $I \in SRI\text{-}S$, if $I(\Delta) \subseteq J(\Delta)$ for all report sets Δ, then $(S\text{-}\{J\}) \cup \{I\}$ is not a signature of F. I.e., a signature is minimal if it cannot be 'reduced' by removing any of its invariants (see (1)) or by replacing any invariant by a 'weaker' one (see (2)).

Proposition 4

$\{\nearrow, \rightleftharpoons\}$ and $\{\nwarrow, \rightleftharpoons\}$ are the only minimal universal signatures. □

A history is a report set Δ which is a complete (w.r.t. certain viewpoint) representation of some phenomenon underlying the reports of Δ. This completeness is to be captured by requiring that Δ includes all reports satisfying the relevant properties which can be attributed to the report sets. In our approach, these properties are the domain of Δ, $dom(\Delta)$, and the simple report invariants generated by Δ, $SRI(\Delta)$.

For every $I \in SRI$, let Φ_I denote any formula (see the definition of a simple relational invariant and Proposition 2) such that $(a,b) \in I(\Delta) \Leftrightarrow \forall o \in \Delta.\ \Phi_I(a,b,o)$. Let Δ be a report set and $S \subseteq SRI$. The *S-closure* of Δ, denoted $\Delta^{\langle S \rangle}$, is the set comprising all observations o such that $dom(o) = dom(\Delta)$ and for all $I \in S$, $(a,b) \in I(\Delta) \Rightarrow \Phi_I(a,b,o)$.

Proposition 5

(1) $\Delta \subseteq \Delta^{\langle S \rangle}$.

(2) If S is a universal signature then $\Delta^{\langle S \rangle} = \Delta^{\langle SRI \rangle}$. □

Consider a report set $\Delta = \{o_1, o_2\}$, where o_1 and o_2 are as in Figure 3. Then $a \rightleftharpoons_\Delta b$, $a \rightleftharpoons_\Delta c$ and $b \rightleftharpoons_\Delta c$. Hence $\Delta^{\langle \rightleftharpoons \rangle} = \{o_1, o_2, o_3, o_4, o_5, o_6\}$, where the o_i $(i = 3,4,5,6)$ are shown in Figure 3. Thus $\Delta \subsetneq \Delta^{\langle \rightleftharpoons \rangle}$. Moreover, $\Delta^{\langle \rightleftharpoons \rangle} = \Delta^{\langle SRI \rangle}$. We now can introduce the central notion of this paper.

A history is a non-empty report set Δ such that $\Delta = \Delta^{\langle SRI \rangle}$.

I.e., a history is a report set which is fully characterised by the invariants it generates. Thus if Δ is a history, denoted $\Delta \in Hist$, then the following essentially describe the same thing.

Δ

$(\varnothing, \rightarrow_\Delta, \leftarrow_\Delta, \leftrightarrow_\Delta, \rightleftharpoons_\Delta, \nearrow_\Delta, \nwarrow_\Delta, \Sigma \times \Sigma\text{-}id_\Sigma)$

$(\nearrow_\Delta, \rightleftharpoons_\Delta)$

$(\nwarrow_\Delta, \rightleftharpoons_\Delta)$

$(I_1(\Delta), \ldots, I_k(\Delta))$ where $\{I_1, \ldots, I_k\}$ is a signature of any F such that $\Delta \in F$.

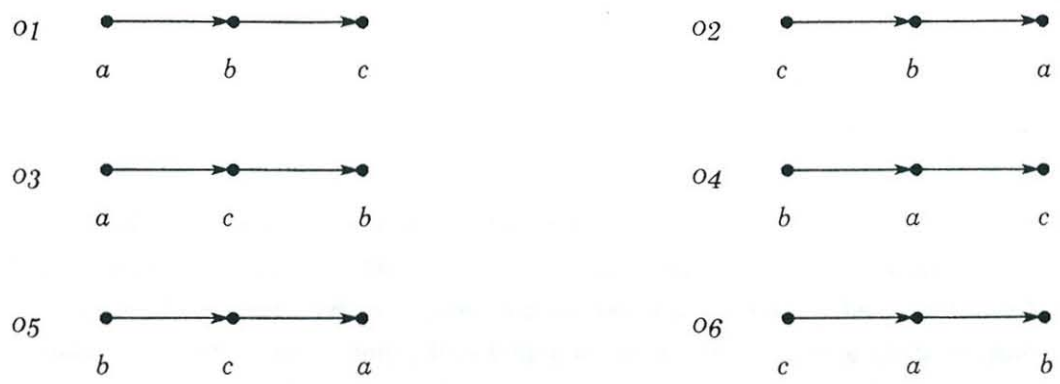

Figure 3

In concurrency theory, the causality relation is sometimes treated as an invariant, and sometimes as the set of all observations (step sequences or interleavings) it generates. We have just shown that this dual treatment can be generalised to other invariants in *SRI*.

3.3 Components and Paradigms

Let Δ be a concurrent history. The set $SRI(\Delta)$ can be treated as any other finite family of sets. In particular, we can find all the components defined by this family, as shown in Figure 4. There are seven non-empty components (non-empty means that there is Δ such that all seven components are non-empty), and we will denote $CSRI(\Delta)=\{\rightarrow_\Delta,\leftarrow_\Delta,\leftrightarrow_\Delta,\leftrightarrows_\Delta,\underline{\rightarrow}_\Delta,\underline{\leftarrow}_\Delta,\|_\Delta\}$.

