Theoretical Computer Science 18 (1982) 227–258
North-Holland Publishing Company

A UNIFIED APPROACH FOR STUDYING THE PROPERTIES OF TRANSITION SYSTEMS

Joseph SIFAKIS
Laboratoire IMAG, 38041 Grenoble Cedex, France

Communicated by R. Milner
Received August 1979
Revised December 1980

Abstract. In this paper a systematic method for generating, comparing and proving the properties of transition systems is presented. It is assumed that any property of a system can be defined by giving a set of 'target' states and a type of reachability. Ten different types of reachability are proposed; by appropriately choosing the set of target states, a family of ten potentially different properties is generated. The main conclusion is that the reachability types and therefore the system properties, can be characterized by simple relations involving the set of the possible initial states and fixed points of certain continuous predicate transformers depending on the set of target states. As a consequence, in order to prove a given property it is sufficient to compute iteratively greatest or least fixed points of continuous predicate transformers.

Some examples are presented which show how the results can be applied to prove the properties of concurrent systems represented by non-deterministic models.

1. Introduction

The elaboration of a general verification theory of systems requires on the one hand the existence of a sufficiently general model on which this theory could be developed and on the other hand a precise and operational definition of the notion of correctness. While for sequential (serial) systems, sequential programs in particular, the verification theory seems to be well established, for parallel systems the two aforementioned requirements subsist; in fact there exists neither a generally accepted notion of correctness for parallel systems nor a general verification method. The aim of this paper is to propose a general framework for tackling the problem of system verification.

Verifying a system means proving the validity of a set of statements about its behaviour. We believe, without limiting the generality of the approach, that a distinction made in the case of sequential programs can be maintained by considering that these statements are of two kinds:

- statements which are valid at every state of the system and characterize all its possible states; these statements correspond to *invariant properties*;
- statements which express the fact that an event may or should occur one or several times if the system is initialized correctly. Such statements correspond to general

0304-3975/82/0000–0000/$02.75

properties of its control such as, termination, liveness, persistence, presence of deadlock, presence of livelock, etc; the term *non-invariant properties* is used to denote them.

Therefore, verifying a system amounts to proving that both a set of invariant properties (weak correctness) and a set of non-invariant properties hold; which properties are considered depends on what it is specified as the 'good' functioning of the system under study. A similar distinction between invariant and non-invariant properties can also be found in [27] ('invariant' and 'temporal' properties) and in [21] ('safety' and 'liveness' properties).

If such a definition of the notion of correctness is adopted, then a system verification theory must offer general methods for proving invariant and non-invariant properties. While invariance is a rather well understood concept, this is not the case for the other properties. The main reason for this is their (surprisingly) great variety: in the literature one can find, three different types of termination, more than five types of liveness, two types of livelock .... Furthermore, there is a lot of confusion about the terminology: different terms often denote the same property and conversely the same term is sometimes used to characterize different situations. For example, the terms 'live' in [13], 'live-4' in [26], 'live-5' in [22], 'immortal' in [16] denote the same property which is quite different from this one in [21]. Also, the statement "a system S has a deadlock" can be given the three different meanings:

(1) "there exists a set of states at which the system S is blocked forever",

(2) "the system S can reach a state at which it is blocked forever",

(3) "the system S will certainly reach a state at which it is blocked forever".

In this paper a systematic method for generating, comparing and proving control properties is presented. The results are developed on a highly abstract relational model, called *transition systems* [16]. This model employs very few primitive notions namely those of state and transition (action) which are at the base of every discrete model. The advantages of working with such a primitive model, especially when studying the properties of concurrent systems, are now widely recognized [16–19, 30].

According to the proposed method, in order to verify a given property it may be necessary to compute the greatest or least fixed point of a predicate transformer which characterizes certain aspects of the functioning of the system under study. From this point of view our approach has been inspired from the work by Van Lamsweerde and Sintzoff [20]; some of their results on deadlock and starvation detection can be derived from those of Section 3.3.2.2. Also, there exists some similarity between our approach and the method by Flon and Suzuki [10] as far as the use of predicate transformers and the representation of parallel systems by non-deterministic models are concerned.

This paper is organized in four parts. In Section 2 we present two fundamental concepts: invariants and trajectories. These are predicates that can be iteratively computed as fixed points of continuous predicate transformers. The continuity of

these predicate transformers is shown to be in relation with the boundedness of the non determinism as in [6, 29, 15, 11].

In Section 3 a systematic method for studying properties is presented; it will be argued that every system property can be defined by giving a target predicate and a type of reachability. Ten different types of reachability are then defined; by appropriately choosing a target predicate a family of ten potentially different properties is generated. The main result of this part is that the reachability types, and consequently the properties of systems, can be characterized by simple relations involving the set of the initial states of a system and invariants or trajectories which depend on the target predicate.

In Section 5 illustration is given of some possible applications of the theory developed in the preceding parts.

2. The concepts of invariant and trajectory

2.1. Preliminary results

2.1.1. Transition systems

A *transition system* [16] is defined by a triplet $S = (Q, T, R)$ where,
- Q is a countable set of *states*,
- $T = \{t_1, t_2, \ldots, t_m\}$ is a set of *transitions*,
- $R = \{R_1, R_2, \ldots, R_m\}$ is a set of binary relations on Q in bijection with the transitions.

