Theoretical Computer Science 64 (1989) 281-304
North-Holland

CONCURRENT SYSTEMS AND INEVITABILITY

Antoni MAZURKIEWICZ*, Edward OCHMAŃSKI
and Wojciech PENCZEK

Institute of Computer Science of PAS, PO Box 22, PL-00-901 Warszawa PKiN, Poland

Communicated by J. de Bakker
Received January 1987
Revised October 1987

Abstract. Concurrent systems viewed as partially ordered sets of states are considered. A property of system states is called inevitable, if the system will eventually reach a state with this property. This notion is discussed within the partial order framework.

Introduction

Sequences of system states are a commonly used tool in analysing the behaviour of discrete systems. Such a sequence is said to be *admissible*, if each but the first state in the sequence results from its predecessor as a result of a system action; and is said to be *nonextendable*, if it starts with an initial state of the system and is either infinite, or its last element is a state to which no system action is applicable. In the theory of sequential systems admissible and nonextendable sequences of system states (we will call them execution sequences) represent processes generated by such systems; and the set of all processes generated by a system defines fully its behaviour.

Execution sequences are also used to analyse concurrent systems. In this case the situation becomes more complex, since admissible and infinite sequences may not describe the full execution of a system. Therefore, in such a case, a third requirement is added to the previous two: to describe a full process generated by a concurrent system a state sequence must not only be admissible and nonextendable, but also *fair*. Roughly speaking, a sequence is fair, if it does not ignore any component of the system. For a precise formulation of fairness assumptions the reader is referred to papers of Lehmann et al. [6], Manna and Pnueli [7], Owicki and Lamport [15], and of Pnueli [18]; here, only a sketch of the main ideas concerning this issue will be given.

The fairness assumption is needed in order to prove that something will happen during the run of a system. Intuitively, we call a property *inevitable*, if sooner or

* Partly sponsored by LPC during the first author's stay at Leiden University, The Netherlands.

0304-3975/89/$3.50

later the system will be in a state meeting this property. To formally prove the inevitability of a property it seems to be natural to consider execution sequences and to check whether each of them contains a state with the property in question; if so, the property is inevitable, otherwise it is not. But, clearly, each of the considered execution sequences must represent the full behaviour of the system, i.e. they must be fair. If the fairness assumption were too weak, e.g. requiring only the nonextendability of sequences, some intuitively inevitable properties could not be proved; if it were too strong, then it would allow the inevitability of some doubtful properties to be proved. The weakest fairness assumption, known to the authors, is that of "justice" [6].

The idea of defining processes of concurrent systems by fair execution sequences, as discussed above, will be referred to as the "sequential approach". In this paper an attempt is made to look at inevitability from the point of view of the so-called "partial order semantics", where processes are represented by partially ordered sets of states rather than sequences of states. In such an approach an assumption which guarantees partially ordered sets to represent the full behaviour of the system is also needed. In this "partial order approach" the difference between nondeterminism and concurrency is made explicit, in contrast to the sequential approach in which this difference disappears. Due to this fact, in a partial order approach one could distinguish two types of fairness assumptions: one, which guarantees that no system component is ignored in a process run ("concurrency fairness"), and another concerning decisions made during the system action ("conflict resolution fairness"). Since the first type of fairness seems to be the weakest possible, we assume every process to have this type of fairness; the second is ignored here, since we do not want to impose any restrictions on the way the conflicts are resolved.

The present paper is certainly not the first dealing with fairness in the partial order framework; in [2] problems of transition fairness and in [12] also of marking fairness in Petri nets have been considered. In the present approach, however, no particular formalism is assumed, nor any kind of events is distinguished with respect to fairness; it is the inevitability of arbitary properties (sets of states) which is discussed here.

The paper is organized as follows. First, a definition of concurrent systems is given; next, the notion of processes is introduced; such processes are intended to describe the full system behaviour, i.e., they have "concurrency fairness". It gives a basis for defining inevitability; this definition relies on the notion of process observations. Next, the notion of observable system is defined and some properties of such systems are considered. Later, a definition of inevitability is given and finally this notion is considered in so-called diamond-discrete concurrent systems. It is not the intention of this work to develop a temporal logic based on the notion of inevitability defined here; the aim is only to offer a possible model for such a logic. Therefore, all tools used here are set-theoretical.

Several papers have influenced the present one: those mentioned above, the paper of Nielsen et al. [13] and of Winkowski [21]. The reader will easily find a relationship between them and the present paper. It should be stressed that the need for the partial order approach for describing the behaviour of nonsequential systems was pointed out very early on by Petri [17].

This paper is based on [10]; some results in Sections 6 and 7 have already appeared in [14] and [16].

1. Basic notions

Standard mathematical notation is used here. The set of all non-negative integers will be denoted by ω. A relation will always be understood to be a binary relation. For any relation $\rightarrow$ in a set S by $\rightarrow^*$ we shall denote its transitive and reflexive closure (in S). Sometimes we shall write xRy for $(x, y) \in R$. Each reflexive, transitive and antisymmetric relation in S is an *ordering relation* in S. A pair $(S, \leq)$, where S is a set and $\leq$ is an ordering relation in S, is a *partially ordered set*, or a *poset* for short. A poset $(S, \leq)$ is *discrete*, if there exists a nontransitive and irreflexive relation $\rightarrow$ in S such that $\leq = \rightarrow^*$ (the successor relation for $\leq$: it is easy to show that the successor relation for a poset is unique, if it exists). In discrete poset, if $x \rightarrow y$, then y is a *successor* of x and x is a *predecessor* of y. Throughout the whole paper, posets in examples are discrete. A poset $(S', \leq')$ is a *subposet* of a poset $(S, \leq)$, if

$$S' \subseteq S, \qquad \leq' = \leq \cap (S' \times S').$$

Thus, any subset of S uniquely determines a subposet of $(S, \leq)$. As usual, we shall write S instead of $(S, \leq)$ if it does not lead to confusion. If S and S' are posets, S' is a subposet of S, we shall write $S' \subseteq S$. Clearly, the family of all subposets of a poset is (partially) ordered by $\subseteq$. Let $\sigma \in S$; then $\uparrow\sigma$ and $\downarrow\sigma$ are defined by

$$\uparrow\sigma = \{\tau \in S \mid \sigma \leq \tau\},$$
$$\downarrow\sigma = \{\tau \in S \mid \tau \leq \sigma\}.$$

Operations $\uparrow$, and $\downarrow$ are extended to subposets of S by defining for each $P \subseteq S$:

$$\uparrow P = \bigcup \{\uparrow\sigma \mid \sigma \in P\},$$
$$\downarrow P = \bigcup \{\downarrow\sigma \mid \sigma \in P\}.$$

From the transitivity of $\leq$ it follows that $\uparrow P = \uparrow(\uparrow P)$ and $\downarrow P = \downarrow(\downarrow P)$, and from its reflexivity $P \subseteq \uparrow P$ and $P \subseteq \downarrow P$. Hence, in particular, $\uparrow S = \downarrow S = S$ and $\uparrow\emptyset = \downarrow\emptyset = \emptyset$. To avoid a number of parentheses we assume $\uparrow$ and $\downarrow$ to bind stronger than other set-theoretical operations, e.g. $\uparrow P' \cap P''$ denotes $(\uparrow P') \cap P''$, not $\uparrow(P' \cap P'')$. For each $\sigma \in S$ the set $\downarrow\sigma$ ($\uparrow\sigma$) is called a *backward* (*forward*) *cone* in S; the union $\downarrow\sigma \cup \uparrow\sigma$ is a *cone* in S determined by σ. If C is a cone in S determined by σ, then $(S - C) \cup \{\sigma\}$ is an *anticone* determined by σ. Let P', P'' be subposets of S; P' is said to be *cofinal* (*coinitial*) with P'', if $\downarrow P' = \downarrow P''$ ($\uparrow P' = \uparrow P''$, respectively). Two elements σ', σ'' of a poset $(S, \leq)$ are *comparable*, if either $\sigma' \leq \sigma''$ or $\sigma'' \leq \sigma'$, and *incomparable* otherwise.