A formula which says that a given relationship between two event occurrences a and b has been observed in Δ is called a *simple trait*. There are three simple traits: $\psi_\rightarrow \equiv \exists o\in\Delta.\ a\rightarrow_o b$, $\psi_\leftarrow \equiv \exists o\in\Delta.\ a\leftarrow_o b$ and $\psi_\leftrightarrow \equiv \exists o\in\Delta.\ a\leftrightarrow_o b$. One can easily show that the relations in $CSRI(\Delta)$ can be defined as conjunctions of simple traits and their negations.

Proposition 6

For every $a,b\in dom(\Delta)$, we have

$a\rightarrow_\Delta b \Leftrightarrow \psi_\rightarrow \wedge \neg\psi_\leftarrow \wedge \neg\psi_\leftrightarrow$

$a\leftarrow_\Delta b \Leftrightarrow \neg\psi_\rightarrow \wedge \psi_\leftarrow \wedge \neg\psi_\leftrightarrow$

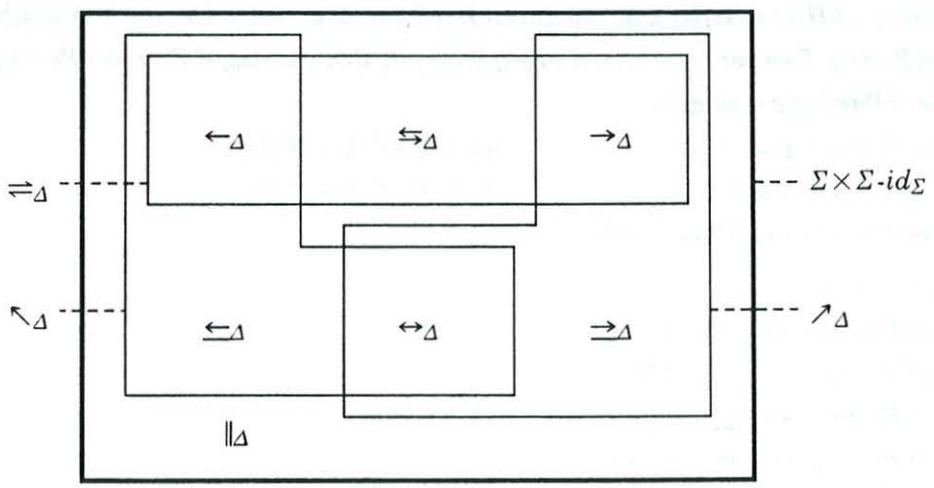

Figure 4

$a \leftrightarrow_\Delta b \;\Leftrightarrow\; \neg\psi_\rightarrow \wedge \neg\psi_\leftarrow \wedge \psi_\leftrightarrow$
$a \leftrightarrows_\Delta b \;\Leftrightarrow\; \psi_\rightarrow \wedge \psi_\leftarrow \wedge \neg\psi_\leftrightarrow$
$a \underline{\rightarrow}_\Delta b \;\Leftrightarrow\; \psi_\rightarrow \wedge \neg\psi_\leftarrow \wedge \psi_\leftrightarrow$
$a \underline{\leftarrow}_\Delta b \;\Leftrightarrow\; \neg\psi_\rightarrow \wedge \psi_\leftarrow \wedge \psi_\leftrightarrow$
$a \| _\Delta b \;\Leftrightarrow\; \psi_\rightarrow \wedge \psi_\leftarrow \wedge \psi_\leftrightarrow$. □

Since we have $\rightarrow_\Delta = (\leftarrow_\Delta)^{-1}$ and $\underline{\rightarrow}_\Delta = (\underline{\leftarrow}_\Delta)^{-1}$ we need to discuss only five components: $\rightarrow_\Delta$, $\|_\Delta$, $\leftrightarrows_\Delta$, $\leftrightarrow_\Delta$ and $\underline{\rightarrow}_\Delta$. The first component (and also an invariant), $\rightarrow_\Delta$, is *causality*. The second component, $\|_\Delta$, should be interpreted as *concurrency* (two events can be observed simultaneously and in both orders). Both causality and concurrency can be found in the models supporting the notion of true concurrency. The third component, $\leftrightarrows_\Delta$, represents what is usually referred to as *interleaving* (two events can be observed in both orders, but not simultaneously), and is usually dealt with on the level of observations rather than invariants. The fourth component (and also an invariant), $\leftrightarrow_\Delta$, can be interpreted as *synchronisation*. It is currently introduced only in its implicit form, e.g., as a silent action in CCS [Mi80]. The fifth component, $\underline{\rightarrow}_\Delta$, is not to our knowledge a part of any of the existing models. It captures *disabling* of one event by another event and was discussed in [Ja87] from where we took its intuitive meaning. As one now may see, the five components describe quite precisely the semantics of the nets of Figure 2, namely $a\|_{H_1}b$, $a\leftrightarrows_{H_2}b$, $a\rightarrow_{H_{31}}b$, $b\rightarrow_{H_{32}}a$, $a\leftrightarrow_{H_4}b$ and $a\underline{\rightarrow}_{H_5}b$.