Transition systems are a primitive model employing very few notions which are at the base of every discrete model. Transitions correspond to actions which can transform, by their execution, the state of the system; the relation R_i describes precisely the effect of the transition t_i. A transition t_i is executable from a state q only if there exists a state q' such that $(q, q') \in R_i$; we then say that t_i is *enabled* by q and that q' is the state *reached from* q after the execution of t_i. The notation $q \to q'$ is an abbreviation for $\exists i \in \{1, \ldots, m\}\ (q, q') \in R_i$. With a transition system can be associated a digraph the vertices of which are the elements of Q and the edges are labelled by T: there is an edge from q to q' labelled by t_i iff $(q, q') \in R_i$. For a path of such a graph given by a sequence of states $q_{i_0} q_{i_1} \ldots q_{i_w}$, with $(q_{i_{j-1}}, q_{i_j}) \in R_{i_j}$ for $j = 1, 2, \ldots, w$, the corresponding sequence of transitions $\sigma = t_{i_1} t_{i_2} \ldots t_{i_w}$ is a sequence of actions executable from the state q_{i_0}. We note $q_{i_0} - \sigma \to q_{i_w}$. Also, $q_{i_0} \to^* q_{i_w}$ is an abbreviation for $\exists \sigma \in T^*\ q_{i_0} - \sigma \to q_{i_w}$ i.e. $\to^*$ represents the reflexive transitive closure of $\to$.

Transition systems are a *sequential non-deterministic* model: sequential in the sense that only one transition can be executed at a time and nondeterministic in the sense that generally more than one transitions are enabled at a given state. Obviously, every sequential discrete system can be represented, at some level of abstraction, by a transition system. Furthermore, in so far as the concurrency in the functioning of a system can be represented by global non-determinism (as in

[20] and [17]), transition systems can be considered a primitive model for the representation of concurrent systems too.

2.1.2. The predicate transformers pre and post

Given a model, in which the actions are represented by binary relations on a set of states, it is generally admitted that predicate transformers provide an elegant manner for dealing with its semantic properties. The functions *pre* and *post* studied in this subsection paragraph are the basic predicate transformers used in this paper. They have already been introduced and studied by several authors, for example [33, 1, 24, 15]; however, for the sake of completeness we recall and comment here some of their useful properties.

Let Q be a countable set and $\mathscr{P}$ the set of unary predicates on Q:

$$P \in \mathscr{P} \Leftrightarrow P : Q \to \{\textbf{true}, \textbf{false}\}.$$

We represent by $\mathscr{L} = (\mathscr{P}, \vee, \wedge, \neg, \top, \bot)$ the complete lattice on $\mathscr{P}$ with respectively $\vee$, $\wedge$, $\neg$, $\top$, $\bot$ the operations of disjunction, conjunction, negation, the 'always true' predicate ($\bigvee\mathscr{P}$), the 'always false' predicate ($\bigwedge\mathscr{P}$). $\mathscr{L}$ is isomorphic to the complete lattice of subsets of Q, $\mathscr{L}' = (2^Q, \cup, \cap, \sim, Q, \emptyset)$; with every $P \in \mathscr{P}$ can be associated its characteristic set $\boldsymbol{P} = \{q \in Q \mid P(q) = \textbf{true}\}$. We use $\sqsubseteq$ and $\subseteq$ in order to represent the order relation on $\mathscr{L}$ and $\mathscr{L}'$ respectively: $(\forall P_1, P_2 \in \mathscr{P})\ (P_1 \sqsubseteq P_2 \Leftrightarrow \boldsymbol{P}_1 \subseteq \boldsymbol{P}_2)$.

Let $\mathscr{F}$ be the set of unary functions mapping $\mathscr{P}$ into $\mathscr{P}$. The elements of $\mathscr{F}$ are called *predicate transformers*. We extend on $\mathscr{F}$ the operations $\vee$, $\wedge$, $\neg$: for $F, G \in \mathscr{F}$ the expressions $F \vee G$, $F \wedge G$, $\neg F$ represent respectively the functions $(F \vee G)(P) = F(P) \vee G(P)$, $(F \wedge G)(P) = F(P) \wedge G(P)$, $\neg F(P) = \neg(F(P))$, where P is a predicate of $\mathscr{P}$. The notation $\tilde{F}$ is used to represent the *dual* of F: $\tilde{F}(P) = \neg F(\neg P)$. In the sequel, the abbreviation $P(q)$ is used in the place of $P(q) = \textbf{true}$.

Let $\mathscr{R}$ be the set of binary relations on Q. For $R \in \mathscr{R}$ we define the function $\text{pre}[R] \in \mathscr{F}$:

$$\text{pre}[R](P)(q) \Leftrightarrow \exists q' \in Q\ P(q') \textbf{ and } (q, q') \in R.$$

Properties 1

(a) $\text{pre}[R](\bot) = \bot$;

(b) $\text{pre}[\emptyset](P) = \bot$;

(c) $\text{pre}[1_Q](P) = P$, where $1_Q = \{(q, q') \in Q^2 \mid q = q'\}$;

(d) $\text{pre}[Q \times Q](P) = T$, if $P \neq \bot$;

(e) For $\{P_i\}_i$ an arbitrary sequence of predicates of $\mathscr{P}$,

$$\text{pre}[R]\left(\bigvee_i P_i\right) = \bigvee_i \text{pre}[R](P_i);$$

(f) For $\{P_i\}_i$ an arbitrary sequence of predicates of $\mathscr{P}$,

$$\text{pre}[R]\left(\bigwedge_i P_i\right) \sqsubseteq \bigwedge_i \text{pre}[R](P_i);$$

(g) $P_1 \sqsubseteq P_2 \Rightarrow \text{pre}[R](P_1) \sqsubseteq \text{pre}[R](P_2)$.

Remark. Properties 1(f) and 1(g) are corollaries of property 1(e) which establishes the distributivity of pre[R] over disjunction. Property 1(g) states that pre[R] is monotonic and property 1(f) is a consequence of 1(g).

The following proposition gives a characteristic property of the set $\mathscr{R}(P) = \{F \in \mathscr{F} \mid \exists R \in \mathscr{R}\ F = \text{pre}[R]\}$.