A poset S is *linearly ordered*, if each two of its elements are comparable. Each linearly ordered subposet of a poset S is called a *chain* in S. A maximal chain in a poset S is called a *line* in S. A subposet P of S is said to be *backward closed*, if $\downarrow P = P$. A poset is *directed*, if together with any two elements σ', σ'' it contains an element σ such that $\sigma' \leq \sigma$ and $\sigma'' \leq \sigma$, and *branching* otherwise. A subset P of S is *bounded by* σ in S, if $\sigma' \leq \sigma$ for every $\sigma' \in P$; P is *bounded*, if it is bounded by an element of S and *unbounded* otherwise. A poset S is *backward finite*, if for any $\sigma \in S$ the set $\downarrow\sigma$ is finite.

The following fact is known as the Zorn–Kuratowski Lemma [22].

Proposition 1.1. *If every chain in a poset $(S, \leq)$ is bounded, then for each ζ_0 in S there is a maximal element ζ of S with $\zeta_0 \leq \zeta$.*

Proof. Can be found e.g. in [5]. □

A family of sets is *monotonically additive*, if it contains the union of any of its subfamilies linearly ordered by inclusion. We shall use the following earlier version of the Zorn–Kuratowski Lemma [4].

Proposition 1.2. *Any member of a monotonically additive family can be extended to a maximal member of this family.*

Proof. Take in the Zorn–Kuratowski Lemma a monotonically additive family of sets ordered by inclusion and observe that any subfamily is bounded by its union. □

2. Concurrent systems

A concurrent system will be viewed here as the set of all possible histories of its activity. As in [13], by history we understand here a set of events which, together with an event, contains all events prior to it. Each such history determines uniquely a (global) state of the system, and any system state can be supplied with the whole of its history; thus, we shall identify histories with system states. The set of such states is ordered by inclusion: earlier states are contained in later ones. It is clear that no state can be repeated during a system run (histories can only grow) nor can its two different executions lead to the same state (a history contains the whole past). In contrast to the sequential case, in nonsequential processes the inclusion ordering of histories is only partial (some initial parts of the same history can be noncomparable by inclusion).

Simple examples of states are: left-closed subsets of an elementary event structure [13], processes in a Petri net [19] from the initial slice to another slice, prefixes of string-vectors [20], and those of traces [9, 11].

Each of the concepts mentioned above leads to a suitable mathematical model of concurrent systems. Since we do not intend to specify any of these models but we are going to keep our framework as general as possible, we simply view such systems as partially ordered sets of states. It turns out that even such a general approach allows us to formulate and discuss our questions.

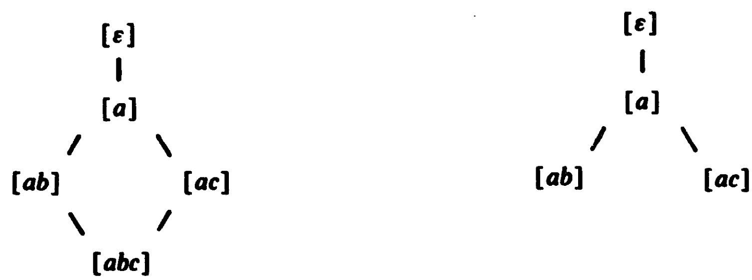
Definition 2.1. By a *concurrent system* (or simply, a *system*), abbreviated as cs, we shall understand here any poset $(S, \leq)$, where S is called the set of *states* of the system, and $\leq$ the *dominating* relation of the system. We shall say that a system S is *discrete* if S is a discrete poset. If $\sigma' \leq \sigma''$, we say that σ'' *dominates* σ', or that σ' is *dominated* by σ''. This notion is extended to sets of states: we say that a subset P of S is dominated by a subset Q of S, if for each state σ in P there is a state in Q dominating σ. Two states σ', σ'' of S are said to be *consistent*, if there is a state σ in S such that $\sigma' \leq \sigma$, $\sigma'' \leq \sigma$ (there is a state dominating both of them), and *inconsistent* otherwise; S is *sequential*, if any two of its consistent states are comparable; it is *conflict-free*, if any two of its states are consistent (i.e., S is directed).

If a state σ' is dominated by another state σ'', it means that there is a run of the system, containing both these states, in which σ'' comes up later than σ'. Thus, in our intended interpretation, two states are comparable, if and only if they appear in the same execution of the system and the history of one of them is an initial part of the history of the other. If σ' is incomparable with σ'', two cases are possible: either both of them are dominated by a common state σ (then they are consistent), or such a common state does not exist (then they are inconsistent). In the first case σ' and σ'' are two states of the same run of the system, but they result in some concurrent actions of the system; they identify different pieces of the same history, exhibited later by σ. In the second case σ' and σ'' are states identifying pieces of two different histories, resulting in a choice (a conflict resolution) made earlier during the system action.

Example 2.1. Consider a concurrent system in which first an action a is executed, and next two actions, b and c, are performed concurrently; after doing it, the system terminates. The following states of the system are possible: $[\varepsilon]$—the initial state, $[a]$—after performing action a, $[ab]$—after performing action a and b, $[ac]$—after a and c, and finally $[abc]$—after performing all actions a, b, c. This system can be defined as the cs $(S, \rightarrow^*)$, where

$$S = \{[\varepsilon], [a], [ab], [ac], [abc]\}$$

and where $\rightarrow$ is the relation defined by the diagram in Fig. 1(a); (in this and the following figures the convention is to represent $x \rightarrow y$ by drawing a line from x to y and placing x above y). States $[ab]$ and $[ac]$ are incomparable here, but consistent (both of them are dominated by $[abc]$). The whole poset represents a single run of the system; an observer of the system can see the sequence $[\varepsilon]$, $[a]$, $[ab]$, $[abc]$;



(a) (b)

Fig. 1.

another observer, however, can see the same run as the sequence [ε], [a], [ac], [abc] and both observations are equally valid.

The reader acquainted with traces [11] can easily see that states here are constructed from prefixes of all traces generated by the system. The system is conflict-free, since its only two incomparable states [ab] and [ac] are consistent.

Example 2.2. Consider now the system in which, first, action *a* is performed yielding a state [a], and next either action *b* or action *c* can be executed, giving [ab] or [ac], respectively, and no further action is performed. This system can be viewed as cs $(S, \rightarrow^*)$, where

$$S = \{[\varepsilon], [a], [ab], [ac]\}$$

and where $\rightarrow$ is defined by the diagram in Fig. 1(b). In this case two different runs are possible in the system. States [ab] and [ac] are incomparable and inconsistent—there is no common state dominating them. Any observer will now notice the same sequence, namely [ε], [a], [ab] or [ε], [a], [ac], depending on which continuation of state [a] will be chosen by the system. This system is sequential, since any two of its consistent states are comparable.