The approach to concurrency which is based entirely on the concept of causality relation requires that for every concurrent history the following holds: if two event occurrences can be observed simultaneously, then they can also be observed in both orders, and vice versa. This means that every concurrent history besides being invariant-closed must also satisfy the following: $(\exists o\in\Delta.\ a\leftrightarrow_o b) \Leftrightarrow (\exists o\in\Delta.\ a\rightarrow_o b) \wedge (\exists o\in\Delta.\ a\leftarrow_o b)$. Note that this formula is built from simple traits. In general, any formula built in this way will be called a *paradigm*, and will characterise the internal structure of of concurrent histories.

Formally, the *paradigms*, $\omega\in Par$, are given by the following syntax.

$\omega ::= true \mid false \mid \psi_\rightarrow \mid \psi_\leftarrow \mid \psi_\leftrightarrow \mid \neg\omega \mid \omega\vee\omega \mid \omega\wedge\omega \mid \omega\Rightarrow\omega$

The evaluation of the formulas $\omega\in Par$ follows the standard rules [Mo76]. Note that in this grammar we need all three basic terms $\psi_\rightarrow$, $\psi_\leftarrow$ and $\psi_\leftrightarrow$.

A history $\Delta\in Hist$ *satisfies* a paradigm $\omega\in Par$ if for all $a,b\in dom(\Delta)$, $a\neq b \Rightarrow \omega(a,b,\Delta)$. We denote this by $\Delta\in Par(\omega)$. Two paradigms, ω and ω_o, are *equivalent*, $\omega\sim\omega_o$, if $Par(\omega)=Par(\omega_o)$. Let $\omega_i\ (i=1,...,5)$ be the following paradigms:

$\omega_1 = \psi_\leftrightarrow \Rightarrow \psi_\rightarrow \vee \psi_\leftarrow$ $\qquad$ $\omega_2 = \psi_\rightarrow \wedge \psi_\leftarrow \Rightarrow \psi_\leftrightarrow$
$\omega_4 = \psi_\rightarrow \Rightarrow \psi_\leftarrow \vee \psi_\leftrightarrow$ $\qquad$ $\omega_3 = \psi_\rightarrow \wedge \psi_\leftrightarrow \Rightarrow \psi_\leftarrow$
$\omega_5 = \psi_\rightarrow \wedge \psi_\leftarrow \wedge \psi_\leftrightarrow \Rightarrow false$

Proposition 7

(1) $\Delta\in Par(\omega_1) \Leftrightarrow \leftrightarrow_\Delta = \varnothing$.
(2) $\Delta\in Par(\omega_2) \Leftrightarrow \leftrightarrows_\Delta = \varnothing$.
(3) $\Delta\in Par(\omega_3) \Leftrightarrow \underline{\rightarrow}_\Delta = \underline{\leftarrow}_\Delta = \varnothing$.
(4) $\Delta\in Par(\omega_4) \Leftrightarrow \rightarrow_\Delta = \leftarrow_\Delta = \varnothing$.
(5) $\Delta\in Par(\omega_5) \Leftrightarrow \|_\Delta = \varnothing$. □

Proposition 8 (equality up to ~)
$Par = \{\omega_{i_1} \wedge \ldots \wedge \omega_{i_k} \mid k \leq 5 \wedge i_j \leq 5\}$. □

From the last proposition it follows that we have $2^5 = 32$ different paradigms. However, the nature of problems considered in Computer Science is such that two of the ω_i's may be safely rejected. The first ω_i that we reject is ω_4 since it rules out causality and hence invalidates the sequential composition construct. For a similar reason we reject ω_5 since it is not compatible with the standard parallel composition operation. Thus we consider $2^3 = 8$ paradigms:

$\pi_1 = true$ $\quad \pi_3 = \omega_2$ $\quad \pi_5 = \omega_1 \wedge \omega_2$ $\quad \pi_7 = \omega_2 \wedge \omega_3$

$\pi_2 = \omega_1$ $\quad \pi_4 = \omega_3$ $\quad \pi_6 = \omega_1 \wedge \omega_3$ $\quad \pi_8 = \omega_1 \wedge \omega_2 \wedge \omega_3$

Proposition 9 (relationship between components and paradigms)

(1) $\Delta \in Par(\pi_1)$.

(2) $\Delta \in Par(\pi_2) \Leftrightarrow \leftrightarrow_\Delta = \emptyset$.

(3) $\Delta \in Par(\pi_3) \Leftrightarrow \leftrightarrows_\Delta = \emptyset$.

(4) $\Delta \in Par(\pi_4) \Leftrightarrow \underline{\rightarrow}_\Delta = \emptyset$.

(5) $\Delta \in Par(\pi_5) \Leftrightarrow \leftrightarrow_\Delta = \leftrightarrows_\Delta = \emptyset$.

(6) $\Delta \in Par(\pi_6) \Leftrightarrow \leftrightarrow_\Delta = \underline{\rightarrow}_\Delta = \emptyset$.

(7) $\Delta \in Par(\pi_7) \Leftrightarrow \leftrightarrows_\Delta = \underline{\rightarrow}_\Delta = \emptyset$.