Proposition 1. *A function $F \in \mathscr{F}$ is an element of $\mathscr{R}(P)$ if and only if $F(\bot) = \bot$ and F is distributive with respect to a countably infinite number of disjunctions, i.e. for every sequence $\{P_i\}_i$ of predicates of $\mathscr{P}$, $F(\bigvee_i P_i) = \bigvee_i F(P_i)$.*

Proof. If $F \in \mathscr{R}(P)$, then $F(\bot) = \bot$ by property 1(a) and $F(\bigvee_i P_i) = \bigvee_i F(P_i)$ by property 1(e).

It remains to prove that if $F(\bot) = \bot$ and $F(\bigvee_i P_i) = \bigvee_i F(P_i)$, then $F \in \mathscr{F}(P)$.

Associate with each element q_i of Q a predicate P_{q_i} of $\mathscr{P}$ such that: $P_{q_i}(q) \Leftrightarrow q = q_i$. Obviously every predicate of $\mathscr{P}$ can be expressed as the countable disjunction of a set of predicates of this type. Let R be the relation obtained from F in the following manner:

$$(\forall q, q_i \in Q)\ ((q, q_i) \in R \Leftrightarrow F(P_{q_i})(q)).$$

Then $\forall q_i \in Q$ pre[R](P_{q_i}) = $F(P_{q_i})$. This implies, $\forall P \in \mathscr{P}$ pre[R](P) = $F(P)$: if $P = \bot$, then the equality trivially holds; else $P = \bigvee_{i \in I} P_{q_i}$, where I is a set of indices and this equality is also verified because of the distributivity of F with respect to disjunction.

Let $S = (Q, T, \{R_i\}_{i=1}^m)$ be a transition system. We represent by pre[S] and post[S] the functions

$$\text{pre}[S] = \bigvee_{i=1}^{m} \text{pre}[R_i],$$

$$\text{post}[S] = \bigvee_{i=1}^{m} \text{post}[R_i]$$

where post[R_i] = pre[R_i^{-1}] and R_i^{-1} is the inverse of the relation R_i.

Also, we put c_i = pre[R_i](⊤) and k_i = post[R_i](⊤). The predicate c_i represents the domain of R_i, i.e. the set of the states enabling t_i.

(1) For $P \in \mathscr{P}$, the predicate pre[S](P) represents the set of all the possible direct predecessors of P, i.e. all the states from which P can be reached by activating *one* transition of S. Also, post[S](P) represents the set of all the possible direct successors of P.

(2) For $P \in \mathscr{P}$, the predicate $\widetilde{\text{pre}}[S](P) = \neg\text{pre}[S](\neg P)$ represents the set of states from which it is not possible to reach $\neg P$ by activating *one* transition in S:

$$\widetilde{\text{pre}}[S](P)(q) \Leftrightarrow \forall q' \in Q\ (q \to q' \Rightarrow P(q')).$$

Obviously, $\bigwedge_{i=1}^{m} \neg c_i \subseteq \widetilde{\text{pre}}[S](P)$, for any $P \in \mathcal{P}$ ($\bigwedge_{i=1}^{m} \neg c_i$ represents the set of 'sink' states of S, i.e. the states which do not enable any transition). The characteristic set of $\widetilde{\text{pre}}[S](P)$ contains, apart from the sink states, the states having all their direct successors in P.

(3) For $P \in \mathcal{P}$, the predicate $(\widetilde{\text{pre}}[S] \wedge \text{pre}[S])(P)$ represents the set of states having at least one successor and whose every direct successor is an element of P.

The function $\widetilde{\text{pre}}[S] \wedge \text{pre}[S]$ should not be considered equal to the wp predicate transformer of Dijkstra [8] as it is remarked in [15] and [11]. In fact, for the definition of wp the relations R_i must be completed by adding transitions leading to a unique sink state when non-termination is possible [29, 1, 11].

The following properties can be proved. Also, every property derivable from these properties by substituting, pre by post, post by pre, and c_i by k_i is true.

Properties 2.

(a) For $\{P_i\}_I$ an arbitrary sequence of predicates of $\mathcal{P}$,

$$\widetilde{\text{pre}}[S]\Big(\bigwedge_i P_i\Big) = \bigwedge_i \widetilde{\text{pre}}[S](P_i);$$

(b) For $\{P_i\}_I$ an arbitrary sequence of predicates of $\mathcal{P}$,

$$\bigvee_i \widetilde{\text{pre}}[S](P_i) \subseteq \widetilde{\text{pre}}[S]\Big(\bigvee_i P_i\Big);$$

(c) $P_1 \subseteq P_2 \Rightarrow \widetilde{\text{pre}}[S](P_1) \subseteq \widetilde{\text{pre}}[S](P_2)$;

(d) $\text{pre}[S](P) \vee \text{pre}[S](\neg P) \vee (\bigwedge_{i=1}^{m} \neg c_i) = \mathsf{T}$;

(e) $(\text{pre}[S] \circ \widetilde{\text{post}}[S])(P) \subseteq P$;

(f) $P \subseteq (\widetilde{\text{pre}}[S] \circ \text{post}[S])(P)$.

Remark. Properties 2(a), 2(b) and 2(c) are the duals of 1(e), 1(f) and 1(g) respectively. Property 2(d) simply expresses the fact that every state is either a sink state or a possible direct predecessor of P or a possible direct predecessor of $\neg P$. Properties 2(e) and 2(f) can be easily proved by using the definitions of pre and $\widetilde{\text{pre}}$.

2.2. Invariants and trajectories as fixed points of monotonic functions

2.2.1. Definitions and properties

Let $S = (Q, T, \{R_i\}_{i=1}^{m})$ be a transition system.

– An *invariant* J of S is a predicate J on Q such that $\forall (q, q') \in Q \times Q$,

$$J(q) \text{ and } q \to q' \Rightarrow J(q').$$

An invariant of $S^{-1} = (Q, T, \{R_i^{-1}\}_{i=1}^{m})$ is called *inverse invariant* of S.

– A *computation* of S [19] from a given state $q_0 \in Q$ is a sequence σ over T which is applicable from q_0 and if σ is finite $(\bigwedge_{i=1}^{m} \neg c_i)(q)$ for every q such that $q_0 - \sigma \to q$. A computation is said to be *non-terminating* if it is an infinite sequence.