3. Processes and their properties

Definition 3.1. Let $S = (S, \leqslant)$ be a cs: any maximal directed subposet of *S* is called a *process* in *S*; the family of all processes in *S* is called the *behaviour* of *S*.

In view of the explanations given above, the interpretation of this notion is clear: each process contains only states which can appear in a single execution of the system (because of the directedness all states of a process are consistent, hence they define pieces of the same history). By maximality, processes in a system correspond to its full executions (i.e. they display "concurrency fairness").

Example 3.1. In Fig. 1(a), $\{[\varepsilon],[a],[ac]\}$ is directed but not maximal, hence it is not a process; in system terms, the concurrent component performing b is ignored. In Fig. 1(b) the same set is a process.

Example 3.2. The following example is intended to show how the above notions are related to typical concurrent systems. As a sample system let us take the well-known system with two active components, SENDER and RECEIVER, communicating with each other by an unbounded BUFFER working according to the first-in-first-out principle.

SENDER sends successive messages to the BUFFER until it (possibly) decides to terminate its activity: then it sends a closing message and halts. RECEIVER takes messages from the BUFFER until it (possibly) gets the closing message; then it halts. SENDER and RECEIVER act independently of each other; if, however, the BUFFER is empty, RECEIVER must wait for a message from SENDER.

Thus, three actions are being performed by the system: sending messages, receiving them, and terminating (the SENDER). A state of the system is then completely defined by knowing how many actions of each type have already been performed. Such information can be represented by a triple (i, j, k) of nonnegative integers indicating how many times the sending action (the integer i), the receiving action (the integer j), and the terminating action (the integer k) have already been executed by the system (however, in the general case, such information may not be sufficient; then tuples of individual histories rather than tuples of numbers have to represent states). Clearly, $k \leq 1$ and $j \leq i$ (the number of receiving actions cannot exceed that of sending actions). Therefore, our specification turns into a system $\mathrm{SBR} = (S, \rightarrow^*)$, with

$$S = \{(i, j, k) \mid j \leq i, k \leq 1\},$$

$$\rightarrow = R_1 \cup R_2 \cup R_3.$$

where

$$R_1 = \{((i, j, 0), (i+1, j, 0)) \mid j \leq i\} \qquad \text{(sending)},$$

$$R_2 = \{((i, j, k), (i, j+1, k)) \mid j < i, k \leq 1\} \qquad \text{(receiving)},$$

$$R_3 = \{((i, j, 0), (i, j, 1)) \mid j \leq i\} \qquad \text{(terminating)}.$$

By its very definition SBR is discrete. The actions in R_1 correspond to sending messages, those in R_2 to receiving messages, and those in R_3 to the termination of sending messages. A graphical form of the system described above is presented in Fig. 2. In this figure lines directed downwards-right correspond to R_1, i.e. to sending actions, those directed downwards-left to R_2, i.e. to receiving actions, and those directed straight down to R_3, i.e. to the terminating (closing) action; "ijk" stands for "(i, j, k)".

```
000
 |  \
001    100
     /  |  \
110    101    200
 |  x      /  |  \
111    210    201    300
     /  |  x      /  |  \
220    211    310    301    400
 |  x      /  |  x      /  |  \
221    320    311    410    401    500
     /  |  x      /  |  x      /  |  \
...    ...    ...    ...    ...    ...    ...
```

Fig. 2. (/ = receive, \ = send, | = terminate).

In this system states $(1,1,0)$, $(2,0,0)$ are incomparable, but consistent: states $(2,1,1)$, $(3,1,0)$ are inconsistent; subsets

$$Q=\{(i,j,k)\,|\,0\leq i, j=k=0\},$$

$$P_\omega=\{(i,j,k)\,|\,0\leq j\leq i,\ k=0\},$$

$$P_n=\{(i,j,k)\,|\,0\leq j\leq i,\ (i\leq n,\ k=0 \text{ or } i=n,\ k=1)\},$$

$(n\in\omega)$, are directed subsets. Q is not a run of the system. In Q only SENDER is active and its sending activity is endless; since RECEIVER is ignored, there is no concurrency fairness. P_ω and P_n are runs of the system. In P_ω both components act forever. In P_n SENDER halts after n steps and RECEIVER consumes all messages having been sent by SENDER. Later we shall prove formally that P_ω and P_n are (all) processes of SBR.

Let $(S,\leq)$ be a system fixed from now on; all directed subsets and processes discussed below are assumed to be in S, and in all introduced notions S is assumed to be known.

Let us mention some general facts about processes. First of all, from the definition of a process it follows that for any two directed subsets P' and P'' with $P'\subseteq P''$, if P' is a process, then so is P'' (in fact, $P'=P''$); and thus, if P'' is not a process, then neither is P'. Secondly, it follows from Proposition 1.2 that each directed subposet can be extended to a process because the family of all directed subposets is monotonically additive. This implies that the behaviour of a system is never empty. It also implies that each state in S is in a process in S. Thirdly, it should be clear that if S is conflict-free, then the only process generated by S is S itself. If S is a sequential system, then every directeu ... is a chain, and hence every process is a line. Fourthly, each process is backward closed; indeed, let P be a process and let $\sigma'\leq\sigma\in P$: then $P\cup\{\sigma'\}$ contains P and is directed; as was shown above, $P=P\cup\{\sigma'\}$; it means that $\sigma'\in P$ which proves backward closedness of P. Finally, we quote a proposition useful for proving directed subposets to be processes. We say that a state is *consistent* with a set of states, if it is consistent with each state of this set. Clearly, each state of a directed subset P is consistent with P.

Proposition 3.1. *A directed subset of S containing all states consistent with itself is a process.*

Proof. Let P be a directed subset containing all states consistent with P and let Q be another directed subset such that $P \subseteq Q$. Let $\sigma \in Q$; then σ is consistent with Q, hence also with P; by assumption, $\sigma \in P$. Thus, $Q \subseteq P$, hence $P = Q$. It means that P is a process. □

In other words, a directed subset P of S is a process, if for any state σ not in P there is a state in P inconsistent with σ.

The converse of the above proposition is not true. In the following example we have a state consistent with all states of a process but not belonging to this process.

Example 3.3. Let $C = (\omega, \rightarrow^*)$ be a cs where

$$\rightarrow = \{(3n, 3n+2) \mid n \in \omega\} \cup \{(3n+1, 3n+2) \mid n \in \omega\}$$
$$\cup \{(3n, 3n+3) \mid n \in \omega\} \cup \{(3n+1, 3n+4) \mid n \in \omega\}.$$

Then $P = \{0, 3, 6, \ldots\}$ and $Q = \{1, 4, 7, \ldots\}$ are two maximal directed subposets of C, i.e. processes in C, each state of P is consistent with each state of Q, but P and Q are disjoint (Fig. 3).

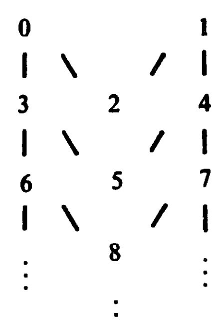


Fig. 3.

This ends our list of general properties of processes.

As an important special case we will now consider terminating processes. We will define termination by analogy with the case of sequential systems, where an execution sequence terminates if it reaches a state from which no further action can be performed.