(8) $\Delta \in Par(\pi_8) \Leftrightarrow \leftrightarrow_\Delta = \leftrightarrows_\Delta = \underline{\rightarrow}_\Delta = \emptyset$. □

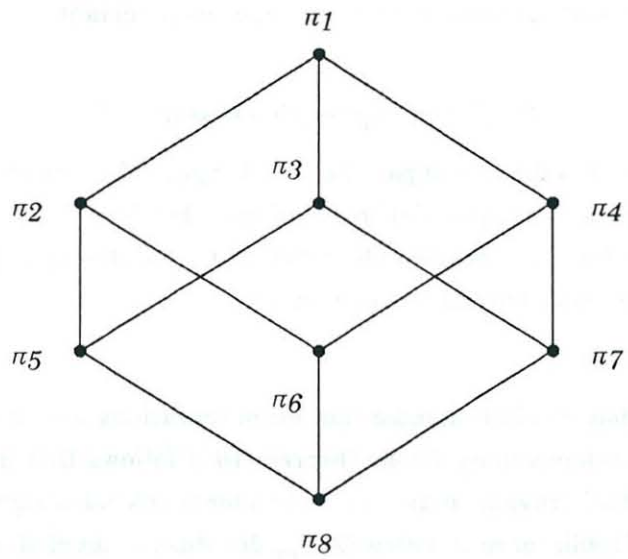

Figure 5

We obtain a hierarchy of eight fundamental paradigms of concurrency shown in Figure 5. In this paper we will only discuss π_1,π_3 and π_8. Paradigm π_1 simply admits all concurrent histories. The most restrictive paradigm, π_8, is the paradigm adopted by the models supporting true concurrency semantics. As we pointed out earlier on, this paradigm has given rise to a number of elegant theories in the field of concurrency, however, it has some limitations such as an inability to model some aspects of systems with priorities. In the next section we will show that π_3 allows us to provide an invariant semantics for inhibitor nets, as well as for priority systems. The following major result characterises minimal signatures of the eight paradigms.

Theorem 10

(1) $\{\nearrow, \rightleftharpoons\}$ is a minimal signature for $Par(\pi_1)$ and $Par(\pi_2)$.

(2) $\{\rightleftharpoons, \rightarrow\}$ is a minimal signature for $Par(\pi_4)$ and $Par(\pi_6)$.

(3) $\{\rightarrow, \nearrow\}$ is a minimal signature for $Par(\pi_3)$ and $Par(\pi_5)$.

(4) $\{\nearrow\}$ is a minimal signature for $Par(\pi_7)$.
(5) $\{\rightarrow\}$ is a minimal signature for $Par(\pi_8)$. □

Thus when the law $\exists o \in \Delta.\ a \leftrightarrow_o b \Leftrightarrow (\exists o \in \Delta.\ a \rightarrow_o b) \wedge (\exists o \in \Delta.\ b \rightarrow_o a)$ holds, then causality is *the only invariant that is needed*, and this fact is a *theorem* in our approach. Note that in the most general case (i.e. $Par(\pi_1)$) the explicit causality invariant is not needed. We also note that under the paradigm π_3 (and any other paradigm which contains it, i.e. π_5, π_7, π_8) we cannot distinguish between PN_2 and PN_3 of Figure 2.

4 Applications

4.1 Interleavings Inside π_8

Paradigm π_8 deserves special attention as it is the only paradigm considered in the present literature. We will show that for histories satisfying paradigm π_8, one only needs the sequential observations. A *base* of a concurrent history Δ is a pair (Δ_o, S), where $\Delta_o \subseteq \Delta$ and $S \subseteq SRI$, such that $\Delta_o^{\langle S \rangle} = \Delta$. In other words, a base provides a complete description of a history in terms of a (possibly smaller) set of observations and a suitable set of simple report invariants.

Proposition 11
If $\Delta \in Par(\pi_8)$ and $\Delta_{itl} = \{o \in \Delta \mid o \in Obs_{itl}\}$ then $(\Delta_{itl}, \{\rightarrow\})$ is a base of Δ. □

The above result means that in the case of paradigm π_8 it is possible to adequately represent a concurrent history by taking only its sequential observations. Clearly, this was the basic idea behind many models [Ma86,KP87], and can be traced back to [Sz30]. One should emphasise, however, that Proposition 11 cannot be extended to any other paradigm.

4.2. Step Sequences Inside π_3

In this and the next section we shall assume that all observations are step sequences, and that every history considered belongs to π_3. From Theorem 10 it follows that in this case $\{\nearrow, \rightarrow\}$ is a minimal signature. We shall provide an axiomatisation for this kind of signature and then define an invariant semantics of inhibitor nets. Below Obs_{step} denotes the set of all step sequences, and Σ_o denotes the set of all event occurrences.

A *pre-ordered* set is a pair (X,R) such that X is a non-empty set and $R \subseteq X \times X$ is an irreflexive ($\neg aRa$) and weakly transitive ($aRb \wedge bRc \Rightarrow a=c \vee aRc$) relation (see [Fr86]).

Note that for any Δ, the causality $\rightarrow_\Delta$ is always a poset, while the weak causality $\nearrow_\Delta$ is always a pre-ordered set. We will show that if $\Delta \subseteq Obs_{step}$ and π_3 holds, then the pair $\{\rightarrow, \nearrow\}$ can be modelled by a certain relational system which we call a *composet*.

A *combined partial order* (or *composet*) is a relational system $co=(X,P,R)$ such that X is a set and $P,R \subseteq X \times X$ are two relations satisfying the following.
(1) $\neg aRa$
(2) $aRb \wedge bRc \Rightarrow a=c \vee aRc$
(3) $aRb \Rightarrow \neg bPa$
(4) $aPb \Rightarrow aRb$

(5) $aRb \wedge bPc \Rightarrow aPc$

(6) $aPb \wedge bRc \Rightarrow aPc$.