– A *trajectory* of S is a predicate W representing the set of the states visited by S when a computation is executed, i.e. W is a predicate on Q such that $\forall q \in Q$,

$$W(q) \Rightarrow \left(\bigwedge_{i=1}^{m} \neg c_i\right)(q) \text{ or } \exists q' \, (q \to q' \text{ and } W(q')).$$

– A *non-terminating* trajectory W of S is a trajectory corresponding to a non-terminating computation, i.e. a predicate W such that $\forall q \in Q$

$$W(q) \Rightarrow \exists q' \, (q \to q' \text{ and } W(q')).$$

Remarks. (1) If J is an invariant of S and for some $q \in Q$, $J(q)$, then J is also verified by all the possible direct successors of q (and consequently by all the possible successors of q). The notion of invariant introduced here corresponds to the notion of 'right invariant' in [24] and the notion of 'q_0-inductive' in [17].

(2) If W is a trajectory of S and for some $q \in Q$, $W(q)$, then, if q is not a sink state, there exists q', $q \to q'$, such that $W(q')$. The notion of non-terminating trajectory has been introduced in [24].

Proposition 2. *Let S be a transition system. The following propositions are equivalent:*

(a) *J is an invariant of S,*

(b) $\text{post}[S](J) \sqsubseteq J$,

(c) $J \sqsubseteq \widetilde{\text{pre}}[S](J)$.

Proof. (a) and (b) are obviously equivalent.

Suppose that $\text{post}[S](J) \sqsubseteq J$. Then, since $\widetilde{\text{pre}}[S]$ is monotonic we have, $(\widetilde{\text{pre}}[S] \circ \text{post}[S])(J) \sqsubseteq \widetilde{\text{pre}}[S](J)$ and by property 2(f), $J \sqsubseteq \widetilde{\text{pre}}[S](J)$.

Conversely suppose that $J \sqsubseteq \widetilde{\text{pre}}[S](J)$. This implies $\text{post}[S](J) \sqsubseteq (\text{post}[S] \circ \widetilde{\text{pre}}[S])(J)$ and by property 2(e) (after interchanging pre and post) we have $\text{post}[S](J) \sqsubseteq J$.

A similar proof can be carried out for Proposition 3.

Proposition 3. *Let S be a transition system. The following propositions are equivalent:*

(a) *J is an inverse invariant of S,*

(b) $\text{pre}[S](J) \sqsubseteq J$,

(c) $J \sqsubseteq \widetilde{\text{post}}[S](J)$.

Proposition 4. *J is an invariant of a transition system S iff $\neg J$ is an inverse invariant of S.*

Proof. Direct consequence of the Propositions 2 and 3.

Proposition 5. *Let S be a transition system.*

(a) *W is a trajectory of S iff W is a solution of $P \sqsubseteq \text{pre}[S](P) \vee \bigwedge_{i=1}^{m} \neg c_i$ iff $\neg W$ is a solution of $(\text{pre}[S] \wedge \widetilde{\text{pre}}[S])(P) \sqsubseteq P$.*

(b) *W is a non-terminating trajectory of S iff W is a solution of* $P \sqsubseteq \mathrm{pre}[S](P)$ *iff* $\neg W$ *is a solution of* $\widetilde{\mathrm{pre}}[S](P) \sqsubseteq P$.

Proof. The relations $P \sqsubseteq \mathrm{pre}[S](P) \vee \bigwedge_{i=1}^{m} \neg c_i$ and $P \sqsubseteq \mathrm{pre}[S](P)$ express directly the definitions of trajectory and non-terminating trajectory respectively. The proof can be completed by taking the dual of these relations (notice that by property 2(d),

$$\mathrm{pre}[S](P) \vee \bigwedge_{i=1}^{m} \neg c_i = \mathrm{pre}[S](P) \vee \widetilde{\mathrm{pre}}[S](P)).$$

2.2.2. Recall of results on the fixed points of monotonic functions

In this subsection, we recall some well-known results [34, 25] on the fixed points of monotonic functions which are used later on.

Definition. Let F be a predicate transformer element of $\mathscr{F}$.

– F is *continuous from below* or *b-continuous* iff for every increasing sequence of predicates $\{P_i\}_i$, $P_i \sqsubseteq P_{i+1}$, $i = 0, 1, 2, \ldots$: $F(\bigvee_i P_i) = \bigvee_i F(P_i)$.

– F is *continuous from above* or *a-continuous* iff for every decreasing sequence of predicates $\{P_i\}_i$, $P_{i+1} \sqsubseteq P_i$, $i = 0, 1, 2, \ldots$: $F(\bigwedge_i P_i) = \bigwedge_i F(P_i)$.

Notation. Being given $F \in \mathscr{F}$ we represent by F^* and $F^\times$ the functions

$$F^* = I \vee F \vee F^2 \vee \cdots = \bigvee_i F^i \quad (I \text{ is the identity function}),$$

$$F^\times = I \wedge F \wedge F^2 \wedge \cdots = \bigwedge_i F^i.$$

The unary operations * and $^\times$ are called respectively *starring* and *crossing*.

Proposition 6. (a) *Let F be a monotonic function of* $\mathscr{F}$, P_1 *a predicate such that* $P_1 \sqsubseteq F(P_1)$ *and* P_0 *the least fixed point of F which is greater than or equal to* P_1:

(i) $F^*(P_1) \sqsubseteq P_0$,

(ii) *if F is b-continuous, then* $F^*(P_1) = P_0$.

(b) *Let F be a monotonic function of* $\mathscr{F}$, P_1 *a predicate such that* $F(P_1) \sqsubseteq P_1$ *and* P_0 *the greatest fixed point of F which is less than or equal to* P_1:

(i) $P_0 \sqsubseteq F^\times(P_1)$,

(ii) *if F is a-continuous, then* $F^\times(P_1) = P_0$.