Definition 3.2. A state is *terminal*, if it is a maximal state of S; a subposet is *terminating*, if it contains a terminal state.

It is easy to see that for each terminating subposet P of S which is directed and backward closed, there exists a terminal state σ such that $P = \downarrow\sigma$ (in fact, if $\sigma \in P$ is terminal, then, by directedness of P, σ must be the greatest state of P, i.e., $P \subseteq \downarrow\sigma$; since, moreover, P is backward closed, also $\downarrow\sigma \subseteq P$). Hence each terminating process is of the form $\downarrow\sigma$ with σ terminal. This also holds in the other direction: for every terminal state σ, $\downarrow\sigma$ is a process. Indeed, let τ be a state not in $\downarrow\sigma$; then τ must be inconsistent with σ since otherwise σ would be not terminal. Thus, $\downarrow\sigma$ contains all states consistent with it and by Proposition 3.1 it is a process. This characterizes terminating processes. Another characterization (of which the easy proof is left to the reader) is: a process is terminating if and only if it is bounded.

Example 3.4. We prove that the family of all processes generated by the SBR system defined in Example 3.2 is the family

$$B = \{P_0, P_1, \ldots, P_n, \ldots, P_\omega\},$$

where

$$P_n = \downarrow(n, n, 1) \quad \text{for } n \in \omega,$$

$$P_\omega = \{(i, j, 0) \mid j \leq i \in \omega\}.$$

These processes are presented in graphical form (using the same conventions as previously) in Fig. 4.

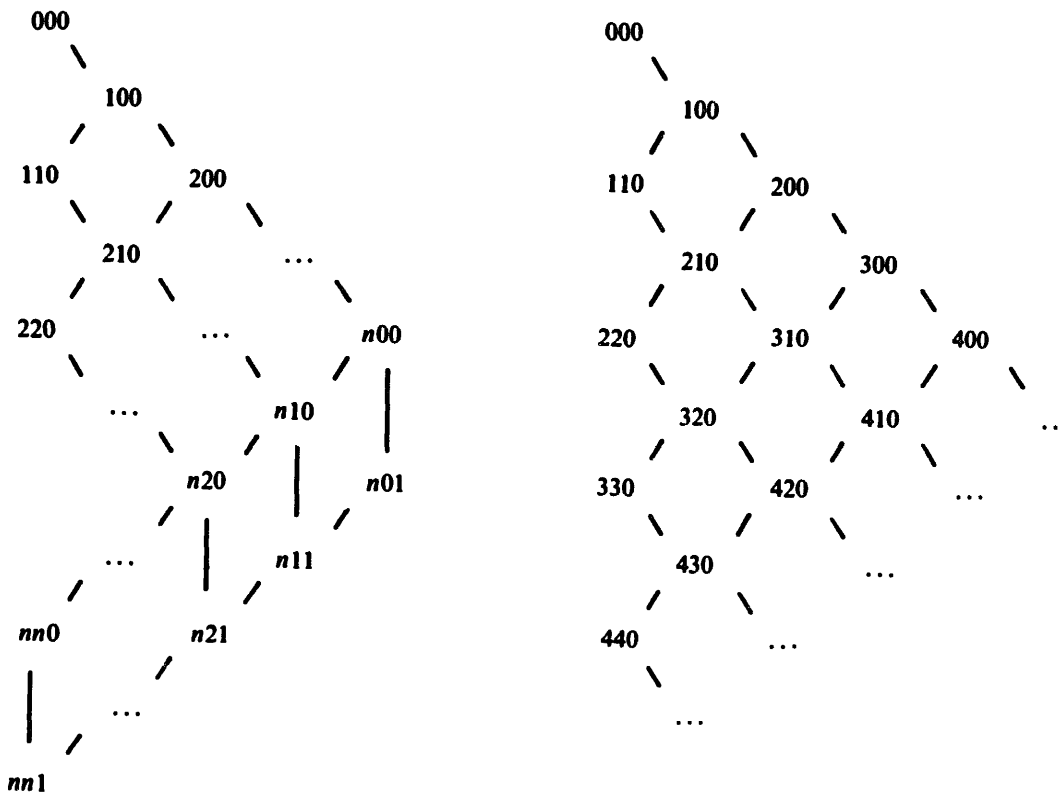
First we prove that each member of the family B is a process. Each P_n $(n \in \omega)$ is a process since $P_n = \downarrow(n, n, 1)$ and $(n, n, 1)$ is a terminal state of SBR. To prove it for P_ω take an arbitrary state not in P_ω, say (i, j, l); it is clearly inconsistent with $(i+1, j, 0)$ belonging to P_ω, hence P_ω is a proces , by Proposition 3.1. Now we prove that the family B is the family of *all* processes generated by SBR. Let P be an arbitrary directed subset in SBR. If P contains a state $(i, j, 1)$ then $P \subseteq P_i$ because $(i, j, 1)$ is inconsistent with every state not in P_i; otherwise $P \subseteq P_\omega$. This proves that each directed subset in SBR is contained in some member of B, i.e. that B is the family of all processes in SBR. Observe that the directed set Q, $Q = \{(i, 0, 0) \mid i \in \omega\}$, is not a process, since it is strictly contained in P_ω.

Thus, turning back to the original interpretation of SBR, we can say that in each process RECEIVER acts as long as possible.

Example 3.5. The behaviour of the system defined in Example 3.3 consists of the following processes:

$$\downarrow(3n+2) \quad (n \in \omega, \text{terminating processes}),$$

$$\{0, 3, 6, \ldots\}, \quad \text{and} \quad \{1, 4, 7, \ldots\} \quad (\text{unbounded processes}).$$



Process P_n Process P_ω

Fig. 4.

We end this section by a characterization of processes in sequential systems. It shows that processes in sequential systems are execution sequences satisfying the usual requirements, as discussed in the introduction.

Proposition 3.2. *Let S be a sequential system, and $P \subseteq S$. P is a process if and only if P is a backward closed chain which is either terminating or unbounded.*

Proof. We have already seen that every process is backward closed, every directed set in a sequential system is a chain, and a process is terminating if and only if it is bounded. Now assume that P is a backward closed chain. First consider the case that P is unbounded. Let Q be a directed set with $P \subseteq Q$. Take $\sigma \in Q$. Since P is unbounded, there exists $\tau \in P$ such that $\tau \leq \sigma$ does not hold. Thus, since Q is a chain, $\sigma \leq \tau$. Hence $\sigma \in P$, because P is backward closed. This shows that $Q = P$. Hence P is a process. Next consider the case that P is terminating. We have shown before that P is of the form $\downarrow\sigma$ with σ terminal. Thus, as also shown before, since σ is terminal, P is a process. $\square$

This explains why in the analysis of sequential systems there is no need for a special fairness assumption: nonextendability is sufficient.

Note that Proposition 3.2 does not hold for nonsequential systems. In example 3.2, the set Q is a backward closed unbounded chain but we have proved that Q is not a process.

4. Lines and observations

The set of states of a process consists of all states that can be noticed during a run of a system. The presence of a state in a process indicates only the possibility (but not the necessity) of its occurrence in a single run of the system. Any two incomparable states of a process are equally likely to be noticed, but none of them will necessarily be noticed (actually, there is no observation of a system run containing both of them together). To infer some invariant properties of a process ("safety" properties) it is quite sufficient to consider all possible process states; to prove some eventualities ("liveness" properties) we must rely on the necessity of the appearance of some states in lines, i.e. we must be sure that each observer of the process, sooner or later, will notice a state with a required property. To prove that a property will eventually hold in a process, we must show that in any (reasonably defined) process observation there is a state meeting the property. Therefore, to speak about eventual properties of processes we must refer to observations. A definition of inevitable properties of processes will be given in Section 6; in this and the next section we define the notion of an observation and discuss some of its properties.