Intuitively, P corresponds to $\rightarrow$, R corresponds to $\nearrow$, and X is a set of step sequence observations. The conditions (1) and (2) say that R is a pre-order; (4) indicates that P is included in R; (1),(4) together with, e.g., (5) imply that P is a poset; (3) is a kind of 'consistency' rule between the two orders; and (5), (6) give a kind of combined transitivity which ties together P and R.

Corollary 12

If (X,R,P) is a composet then (X,P) is a partially ordered set and (X,R) is a pre-ordered set. □

Proposition 13

If $\Delta \subseteq Obs_{step}$ then $(dom(\Delta), \rightarrow_\Delta, \nearrow_\Delta)$ is a composet. □

The above proposition is not true if $\Delta\text{-}Obs_{step} \neq \emptyset$ since (5) and (6) may not hold. A relational system $rs=(X,P,R)$ with $P,R \subseteq X \times X$ is called a π_3^{step}*-history descriptor* if $X \subseteq \Sigma_o$, $R = \nearrow_{\Delta(rs)}$ and $P = \rightarrow_{\Delta(rs)}$, where

$$\Delta(rs) = \{o \in Obs_{step} \mid dom(o) = X \wedge (\forall a,b \in X.\ aRb \Rightarrow a \rightarrow_o b \vee a \leftrightarrow_o b) \wedge (\forall a,b \in X.\ aPb \Rightarrow a \rightarrow_o b)\}.$$

Theorem 14 (axiomatisation of $\{\nearrow, \rightarrow\}$)

Let X be a finite set. A relational system $rs=(X,R,P)$ is a π_3^{step}-history descriptor *if and only if* $X \subseteq \Sigma_o$ and rs is a composet. □

The assumption $\Delta \subseteq Obs_{step}$ is essential. Without it the result does not hold. The above theorem provides an axiomatisation of signatures for histories involving finite step sequences and conforming to the paradigm π_3 (and all paradigms below π_3 in the hierarchy of Figure 5). It says that every finite composet of event occurrences may be interpreted as a representation of a finite concurrent history of the kind described above. In other words, in this case concurrent histories can be unambiguously described by composets (in the same way as the histories in π_8 can be described by causal partial orders). For infinite histories the axiomatisation is less elegant as we have to take into account the fact that step sequences are initially finite posets. We will not discuss this issue in detail, but basically one needs to provide an analysis similar to that for infinite causal partial orders (see [BD85]).

There is certain similarity between our definition of the composet and the axioms for strong and weak precedence relation presented in [La86]. However, the way these two concepts are derived, the motivations, and the reasons for their introduction are quite different. Hence this similarity is either accidental or, as we would suggest, the composet is a natural generalisation of the concept of the partial order, and it may be useful for various, perhaps unrelated, applications.

4.3. Composet Semantics of Inhibitor Nets

In this section we outline a method of constructing the set of composets of a concurrent system represented by a 1-safe Petri nets with inhibitor arcs [Pe81]. Note that 1-safeness means that each place may hold at most one token. An inhibitor arc between place p and transition (event) t means that t can only be enabled if p is not marked. In the diagrams inhibitor arcs are identified by small circles. A technique similar to that described below might be used for other kinds of inhibitor nets,

as well as for various priority models and nets (see [JL88]), however this would usually require the introduction of some new formal concepts.

The standard approach in which the partial order semantics of ordinary 1-safe Petri nets is derived employs *occurrence nets* [Re85]. An occurrence net can be regarded as a representation of a causality relation on event occurrences (or a single abstract history of the net). It is an unmarked acyclic net whose each place has at most one input and one output transition. Occurrence nets are obtained by unfolding marked nets and resolving the conflicts via the firing rules, as shown in Figure 6(a,b). Each occurrence net induces a poset on event occurrences derived in the following way: First an auxiliary relation $\rightarrow_{aux}$ is derived by transforming each three-node path $event_1 \rightarrow place \rightarrow event_2$ in the graph of the occurrence net into a pair $event_1 \rightarrow_{aux} event_2$. Then a poset is obtained by taking the transitive closure of $\rightarrow_{aux}$. For the occurrence net of Figure 6(b), the relation $\rightarrow_{aux}$ is shown in Figure 6(c), and the resulting poset is shown in Figure 6(d).

The way in which we construct composets for an inhibitor net will closely follow the above procedure. Let N_I be the inhibitor net shown in Figure 7(a). We first define an *occurrence net* of an inhibitor net by generalising in a straightforward way the standard definition of an occurrence net of an ordinary Petri net. The only new element is the handling of the inhibitor arcs. Since in the occurrence net places represent tokens, it is not possible to join c with place 2 using an inhibitor arc.