Proposition 7. *Let F be a monotonic function of* $\mathscr{F}$.

(a) *F is a b-continuous iff* $\tilde{F}$ *is a-continuous.*

(b) *For every predicate P of* $\mathscr{P}$, *if* P_0 *is the least fixed point of F which is greater than or equal to P, then* $\neg P_0$ *is the greatest fixed point of* $\tilde{F}$ *less than or equal to* $\neg P$ *(and conversely). Furthermore,* $F^*(P) = \neg(\tilde{F}^\times(\neg P))$.

2.2.3. Continuity of the monotonic functions constructed from elements of $\mathcal{R}(P)$

It was recalled that continuity is a sufficient condition for the iterative computation of least or greatest fixed points of a function F as the upper or least bound of the sequence of predicates $(P, F(P), F^2(P), \ldots)$. In this subsection we study under which conditions the functions constructed from elements of $\mathcal{R}(P)$ by effectuating the usual lattice operations (disjunction, conjunction, complementation) and the operations of starring and crossing are a-continuous or b-continuous. It is shown that bounded non-determinism [9, 29] for a transition system, i.e. the property that every state has a finite number of direct successors, is a necessary and sufficient condition for the predicate transformers used in Subsection 2.2.1 to be both continuous from above and from below.

Proposition 8. (a) *Every function* $\mathrm{pre}[R]$ *of* $\mathcal{R}(P)$ *is b-continuous.*

(b) *A function* $\mathrm{pre}[R]$ *of* $\mathcal{R}(P)$ *is a-continuous iff* R *is image-finite, i.e.* $\forall q \in Q\ \exists k \in \mathbb{N}$ *such that* $|\{q' \in Q \mid (q, q') \in R\}| < k$.

Proof. (a) Every function $\mathrm{pre}[R]$ is not only b-continuous but also distributive with respect to disjunction (property 1(e)).

(b) We prove first that if for every state q, $|\{q' \mid (q, q') \in \mathcal{R}\}|$ is finite, then $\mathrm{pre}[R]$ is a-continuous.

Let $\{P_i\}_i$, $P_{i+1} \sqsubseteq P_i$, $i = 0, 1, 2, \ldots$, be a decreasing sequence of predicates. Then, since $\mathrm{pre}[R]$ is a monotonic function we have, $\mathrm{pre}[R]\ (\bigwedge_i P_i) \sqsubseteq \bigwedge_i \mathrm{pre}[R](P_i)$. Thus, it remains to show that

$$(\forall q \in Q)\left(\bigwedge_i \mathrm{pre}[R](P_i)(q) \Rightarrow \mathrm{pre}[R]\left(\bigwedge_i P_i\right)(q)\right).$$

We have

$$\left(\bigwedge_i \mathrm{pre}[R](P_i)\right)(q) \Leftrightarrow \forall s \in \mathbb{N}\ (\mathrm{pre}[R](P_s))(q)$$

$$\Leftrightarrow \forall s \in \mathbb{N}\ \exists q_s \in Q\ (P_s(q_s)\ \textbf{and}\ (q, q_s) \in R)$$

$$\Rightarrow \forall s \in \mathbb{N}\ \exists q_s \in Q\left(\left(\bigwedge_{i=0}^{s} P_i\right)(q_s)\ \textbf{and}\ (q, q_s) \in R\right).$$

But since q has a finite number of direct successors, only finitely many q_s can occur. The sequence $\{P_i\}_i$ being decreasing, there exists some q_s verifying every predicate P_i. Thus,

$$\exists q_s \in Q\left(\left(\bigwedge_i P_i\right)(q_s)\ \textbf{and}\ (q, q_s) \in R\right) \Leftrightarrow \mathrm{pre}[R]\left(\bigwedge_i P_i\right)(q).$$

Suppose that there exists $q \in Q$ such that $|\{q' \mid (q, q') \in R\}|$ is infinite. Consider $\{q_{s_i}\}_i$ a sequence of distinct elements of $\{q' \mid (q, q') \in R\}$ and the decreasing sequence

of sets of indices:

$$I_0 = \mathbb{N}, I_1 = \mathbb{N} - \{0\}, \ldots, I_k = \mathbb{N} - \{0, 1, \ldots, (k-1)\}.$$

Then the sequence of predicates $\{P_i\}_i$ with $P_i = \{q_{s_j}\}_{j \in I_i}$ is also decreasing. Furthermore $\bigwedge_i P_i = \bot$ which implies that $\mathrm{pre}[R](\bigwedge_i P_i) = \bot$.

On the other hand, $\forall i \in \mathbb{N}\ \mathrm{pre}[R](P_i)(q)$ which is equivalent to $(\bigwedge_i \mathrm{pre}[R](P_i))(q)$. Consequently we have $\bigwedge_i \mathrm{pre}[R](P_i) \neq \mathrm{pre}[R](\bigwedge_i P_i)$.

Proposition 9. (a) *If F_1 and F_2 are two b-continuous functions of $\mathcal{F}$, then $F_1 \vee F_2$ and $F_1 \wedge F_2$ are b-continuous too.*

(b) *If F_1 and F_2 are two a-continuous functions of $\mathcal{F}$, then $F_1 \vee F_2$ and $F_1 \wedge F_2$ are a-continuous too.*

Proof. (a) If F_1 and F_2 are both b-continuous, then obviously $F_1 \vee F_2$ is also b-continuous.