In the sequential approach lines are of the main interest and considered as the only representatives of processes generated by concurrent systems. However, as has already been mentioned above, in such an approach the distinction between lines in the same process and those of different processes disappears. In the partial order approach lines of the system behaviour are also considered; but since they are related to each process separately, and only the set of all processes defines the system behaviour, this difference is preserved and gives rise to another setup. In this approach, the behaviour of a system is not a set of lines, as in the sequential approach, but a set of processes containing sets of lines (each process contains at least one line).

An important notion connected with lines is that of domination. In general, lines in a process, being its linear subsets, do not contain all its states. However, remembering that the history of a state σ contains (as prefixes) the histories of all states dominated by σ, the amount of information about a process supplied by its line depends upon how big a set of process states is dominated by this line. Because of maximality, lines are not comparable by inclusion; instead, we can compare their dominating properties and say that line V "supplies more information" than U, if V dominates more process states than U, i.e. if $\downarrow U \subseteq \downarrow V$, or equivalently, if $U \subseteq \downarrow V$.

Clearly, a line dominating all states of P, i.e. cofinal with P, is a line with "maximal amount of information". The above intuitions lead to the following definition.

Definition 4.1. A line V of a process P is said to be an *observation* of P, if $P = \downarrow V$ (P is cofinal with V).

Observations give full information about processes; since each state identifies its history, states of an observation identify the total history of the observed process.

Example 4.1. Consider the SBR system defined in Example 3.2. The subset Q defined in this example is a line of P_ω; it is not an observation of P_ω, since it dominates only itself, leaving all states $(i, j, 0)$ with $j > 0$ undominated. On the other hand, the chain V: $V = \{(i, j, 0) \mid j \leq i \leq j+1\}$, is an example of an observation of P_ω, since

(i) it is a line and

(ii) each state $(n, m, 0)$ in P_ω is dominated by a state in V, namely by $(n, n, 0)$.

Definition 4.2. A process is *observable*, if there exists an observation of it, and *unobservable* otherwise; a system is *observable*, if every process in it is observable and *unobservable* otherwise.

Proposition 4.1. *Each chain in a process P can be extended to a line in P.*

Proof. By Proposition 1.2, since the family of all chains in a process is monotonically additive. □

Theorem 4.1. *Each bounded chain in an observable process P can be extended to an observation of P.*

Proof. If P is empty, the assertion trivially holds; otherwise, let V be an observation of P and let L be a chain bounded by σ (i.e. $L \subseteq \downarrow\sigma$) for some $\sigma \in P$. Set $U = V \cap \uparrow\sigma$ and $T = L \cup U$; clearly, T is a chain in P; we shall show that T dominates P. Let ζ be a state of P; by directedness of P, there is $\tau \in P$ dominating ζ and σ. Since V is an observation of P, there is $\tau' \in V$ dominating τ. Since $\sigma \leq \tau \leq \tau'$, $\tau' \in U$; since $\zeta \leq \tau \leq \tau'$, τ' dominates ζ. This means that P is cofinal with U, hence P is cofinal with $L \cup U$. By extending $L \cup U$ to a maximal chain (Proposition 4.1) we get the required observation, since extensions preserve cofinality. □

By the above theorem, there is no way, in general, to decide whether a line with a given initial segment is an observation or not. This indicates the nonfinitary character of the notion of observation.

Corollary. *For each state in an observable process P there is an observation of P containing this state.*

At the end of this section we give an example of an unobservable system.

Example 4.2 (Marek [8]). Let RN be the system with all finite subsets of real numbers as states and inclusion as dominating relation. RN is obviously directed, since the union of any two finite sets contains both of them and is finite. Thus, RN is conflict-free and is its own (only) process. Any chain in RN can dominate at most a countable family of states, since it consists of a countable number of finite sets only; thus, it cannot dominate the whole set RN, since RN is not countable. Therefore, RN is not cofinal with any of its lines, i.e. there is no observation of RN.

Observe that RN is discrete (!) and backward finite.

Properties of unobservable systems are interesting on their own and will be discussed separately elsewhere. Here, we concentrate only on observable systems. Some important classes of such systems will be discussed in the next section.

5. Observable systems

Definition 5.1. A system is *cone-countable*, if each cone in S is countable.

Note that the set of all states in a cone-countable system may be not countable; any countable system, however, is clearly cone-countable.

Proposition 5.1. *Each process in a cone-countable system is countable.*

Proof. Let S be a cone-countable system and let P be a process in S. If P is empty, the proof is completed. Let $\sigma \in P$ and $P' = \uparrow\sigma$, $P'' = \downarrow P'$. Since S is cone-countable, $\uparrow\sigma$, $\downarrow\sigma$ are countable. Thus, P' is countable, and so is P'', since the union of a countable family of countable sets is also countable. It remains to show that $P \subseteq P''$. Let $\sigma_0 \in P$; by directedness of P there is $\sigma' \in P$ such that $\sigma_0 \leq \sigma'$, $\sigma \leq \sigma'$; hence $\sigma' \in P'$ and, by definition, $\sigma_0 \in P''$. It proves $P \subseteq P''$. □

Theorem 5.1. *Cone-countable systems are observable.*

Proof. Let P be a process in a cone-countable system, Q be the set of states of P. If P is empty, then P is (the only) observation of P. Assume P to be not empty; by Proposition 5.1, Q is countable; let then

$$(\sigma_0, \sigma_1, \ldots, \sigma_n, \ldots)$$

be an enumeration of Q (i.e. $Q=\{\sigma_n \mid n\in\omega\}$). Define a sequence $V=(\tau_0,\tau_1,\ldots,\tau_n,\ldots)$ inductively as follows:

$\tau_0=\sigma_0$:

τ_{n+1} is a state in P dominating τ_n and σ_{n+1}.

Clearly, V is linearly ordered since $\tau_n \leq \tau_{n+1}$ for each n; moreover, V dominates P. Indeed, let $\sigma' \in P$: then $\sigma'=\sigma_j$ for some j, hence $\sigma' \leq \tau_j$. Thus, each state of P is dominated by a state of V. Extending V to a line (Proposition 4.1) we get an observation of P. □

Note that an observable process can also admit lines which are not observations, as follows from Example 4.1. In the systems discussed in the rest of this section all lines of processes are observations, and hence they are observable (because by Proposition 4.1 each process contains at least one line).

Definition 5.2. A system S is *terminating* iff all its processes are terminating.

Proposition 5.2. *Each line of a terminating process is an observation.*

Proof. Assume P to be terminating; hence there is a terminal state σ such that $P=\downarrow\sigma$. Let V be a line in P; since V is maximal, $\sigma\in V$: hence $\sigma\subseteq\downarrow V$. This proves $P\subseteq\downarrow V$. □

Corollary. *Terminating systems are observable.*

Definition 5.3. A system S is *strongly synchronized*, if for each state $\sigma\in S$ there is at most a finite number of states incomparable and consistent with σ.

Example 5.1. All systems discussed above (except SBR and RN) are strongly synchronized.