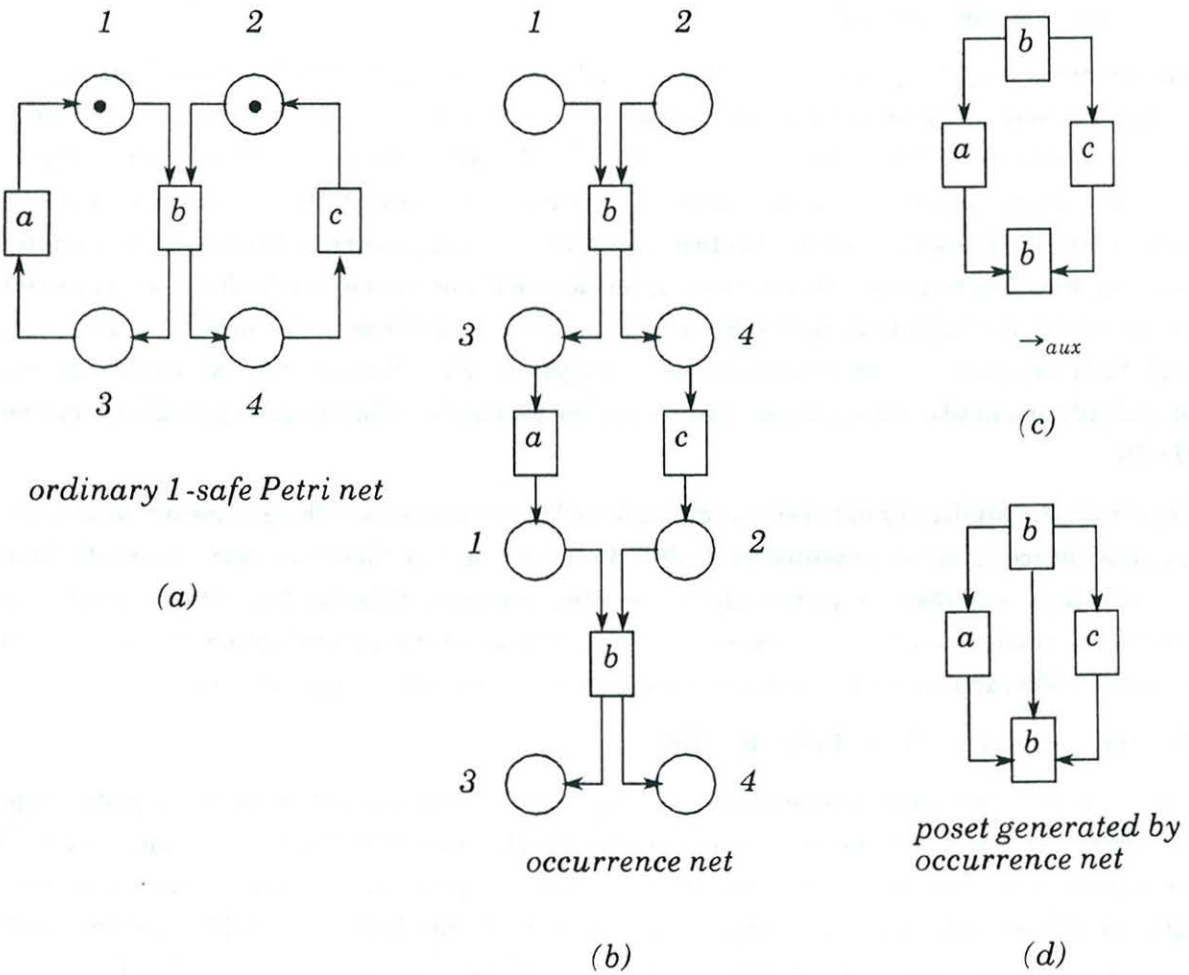

Figure 6

However, we can join c with the *complement place* [Re85] of 2, i.e. place 5, using an activator arc (with a black dot at one end). Intuitively, this means that c can be executed only when 5 is marked. We also note that there is no restriction on the number of activator arcs which can be adjacent to a single place. A possible occurrence net for the inhibitor net N_I is shown in Figure 7(b). The next step is to transform the structural relationships embedded in the graph of the occurrence net into two auxiliary relations, $\rightarrow_{aux}$ and $\nearrow_{aux}$, from which the composet can be derived. There are three structural relationships which we need to consider, as shown in Figure 8(a). For the occurrence net of Figure 7(b) the two auxiliary relations are shown in Figure 8(b). The final step has to take into account the various transitivities which hold for a composet. More precisely, if $\rightarrow_{aux}$ and $\nearrow_{aux}$ have been defined for an occurrence net ON with Σ being the set of event occurrences, then the composet induced by ON is defined as $co(ON)=(\Sigma,\rightarrow,\nearrow)$, where $(\Sigma,\rightarrow,\nearrow)$ is a minimal (w.r.t. set inclusion for both $\rightarrow$ and $\nearrow$) composet such that $\rightarrow_{aux}\subseteq\rightarrow$ and $\nearrow_{aux}\subseteq\nearrow$. It can be shown that $co(ON)$ is well-defined (i.e. it always exists and is uniquely defined). The algorithm for deriving $co(ON)$ is a straightforward generalisation of an algorithm which yields the transitive closure of the auxiliary relation in the construction of the poset for an ordinary occurrence net. For the occurrence net of Figure 7(b), the resulting $\rightarrow$ and $\nearrow$ are shown in Figure 8(c).