Let $\{P_i\}_i$, $P_i \sqsubseteq P_{i+1}$, $i = 0, 1, 2, \ldots$ and increasing sequence of predicates. We have

$$F_1(P_i) \wedge F_2(P_i) \sqsubseteq F_1\Big(\bigvee_i P_i\Big) \wedge F_2\Big(\bigvee_i P_i\Big)$$

$$\Rightarrow \bigvee_i F_1(P_i) \wedge F_2(P_i) \sqsubseteq F_1\Big(\bigvee_i P_i\Big) \wedge F_2\Big(\bigvee_i P_i\Big).$$

Conversely

$$\Big(F_1\Big(\bigvee_i P_i\Big) \wedge F_2\Big(\bigvee_i P_i\Big)\Big)(q) \Rightarrow \Big(\Big(\bigvee_i F_1(P_i)\Big) \wedge \Big(\bigvee_i F_2(P_i)\Big)\Big)(q)$$

$$\Rightarrow \exists r, s \in \mathbb{N}\ F_1(P_r)(q) \text{ and } F_2(P_s)(q)$$

$$\Rightarrow (F_1(P_w) \wedge F_2(P_w))(q)$$

where $w = \mathrm{Max}\{r, s\}$

$$\Rightarrow \Big(\bigvee_i F_1(P_i) \wedge F_2(P_i)\Big)(q).$$

(b) By duality.

Lemma 1. (1) *Every constant function of $\mathcal{F}$ is both b-continuous and a-continuous.*

(2) *If F_1 and F_2 are two b-continuous (a-continuous) functions of $\mathcal{F}$, then their composition $F_1 \circ F_2$ is also b-continuous (a-continuous).*

Proposition 10. (a) *If $F \in \mathcal{F}$ is b-continuous, then F^* is b-continuous too.*

(b) *If $F \in \mathcal{F}$ is a-continuous, then $F^\times$ is a-continuous too.*

Proof. (a) Let $\{P_i\}_i$ be an increasing sequence of predicates. We have

$$F^*\Big(\bigvee_i P_i\Big) = \bigvee_j F^j\Big(\bigvee_i P_i\Big) = \bigvee_j \bigvee_i F^j(P_i) \quad \text{(Lemma 1)}$$

$$= \bigvee_i \bigvee_j F^j(P_i) = \bigvee_i F^*(P_i).$$

(b) By duality.

The following theorem is a consequence of Propositions 8, 9, 10 and Lemma 1.

Theorem 1. *Let $\mathcal{R}_b$ be the set of the image-finite relations of $\mathcal{R}$ and $\mathcal{R}_b(P)$ the set of the elements of $\mathcal{R}(P)$ defined from relations of $\mathcal{R}_b$.*

Every function constructed from elements of $\mathcal{R}_b(P)$ and constants by effectuating a finite number of times the operations of conjunction, disjunction, dualization, composition, is both a-continuous and b-continuous. Furthermore, starring preserves a-continuity and crossing preserves b-continuity.

According to this theorem all the functions constructed from pre[S] in Subsection 2.1.2 for a transition system $S = (Q, T, \{R_i\}_{i=1}^m)$ are both b-continuous and a-continuous iff the relations R_i belong to $\mathcal{R}_b$. This condition implies that the 'non-determinism' of the system S is bounded in the sense of Dijkstra [9] (see also [29, 15, 11].

In the sequel we suppose that the transition systems studied are such that the relations R_i belong to $\mathcal{R}_b$ (are image-finite).

2.2.4. Computing invariants and trajectories

Solving inequalities of the type $P \sqsubseteq G(P)$ or $H(P) \sqsubseteq P$ is equivalent to computing respectively fixed points of $I \wedge G$ and $I \vee H$ where I is the identity function:

$$P \sqsubseteq G(P) \Leftrightarrow P = P \wedge G(P) \Leftrightarrow P = (I \wedge G)(P),$$

$$H(P) \sqsubseteq P \Leftrightarrow P = P \vee H(P) \Leftrightarrow P = (I \vee H)(P).$$

Furthermore, for every predicate P_0 of $\mathcal{P}$ we have: $(I \wedge G)(P_0) \sqsubseteq P_0$ and $P_0 \sqsubseteq (I \vee H)(P_0)$. And by Proposition 6:

– if G is a-continuous, then $(I \wedge G)^\times(P_0)$ is the greatest solution of $P \sqsubseteq G(P)$ which is less than or equal to P_0;

– if G is b-continuous, then $(I \vee H)^*(P_0)$ is the least solution of $H(P) \sqsubseteq P$ which is greater than or equal to P_0.

$(I \wedge G)^\times(P_0)$ (respectively $(I \vee H)^*(P_0)$) can be computed iteratively as the limit of the decreasing (increasing) sequence $\{X_i\}_i$, defined by

$$X_{k+1} = X_k \wedge G(X_k) \quad (\text{resp. } X_{k+1} = X_k \vee H(X_k)) \quad \text{with } X_0 = P_0.$$

The general term X_k is equal to $(I \wedge G)^k(P_0)$ $((I \vee H)^k(P_0))$.

The halting criterion for this iterative computation is stabilization: there exists some $s \in \mathbb{N}$ such that $X_s = X_{s+1}$. Remark that continuity is not a crucial property from a practical point of view; even if G or H is not continuous, if stabilization is reached in such an iterative computation, then X_s is a fixed point. Thus, the hypothesis that the considered systems are of finite non-determinism is not too restrictive as far as the possibility to exploit the results exposed in this paper is concerned, while it allows a more elegant presentation.

In order to simplify the notations we represent in the sequel by pre and post the functions pre[S] and post[S] defined for a system $S = (Q, T, \{R_i\}_{i=1}^m)$. Also, we write $\bigwedge \neg c_i$ instead of $\bigwedge_{i=1}^m \neg c_i$ and $\bigvee c_i$ instead of $\bigvee_{i=1}^m c_i$ when no confusion is possible.