Theorem 5.2. *Each line in a process of a strongly synchronized system is an observation of that process.*

Proof. Assume S to be strongly synchronized and let P be a process in S, V be a line of P, and $\sigma\in P$. We have to prove that $\sigma\in\downarrow V$. If $\sigma\in V$, the proof is finished. Assume $\sigma\notin V$. Since V is a maximal linearly ordered subposet of P, there are some states in V not comparable with σ. Since S is strongly synchronized, there are a finite number of such states. Let σ_0 be the greatest of them; it cannot be the greatest element of V, since then V would be strictly contained in the chain $V\cup\{\sigma'\}$, where σ' is a state dominating both σ_0 and σ; hence, there is a state σ'' in V dominating σ_0 and different from σ_0; by definition of σ_0, σ'' dominates σ. Thus, $\sigma\in\downarrow V$. □

Corollary. *Strongly synchronized systems are observable.*

Proposition 5.3. *Sequential systems are observable.*

Proof. There are no consistent and incomparable states in sequential processes; hence, they are strongly synchronized. □

Proposition 5.4. *The (only) observation of a process generated by a sequential system is the process itself.*

Proof. Let P be a process generated by a sequential system. Then P is a line in P dominating P, hence it is an observation of P. □

The above fact shows why in the sequential approach to concurrent systems observations are used as representatives of processes: in sequential systems processes can be identified with their observations.

6. Inevitability

In this section the notion of inevitability will be introduced and discussed. The difference between reachability and inevitability is that the first refers to a possibility, while the second refers to the necessity of reaching a state with a given property during the system run. Clearly, any inevitable property is reachable, but not the other way around.

We define inevitability in observable systems making use of observations of its processes. Informally, a property is inevitable, if any sequential observer of the system run will notice, sooner or later, a state with this property, provided his observation, though sequential, contains (directly or indirectly) information about every state occurring in the system run (i.e., it is an observation as defined formally in Definition 4.1).

The intuitions given above lead to the following formal definition. Let S be a fixed, nonempty, observable concurrent system and let a *property* mean a subset of S. Given two sets, we say that one of them *intersects* the second, if their intersection is nonempty.

Definition 6.1. A property Q is *inevitable* in a process P of S, if any observation of P intersects Q and *avoidable* in P otherwise. A property is inevitable in S, if it is inevitable in each process of S and avoidable otherwise.

The following theorem gives a necessary condition for inevitability in observable processes; informally, it states that if Q is inevitable, then either now it is Q, or it was Q, or it will be Q.

Theorem 6.1. *If Q is inevitable in a process P, then $P \subseteq \downarrow Q \cup \uparrow Q$.*

Proof. Assume Q to be inevitable in P. Let $\sigma \in P$. By the corollary to Theorem 4.1 there is an observation V of P containing σ. Since Q is inevitable in P, there is $\xi \in Q \cap V$. Since V is a line, either $\sigma \leq \xi$, or $\xi \leq \sigma$. Since $\xi \in Q$, either $\sigma \in \downarrow Q$, or $\sigma \in \uparrow Q$, i.e. $\sigma \in \downarrow Q \cup \uparrow Q$. □

The converse of this theorem does not hold, as follows from the example below.

Example 6.1. Let $S1 = (\omega, \rightarrow^*)$ with

$$\begin{aligned} \rightarrow = & \{(2n, 2n+1) \mid n \in \omega\} \cup \{(2n, 2n+2) \mid n \in \omega\} \\ & \cup \{(2n+1, 2n+3) \mid n \in \omega\}, \end{aligned}$$

(Fig. 5) and let $Q = \{1, 4\}$. Then $S1$ is its only process, $S1 \subseteq \downarrow Q \cup \uparrow Q$, but Q is avoidable in $S1$: $\{0, 2, 3, 5, 7, \ldots\}$ is an observation of $S1$ not intersecting Q.

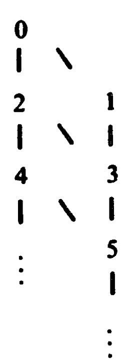


Fig. 5.

Proposition 6.1. *Any anticone is intersected by any line.*

Proof. If a chain V does not intersect the anticone determined by σ, then $V \cup \{\sigma\}$ is a chain again; hence V is not maximal, i.e. it is not a line. □

Theorem 6.2. *Any anticone is inevitable in S.*

Proof. Trivial. □

Theorem 6.3. *For each σ in a process P the forward cone $\uparrow\sigma$ is inevitable in P. Informally: any process state will be eventually in the past of this process.*

Proof. Let $\sigma \in P$. If V is an observation of P, then V dominates σ; hence V intersects $\uparrow\sigma$. □

Note that anticones are inevitable in the whole system S (Theorem 6.2) whereas forward cones not necessarily. In order to discuss more closely the notion of inevitability, consider the following three conditions:

C1(P, Q): any line in P dominating P intersects Q;

C2(P, Q): any line in P undominated by $P \cap Q$ intersects Q;

C3(P, Q): any line in P dominating $P \cap Q$ intersects Q.

By definition, Q is inevitable in P iff C1(P, Q). Now, we intend to find relationships of inevitability with conditions C2(P, Q) and C3(P, Q). The following theorem holds.

Theorem 6.4. C1(P, Q) *implies* C2(P, Q).

Proof. Assume C1(P, Q) and let V be a line in P undominated by $P \cap Q$. Hence there is $\sigma \in V$ such that $\sigma \notin \downarrow(P \cap Q)$. The chain $V \cap \downarrow\sigma$ can be (Theorem 4.1) extended to an observation U of P. Since Q is inevitable in P, $U \cap Q \neq \emptyset$. Since $\sigma \notin \downarrow(P \cap Q)$, $(U \cap \uparrow\sigma) \cap Q = \emptyset$; thus, it must be $(U \cap \downarrow\sigma) \cap Q \neq \emptyset$; but $(U \cap \downarrow\sigma) = (V \cap \downarrow\sigma)$, hence $V \cap Q \neq \emptyset$. □

The above theorem cannot be converted. This is shown by the following example.

Example 6.2. Let $S2 = (\omega, \rightarrow^*)$ with

$$\rightarrow = \{(2n, 2n+1) \mid n \in \omega\} \cup \{(2n, 2n+2) \mid n \in \omega\}$$
$$\cup \{(2n+1, 2n+3) \mid n \in \omega\} \cup \{(2n+1, 2n+4) \mid n \in \omega\},$$

(Fig. 6) and let $Q = \{1, 3, 5, \ldots\}$, $P = S2$; since $S2$ is conflict free, P is the only process of $S2$. Then C2(P, Q) holds for Q, since $\downarrow Q = P$, hence every line in P is dominated by Q. But Q is avoidable in P, since $\{0, 2, 4, \ldots\}$ is an observation of P not intersecting Q. □

It is clear that C3(P, Q) implies C1(P, Q). The following example shows that this implication cannot be converted.

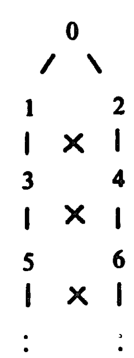


Fig. 6.