Final Comments

Our main goal was to show that in order to cope properly with general concurrent behaviours one should not be restricted only to poset based structures. We also tried to show that causality is only one of many possible invariants. The other invariants can be derived in a natural way when we use the bottom-top approach starting from the concept of observation as the primary notion. Although in this paper we defined observations as a certain kind of poset, concepts such as invariant, signa-

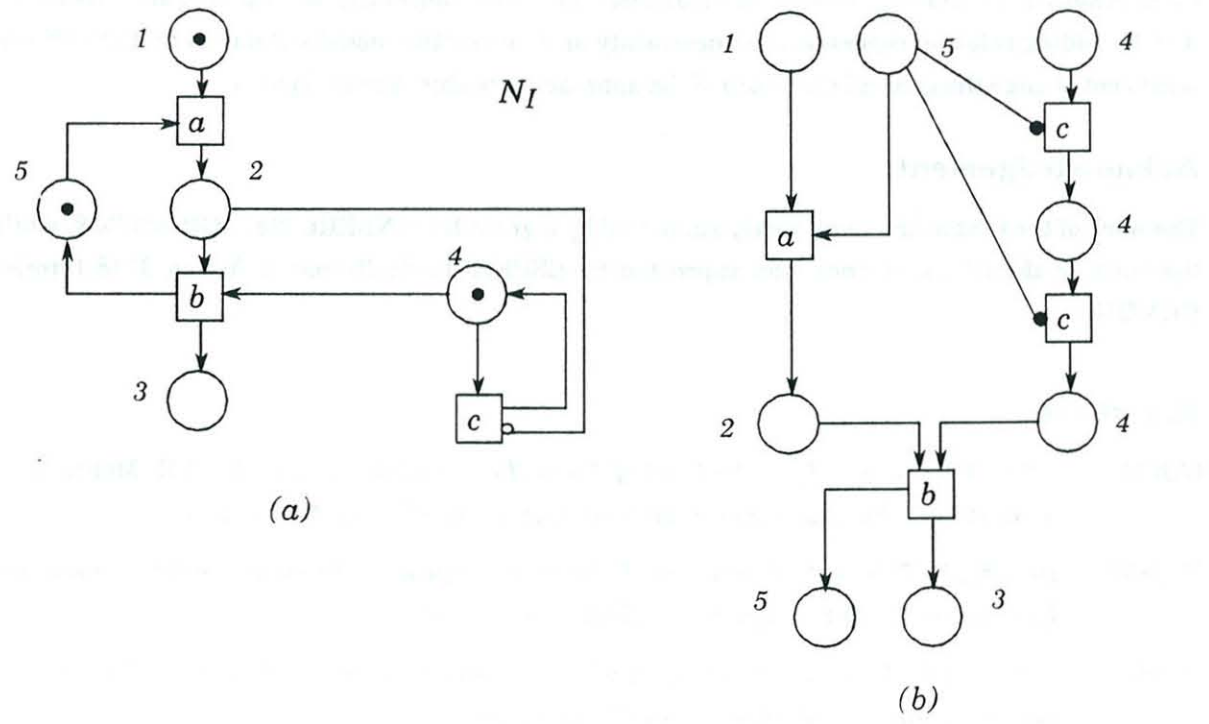

Figure 7

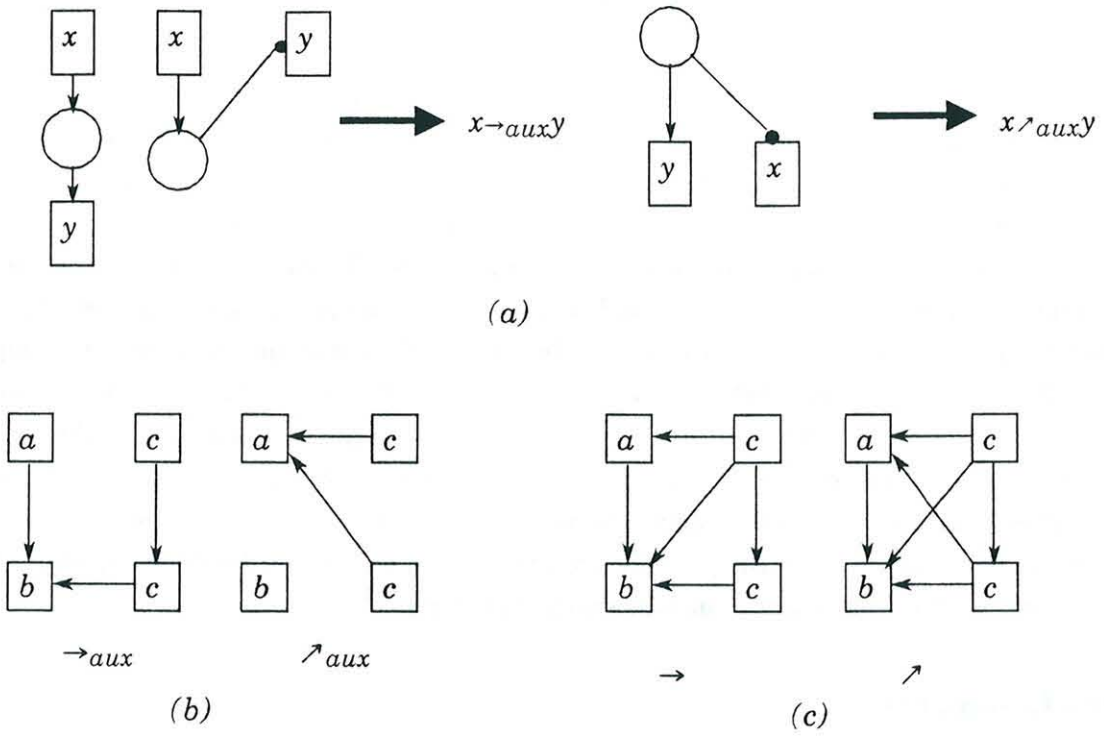

Figure 8

ture, S-closure, history, etc., are not associated with any specific definition of an observation. This paper presents a simplified version of a more general approach. In [JK90a] a general concept of 'report system' is defined, and all the concepts from Section 3 can be rendered in terms of 'reports' - generalising the notion of an observation. Consequently, the results presented here are just special cases of more general results obtained in [JK90a]. The extension of the definition of an observation (e.g. by adding relation representing uncertainty or by using the model similar to that of [AK85]) would not change the general structure of the approach introduced in this paper.