Proposition 11. *Let P_0 a predicate of $\mathcal{P}$.*

(a) $\widetilde{\text{pre}}^{\times}(P_0)$ *is the greatest invariant of S less than or equal to P_0.*

(b) $\text{post}^*(P_0)$ *is the least invariant of S greater than or equal to P_0.*

(c) $\widetilde{\text{post}}^{\times}(P_0)$ *is the greatest inverse invariant of S less than or equal to P_0.*

(d) $\text{pre}^*(P_0)$ *is the least inverse invariant of S greater than or equal to P_0.*

Proof. This proposition is a direct consequence of the results of the preceding subsection if it is taken into account that $(I \vee F)^*(P) = F^*(P)$ if F is distributive with respect to disjunction and $(I \wedge F)^{\times}(P) = F^{\times}(P)$ if F is distributive with respect to conjunction.

Proposition 12. (a) *J is the greatest invariant of S less than or equal to $P_0 \in \mathcal{P}$ iff $\neg J$ is the least inverse invariant of S greater than or equal to $\neg P_0$, i.e.* $J = \widetilde{\text{pre}}^{\times}(P_0) = \neg\text{pre}^*(\neg P_0)$.

(b) *J is the least invariant of S greater than or equal to $P_0 \in \mathcal{P}$ iff $\neg J$ is the greatest inverse invariant of S less than or equal to $\neg P_0$, i.e.* $J = \text{post}^*(P_0) = \neg\widetilde{\text{post}}^{\times}(\neg P_0)$.

Proof. Direct consequence of the preceding proposition and Proposition 7(b).

Observation. The predictions $\text{pre}^*(P_0)$ and $\text{post}^*(P_0)$ represent respectively the set of all the (possible) predecessors and the set of all the (possible) successors of the states verifying P_0. Thus, Proposition 12(a) means that the greatest invariant contained in P_0 is equal to the complement of the predicate representing the set of all the possible predecessors of $\neg P_0$.

Proposition 13. *The set of the invariants of a transition system S forms a distributive sub-lattice of $\mathcal{L}$.*

Proof. Let J_1 and J_2 be two invariants, $J_1 = \widetilde{\text{pre}}(J_1)$ and $J_2 = \widetilde{\text{pre}}(J_2)$.

Then,

$$J_1 \wedge J_2 = \widetilde{\text{pre}}(J_1) \wedge \widetilde{\text{pre}}(J_2) = \widetilde{\text{pre}}(J_1 \wedge J_2).$$

Also,

$$J_1 \vee J_2 = \widetilde{\text{pre}}(J_1) \vee \widetilde{\text{pre}}(J_2) = \widetilde{\text{pre}}(J_1 \vee J_2).$$

The greatest and the least invariant are respectively $\top$ and $\bot$.

$\mathscr{L}$ being distributive, the lattice of the invariants is distributive too.

Observation. The atomic elements of the lattice of the invariants of a transition system S are predicates whose characteristic set contains either a sink state or the states of a strongly connected component of the graph associated with S.

Definition. An invariant J of S is called *deadlock-free* invariant if $J \sqsubseteq \bigvee c_i$.

Observation. If J is a deadlock-free invariant of S and S is initialized at a state q, such that $J(q)$, then S can never reach a sink state (it never deadlocks).

Proposition 14. *J is a deadlock-free invariant of S iff $J \sqsubseteq (\text{pre} \wedge \widetilde{\text{pre}})(J)$.*

Proof. Omitted.

Observation. The greatest deadlock-free invariant of S is $(I \wedge \text{pre} \wedge \widetilde{\text{pre}})^{\times}(\top) = (I \wedge \text{pre} \wedge \widetilde{\text{pre}})^{\times}(\bigvee c_i)$. If we remark that $\text{pre}(\bigvee c_i) \sqsubseteq \bigvee c_i$, we find that the greatest deadlock-free invariant of S is equal to $\widetilde{\text{pre}}^{\times}(\bigvee c_i)$, i.e. it is the greatest invariant less than or equal to $\bigvee c_i$. Also, if we use the equality $F^*(P) = \neg(\tilde{F}^{\times}(\neg P))$ (Proposition 7), we find that the greatest deadlock-free invariant is equal to $\neg(\text{pre}^*(\bigwedge \neg c_i))$; this predicate characterizes the set of the states which are not possible predecessors of the set of the sink states.

Proposition 15. (a) *The greatest trajectory, of a transition system S, less than or equal to $P_0 \in \mathscr{P}$ is W*

$$W = (I \wedge (\text{pre} \vee \widetilde{\text{pre}}))^{\times}(P_0) = \neg(I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(\neg P_0).$$

(b) *The greatest non-terminating trajectory of a transition system S, less than or equal to $P_0 \in \mathscr{P}$ is W*

$$W = (I \wedge \text{pre})^{\times}(P_0) = \neg(I \vee \widetilde{\text{pre}})^*(\neg P_0).$$

Proof. Omitted.

Proposition 16. *Let S be a transition system, $P_0 \in \mathscr{P}$ and q a state of S.*

(a) $\text{pre}^*(P_0)(q)$ *iff there exists a computation from q such that for some state q' visited in this computation $P_0(q')$.*

(b) $(I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_0)(q)$ *iff for every computation from q there exists a state q' visited in this computation such that $P_0(q')$.*

(c) $(I \vee \widetilde{\text{pre}})^*(P_0)(q)$ *iff for every non-terminating computation from* q *there exists a state* q' *visited in this computation such that* $P_0(q')$.

Proof. (a) Notice that $\text{pre}^*(P_0)(q)$ means that q is a possible predecessor of some state q', $P_0(q')$, i.e. there exists q', $P_0(q')$, and a sequence σ over T such that $q - \sigma \rightarrow q'$. So, σ can be the prefix of a computation starting from q.

(b) Suppose that $(I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_0)(q)$ and that there exists a computation from q such that no one of the states visited in this computation satisfies P_0. This means that there exists a trajectory W, corresponding to this computation, such that $(I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_0) \wedge W \neq \bot$ and $W \wedge P_0 = \bot$. By Proposition 15(a), $W_0 = \neg(I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_0) = (I \wedge (\text{pre} \vee \widetilde{\text{pre}}))^\times(\neg P_0)$ is the greatest trajectory contained in $\neg P_0$.