Example 6.3. Let $S3 = (\omega, \rightarrow^*)$, where

$$\begin{aligned}\rightarrow = &\{(0,1),(0,2)\}\\ &\cup\{(3n+1, 3n+4) \mid n \in \omega\} \cup \{(3n+2, 3n+3) \mid n \in \omega\}\\ &\cup\{(3n+1, 3n+5) \mid n \in \omega\} \cup \{(3n+2, 3n+4) \mid n \in \omega\}.\end{aligned}$$

$S3$ is conflict-free, hence its only process P is $S3$ itself. The set $Q = \{2, 5, 8, 11, 14, \ldots\}$ is inevitable in P. But Q does not satisfy C3(P, Q), since the line $\{0, 1, 4, 7, 10, \ldots\}$ dominates Q and does not intersect Q (Fig. 7).

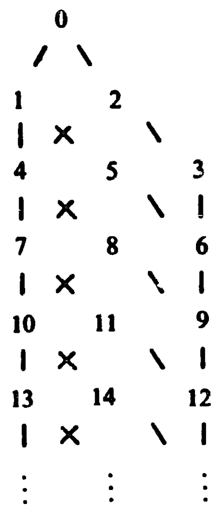


Fig. 7.

The last results of this section are summarized in the diagram below indicating whether implication holds or does not:

$$\mathrm{C3}(P, Q) \rightarrow \mathrm{C1}(P, Q) \rightarrow \mathrm{C2}(P, Q),$$

$$\mathrm{C3}(P, Q) \not\leftarrow \mathrm{C1}(P, Q) \not\leftarrow \mathrm{C2}(P, Q).$$

These facts indicate that the notion of inevitability adopted here can be included among the other possible candidates for its definition.

7. Observable diamond-discrete systems

In this section we will show that for a class of observable systems, inevitability can be expressed by condition C3(P, Q) from the last section. This class is the class of diamond-discrete systems defined below.

Definition 7.1. A poset P has the *diamond property*, if any two different and consistent states of S with a common predecessor also have a common successor. A system is *diamond-discrete* iff it is discrete and each process in its behaviour has the diamond property.

The diamond property assumption is quite natural in discrete systems; e.g. processes in systems generated by Condition/Event Petri Nets have the diamond property; the unobservable system RN from Example 4.2 also has this property.

Let $(S, \to^*)$ be an observable, diamond-discrete system fixed from now on: all processes discussed below are assumed to be in S. Let P be a process and $Q \subseteq P$. Let Suc(Q) be the set of all successors of states in Q:

$$\mathrm{Suc}(Q) = \{\sigma \in P \mid \text{there is } \tau \text{ in } Q \text{ such that } \tau \to \sigma\}.$$

Proposition 7.1. *Let P be an unbounded process, $\sigma_0, \sigma', \sigma'' \in P$, $n, m \in \omega$. If $\sigma_0 \to^n \sigma'$ and $\sigma_0 \to^m \sigma''$, then there is τ in P such that $\sigma' \to^m \tau$ and $\sigma'' \to^n \tau$.*

Proof. It is not difficult to prove, using unboundedness, that for every $\sigma_1 \in P$ there exists $\sigma_2 \in P$ such that $\sigma_1 \to \sigma_2$. This implies that in P any two (not necessarily different) states with a common predecessor also have a common successor. From this the statement is easy to show, using double induction w.r.t. m and n. □

Now we formulate two auxiliary lemmas holding for diamond-discrete processes.

Lemma 7.1. *Let P be an unbounded process and $V \subseteq P$. If V dominates* Suc(V), *then V dominates $\uparrow V$.*

Proof. Assume V dominates Suc(V). Let $\xi \in \uparrow V$. Thus $\sigma \to^n \xi$ for some $\sigma \in V$, $n \in \omega$. We prove this lemma by induction on n. If $n = 0$ then $\xi = \sigma \in V \subseteq \downarrow V$. Let $n > 0$ and assume, as induction hypothesis, that $\sigma \to^{n-1} \zeta$ implies $\zeta \in \downarrow V$. Since $\sigma \to^n \xi$ there is $\zeta \in \uparrow V$ such that $\sigma \to^{n-1} \zeta \to \xi$. By induction hypothesis $\zeta \in \downarrow V$, thus $\zeta \leqslant \tau$ for some $\tau \in V$. By Proposition 7.1 there is $\sigma' \in P$ such that $\xi \leqslant \sigma'$ and $\tau \to \sigma'$. Thus $\xi \leqslant \sigma' \in \mathrm{Suc}(V) \subseteq \downarrow V$, so $\xi \in \downarrow V$. □

Lemma 7.2. *Let P be an unbounded process and $\emptyset \neq V \subseteq P$. Then V dominates P iff V dominates* Suc(V).

Proof. (⇒): obvious.

(⇐): Let $\xi \in P$ and $\tau \in V$. Since P is a process there is $\sigma \in P$ such that $\xi \leqslant \sigma$ and $\tau \leqslant \sigma$. Since $\tau \leqslant \sigma$, we have $\sigma \in \downarrow V$ by Lemma 7.1; since $\xi \leqslant \sigma$ it holds that $\xi \in \downarrow V$. □

The above lemma does not hold for processes that are not diamond-discrete as can be seen from the following example.

Example 7.1. Consider system $S3$ depicted in Fig. 7. $S3$ is not diamond-discrete, since e.g. $2 \to 4$ and $2 \to 3$, and 3 is consistent with 4, but 4 and 3 have no common direct successor. The set $V = \{0, 1, 4, 7, \ldots\}$ dominates $\mathrm{Suc}(V) = \{1, 2, 4, 5, 7, 8, \ldots\}$, but V does not dominate $S3$.

Now we are ready to prove the main result of this section, characterizing inevitability in observable diamond-discrete processes.

Definition 7.2. A property Q is said to be *locally inevitable in* a process P, if any line in P dominating $P \cap Q$ intersects Q (i.e. if the condition C3(P, Q) from Section 6 holds).

Theorem 7.1. *Let P be a process in an observable diamond-discrete system S and let $Q \subseteq S$. Then Q is inevitable in P iff Q is locally inevitable in P.*

Proof. Clearly, we may assume that $Q \subseteq P$.

($\Leftarrow$): Obvious; any observation dominates Q.

($\Rightarrow$): If P is bounded, then this trivially holds, since then any line in P is an observation of P (by Proposition 5.2 and the fact that a process is bounded if and only if it is terminating), hence any line dominating $P \cap Q$ is an observation of P. Consider now the case that P is an unbounded process. Assume Q is inevitable in P and let $V \subseteq P$ be a line dominating $P \cap Q$. We have to show that V intersects Q. If V is an observation, then V intersects Q by the definition of inevitability. Assume now that V is not an observation, i.e. that V does not dominate P. By Lemma 7.2 there is $\xi \in \mathrm{Suc}(V)$ such that $\xi \notin \downarrow V$. It follows from Theorem 5.2 that the chain $(V \cap \downarrow \xi) \cup \{\xi\}$ can be extended to an observation V'. Since Q is inevitable, V' intersects Q. Let $\sigma \in V' \cap Q$. If $\xi \leq \sigma$ and $\sigma \in Q$, then, since Q is dominated by V, $\xi \leq \sigma \in \downarrow V$, thus $\xi \in \downarrow V$. This contradicts $\xi \notin \downarrow V$, hence $\sigma \leq \xi$ and $\sigma \neq \xi$. But then $\sigma \in V$, hence V intersects Q. □

Our result can be generalized for systems.

Corollary. *Let S be an observable diamond-discrete system and $Q \subseteq S$ be a property. Then Q is inevitable in S iff Q is locally inevitable in each process of S.*

It should be stressed that local inevitability can be proved without using the observation concept; hence, to check the local inevitability of a property it may be sufficient to take into account only a part of the whole system.