Acknowledgement

The work of the first author was partly supported by a grant from NSERC No. OGP 0036539, while the work of the second author was supported by ESPRIT Basic Research Action 3148 (project DEMON).

References

[AK85] Allen J.F., Kentz H.A., *A Model of Naive Temporal Reasoning*, In: J.R. Mobbs, R.C. Moore (Eds.), Formal Theories of the Commonsense World, Ablex 1985.

[BD85] Best E., Devillers R., *Concurrent Behaviour: Sequences, Processes and Programming Languages*, GMD-Studien Nr. 99, GMD, Bonn, 1985.

[BD87] Best E., Devillers R., *Sequential and Concurrent Behaviour in Petri Net Theory*, Theoretical Computer Science, 55 (1987), pp. 87-136.

[BK91] Best E., Koutny M., *Petri Net Semantics of Priority Systems*, to appear in Theoretical Computer Science.

[Fi70] Fishburn P.C., *Intransitive Indifference with Unequal Indifference Intervals*, J. Math. Psych. 7, 1970, pp. 144-19.

[Fi85] Fishburn P.C., *Interval Orders and Interval Graphs*, J. Wiley, 1985.

[Fr86] Fräise R., *Theory of Relations*, North Holland 1986.

[Ho85] Hoare C.A.R., *Communicating Sequential Processes*, Prentice-Hall, 1985.

[Ja87] Janicki R., *A Formal Semantics for Concurrent Systems with a Priority Relation*, Acta Informatica 24, 1987, pp.33-55.

[JK90] Janicki R., Koutny M., *Observing Concurrent Histories*, in: Real-Time Systems, Theory and Applications, H.M.S. Zedan (Ed.), Elsevier Science Publishers B.V. (North-Holland), 1990, pp 133-142.

[JK90a] Janicki R., Koutny M., *A Bottom-Top Approach to Concurrency Theory Part I: Observations, Invariants and Paradigms*, Technical Report No. 90-04, Dept. of Comp. Sci. and Syst., McMaster University, 1990.

[JL88] Janicki R., Lauer P.E., *On the Semantics of Priority Systems*, 17th Annual International Conference on Parallel Processing, Vol. 2, pp. 150-156, 1988, Pen. State Press.

[KP87] Katz S., Peled D., *Interleaving Set Temporal Logic*, 6th ACM Symposium on Principles of Distributed Computing, Vancouver 1984, pp. 178-190.

[La85] Lamport L., *What It Means for a Concurrent Program to Satisfy a Specification: Why No One Has Specified Priority*, 12th ACM Symposium on Principles of Programming Languages, New Orleans, Louisiana, 1985, pp. 78-83.

[La86] Lamport L., *On Interprocess Communication, Part I: Basic formalism, Part II: Algorithms*, Distributed Computing 1(1986), pp. 77-101.

[LH82] Lengauer C., Hehner E.C.R., *A Methodology for Programming with Concurrency: An Informal Presentation*, Science of Computer Programming 2 (1982), pp. 1-18.

[Ma86] Mazurkiewicz A., *Trace Theory*, Lecture Notes in Computer Science 225, Springer 1986, pp. 297-324.

[Mi80] Milner R., *A Calculus of Communicating Systems*, Lecture Notes in Computer Science, vol. 92, Springer 1980.

[Mo76] Monk J.D., *Mathematical Logic*, Springer 1976.

[Pe81] Peterson J.L., *Petri Net Theory and the Modeling of Systems*, Prentice Hall, 1981.

[Pr86] Pratt V., *Modelling Concurrency with Partial Orders*, Int. Journal of Parallel Programming 15, 1 (1986), pp. 33-71.

[Re85] Reisig W., *Petri Nets*, Springer 1985.

[Sz30] Szpilrajn-Marczewski E., *Sur l'extension de l'ordre partial*, Fundamenta Mathematicae 16 (1930), pp. 386-389.

[Wn14] Wiener N., *A Contribution to the Theory of Relative Position*, Proc. Camb. Philos. Soc. 17 (1914), pp. 441-449.

[Wi82] Winskel G., *Event Structure Semantics for CCS and Related Language*, Lecture Notes in Computer Science 140, Springer 1982, pp. 561-567.

DISCUSSION

Rapporteur: J. Harley

Colin Bron asked, in relation to the three types of semantics discussed in the paper, whether Dr. Koutny had defined composition rules for his system. Dr. Koutny replied that he could have defined composition operations, based on the composition of partial ordes, but that he had not looked at this problem yet.

Brian Randell asked what insights should we gain regarding notations such as CSP and CCS – in terms of advantages or disadvantages. Dr. Koutny replied that the problems were essentially the same as those with Petri Nets, and remarked that this kind of problem is orthogonal to that of comparing Petri Nets to CSP and so on. He also emphasised that one important technique to be gained from his work was that of applying reduction techniques.

Alexander Jakovlev asked whether Dr. Koutny had made any comparisons to other paradigms. Dr. Koutny said that this was another topic for future work.

Chris Holt asked what the problems might be if one introduced "real" time into the system. Dr. Koutny replied that one would need a non-linear, partial order time domain, and that this was not included in the present model as it would be too complex.