8. Final remarks

A property has been defined as inevitable in an observable concurrent system, if any observation of the system has a state with this property. Since observations are linearly ordered sets of states, this definition is quite close to that used in the sequential approach. However, there are some differences which mean that notions of inevitability in the sequential and the present approaches are not equivalent. Namely, roughly speaking, there are "more" observations than "just" lines, and consequently (cf. [7]), than "fair" lines. This implies that some properties inevitable in the sequential approach may not be in the present, "partial order approach". The following example illustrates this difference.

Example 8.1. Consider the system $A=(\omega,\rightarrow^*)$ with $\rightarrow$ defined by the diagram in Fig. 8(a). A is sequential (any two consistent states are comparable) and it can be viewed as a representation of the following, sequential and nondeterministic program:

$$(\mathbf{true}\rightarrow x:=x+2)^* \,\square\, \mathbf{true}\rightarrow (x:=x+1; \mathbf{stop}),$$

where $\square$ denotes the choice operator.

According to the justice assumption, this program must always terminate, since the terminating instruction $\mathbf{true}\rightarrow(x:=x+1; \mathbf{stop})$ is continuously enabled. In our wording it would mean that the justice assumption implies inevitability of the property $Q=\{1,3,5,\ldots\}$. But in our approach this property is *not* inevitable, since the sequence $(0,2,4,\ldots)$ is a process in C (and at the same time its own observation) not intersecting Q. (We would not assume the "conflict resolution fairness" in our setup.)

(a) (b)

Fig. 8.

Consider now the system $B=(\omega,\rightarrow^*)$ with $\rightarrow$ defined in Fig. 8(b) which can be thought as representing the following nonsequential but conflict-free program:

$$(\mathbf{true}\rightarrow x:=x+2)^* \,\|\, \mathbf{true}\rightarrow (x:=x+1; \mathbf{stop})$$

($\|$ denotes the parallel composition operator). In this case both—sequential and partial—approaches lead to the same conclusion, namely that the termination of the right-hand component of the system is inevitable. The argument in the sequential approach is the same as before: the terminating instruction is continuously enabled, hence it must eventually be executed; in our approach, the situation is changed radically in comparison with system A, since there is now only one process (the system itself), and the sequence $(0, 2, 4, \ldots)$ is not an observation anymore (it does not dominate e.g. any state from $Q = \{1, 3, 5, \ldots\}$). Our argument for the inevitability of Q is Theorem 6.3: the future of any process state is inevitable in this process; 1 is a state of the considered process, and $\uparrow 1 = Q$ is its future.

From the above example it also follows that a property can be live in a system (as defined by Alpern and Schneider in [1]), but not inevitable in our sense; termination of the system C is a live property in C while it is not inevitable in our approach.

The point, as already discussed in the introduction, is that in any sequential approach every fairness assumption must impose some restrictions on possible conflict resolutions, because in such approaches conflicts and concurrency have the same effect in execution sequences. In the partial order approach, which admittedly leads to more complex notions, the separation of these two phenomena is possible. Thus, one can reformulate the justice assumption:

> "A set of instructions cannot be continuously enabled and never executed"

to the following form adequate to the partial order approach:

> "A set of instructions cannot be continuously and *concurrently* enabled and never executed"

or, in a more appropriate wording:

> "A set of instructions cannot be permanently concurrent to the remaining instructions and never completed".

Clearly, these formulations are vague and imprecise as long as the terms they contain are not precisely defined; defining them, however, exceeds the scope of this paper. Anyway, they give an idea about the relationship of these two approaches.

Acknowledgment

The authors wish to thank Professor G. Rozenberg and Professor P.S. Thiagarajan for their help and encouragement. The help and criticism of Dr. J. Engelfriet, who was the first reader of a preliminary version of this paper and whose remarks improved it considerably, are gratefully acknowledged. The authors would like to thank anonymous referees for their valuable comments and suggestions.

References

[1] B. Alpern and F.B. Schneider, Defining liveness, *Inform. Process. Lett.* **21** (1985) 181-185.
[2] E. Best, Fairness and conspiracies, *Inform. Process. Lett.* **18** (1984) 215-220.
[3] U. Goltz and W. Reisig, The non-sequential behaviour of Petri nets, *Inform. Contr.* **57**(2/3) (1983) 125-147.
[4] K. Kuratowski, Une méthode d'élimination des nombres transfinis des raisonnements mathématiques, *Fund. Math.* **3** (1922) 76-108.
[5] K. Kuratowski and A. Mostowski, *Set Theory* (North-Holland, Amsterdam, 1967).
[6] D. Lehmann, A. Pnueli and A. Stavi, Impartiality, justice and fairness: The ethics of concurrent termation, in: *Proc. Internat. Conf. Automata, Languages and Programming*, Lecture Notes in Computer Science **115** (Springer, Berlin, 1981) 264-277.
[7] Z. Manna and A. Pnueli, Verification of concurrent programs: Temporal proof principles, in: D. Kozen, ed., *Logic of Programs*, Lecture Notes in Computer Science **154** (Springer, Berlin, 1982) 200-252.
[8] W. Marek, Private communication (1986).
[9] A. Mazurkiewicz, Traces, histories, graphs: Instances of a process monoid, in: M. Chytil, ed., *Mathematical Foundations of Computer Science 1984*, Lecture Notes in Computer Science **176** (Springer, Berlin, 1984) 115-133.
[10] A. Mazurkiewicz, Complete processes and inevitability, Report 86-06, University of Leiden (1986).
[11] A. Mazurkiewicz, Trace theory, in: W. Brauer, W. Reisig and G. Rozenberg, eds., *Advances in Petri Nets 1986, Part II*, Lecture Notes in Computer Science **255** (Springer, Berlin, 1987) 279-324.
[12] A. Merceron, Fair processes, in: G. Rozenberg ed., *Advances in Petri Nets 1987*, Lecture Notes in Computer Science **266** (Springer, Berlin, 1987) 181-195.
[13] M. Nielsen, G. Plotkin and G. Winskel, Petri nets, event structures and domains, Part 1, *Theoret. Comput. Sci.* **13** (1981) 85-108.
[14] E. Ochmański, Inevitability in concurrent systems, *Inform. Process. Lett.* **25** (1987) 221-225.
[15] S. Owicki and L. Lamport, Proving liveness properties of concurrent programs, *ACM Trans. Programming Languages and Systems* 4(3) (1982) 455-495.
[16] W. Penczek, Inevitability in diamond-discrete systems, ICS PAS Report 598 (1986).
[17] C.A. Petri, Non-sequential processes, GMD-ISF Report 77-05 (1977).
[18] A. Pnueli, The temporal semantics of concurrent programs, *Theoret. Comput. Sci.* **13** (1981) 45-60.
[19] W. Reisig, *Petri Nets—An Introduction*, EATCS Monographs on Theoretical Computer Science (Springer, Berlin, 1985).
[20] M.W. Shields, Nonsequential Behaviour; Part 1, Tech. Rept. CSR-120-82 Dept. of Computer Science, University of Edinburgh (1982).
[21] J. Winkowski, Behaviours of concurrent systems, *Theoret. Comput. Sci.* **12** (1980) 39-60.
[22] M. Zorn, A remark on method in transfinite algebra, *Bull. Amer. Math. Soc.* **41** (1935) 667-670.