From $W \wedge P_0 = \bot$ we have that W is a trajectory contained in $\neg P_0$ and consequently $W \subseteq W_0$. But this contradicts the hypothesis $\neg W_0 \wedge W \neq \bot$. Conversely, suppose that for every computation from q there exists a state q' such that $P_0(q')$ and that $\neg(I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_0)(q)$. Then, according to Proposition 15(a) q belongs to the greatest trajectory contained in $\neg P_0$ and this contradicts the hypothesis.

(c) A similar proof can be carried out.

Proposition 17. *Let* S *be a transition system,* $P_0 \in \mathcal{P}$ *and* q *a state of* S.

(a) *If* $\text{pre}^*(P_0)(q)$, *then there exists a trajectory* W, $W(q)$, *such that* $W \wedge P_0 \neq \bot$.

(b) $(I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_0)(q)$ *iff for every trajectory* W, $W(q)$ *implies that* $W \wedge P_0 \neq \bot$.

(c) $(I \vee \widetilde{\text{pre}})^*(P_0)(q)$ *iff for every non-terminating trajectory* W, $W(q)$ *implies that* $W \wedge P_0 \neq \bot$.

Proof. A proof similar to the proof of proposition 16 can be carried out. However, notice that the converse of (a) is not true: if q is a possible successor of some state of P_0 but it is not a possible predecessor of any state of P_0, then there exists a trajectory W such that $W(q)$ and $W \wedge P_0 \neq \bot$ but $\neg\text{pre}^*(P_0)(q)$.

Observation. The predicate $A = (I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_0)$ is the least solution of $(\text{pre} \wedge \widetilde{\text{pre}})(P) \subseteq P$ which is greater than or equal to P_0. Thus, A contains all the states which are not sink states and all their direct successors belong to A. In a similar manner, the predicate $B = (I \vee \widetilde{\text{pre}})^*(P_0)$ is the least solution of $\widetilde{\text{pre}}(P) \subseteq P$ which is greater than or equal to P_0. B contains all the states which if they are not sink states then all their direct successors belong to B.

Proposition 18. *The greatest non-terminating trajectory of a transition system is equal to* $W = \text{pre}^\times(\bigvee_i c_i)$.

Proof. The greatest non-terminating trajectory is equal to $(I \wedge \text{pre})^\times(\top)$. By taking into account that $\text{pre}(\top) = \bigvee_i c_i$ and that $\text{pre}(\bigvee_i c_i) \subseteq \bigvee_i c_i$ we obtain: $(I \wedge \text{pre})^\times(\top) = \text{pre}^\times(\bigvee_i c_i)$.

3. The properties of transition systems

In the introduction of this paper we adopted the hypothesis that the problem of system verification with respect to a given specification can be decomposed in two parts: one part corresponds to invariant properties and the other to non-invariant or 'time dependent' ones.

The aim of this section is to provide general definitions of these two classes of properties and general methods for their verification as well. Since most of the properties of a system depend on the choice of its initial state, we consider in the sequel that a system is a doublet (S, P_1) where S is a transition system and P_1 a predicat representing the set of all the possible initial states of S.

3.1. Reachability types

The starting point of our approach is that every property of a given system (S, P_1) is specified as a pair (P_2, RT) where,

– P_2 is a predicate characterizing a set of *target* states;

– RT is a *reachability type* which specifies *how* states of P_2 can be reached when the system is initialized at an arbitrary state of P_1.

The choice of P_2 determines which kind of events are of interest for a property while the reachability type determines how often these events can or should occur. For example, a termination property is defined by taking $P_2 = \text{HALT}$ (**HALT** contains all the final states) and by choosing a reachability type which can express the facts "every computation terminates properly" or "there exists some computation terminating properly" or "for any prefix σ of an arbitrary computation from a state q_0, a sequence x of transitions can be found such that $q_0 - \sigma x \rightarrow q$ and HALT(q)", etc.

The reachability types correspond to different types of 'causality relation' or 'temporal implication' which can exist between the two statements: "the system is initialized at a state of P_1" and "the system is at a state of P_2".

We introduce here ten different reachability types. One of them is used to characterize invariant properties and it expresses the fact that something always holds after the system is initialized at a state of P_1. The other nine reachability types express the idea that something will eventually become true in the future (but after becoming true it can become false again) and they are used for defining non-invariant properties.

Definitions. Let S be a transition system and $P_1, P_2 \in \mathcal{P}$,

– P_2 is *potentially reachable* or *p-reachable* from P_1 if for every state q, $P_1(q)$, there exists a computation from q such that for some state q' visited in this computation $P_2(q')$, i.e. $P_1 \sqsubseteq \text{pre}^*(P_2)$, according to Proposition 16(a).

– P_2 is *obligatorily reachable* or *o-reachable* from P_1 if for every state q, $P_1(q)$, and every non-terminating computation from q there exists a state q' visited in this

computation such that $P_2(q')$. According to Proposition 16(c), in this case, $P_1 \sqsubseteq (I \vee \widetilde{\text{pre}})^*(P_2)$.

- P_2 is *inevitably reachable* or *a-reachable* from P_1 if for every state q, $P_1(q)$, and every computation from q there exists a state q' visited in this computation such that $P_2(q')$. According to Proposition 16(b) in this case, $P_1 \sqsubseteq (I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_2)$.
- P_2 is *always reachable* or *a-reachable* from P_1 if for every state q, $P_1(q)$, every possible successor q' of q ($q \rightarrow^* q'$) is such that $P_2(q')$, i.e. $\text{post}^*(P_1) \sqsubseteq P_2$.

The first three reachability types in the preceding definition are called types of *simple* reachability. They characterize situations where a state of P_2 is reachable *at least once* from every state of P_1. A fundamental difference between simple reachability and a-reachability is that in the former case the predicate P_2 becomes true in an intermittent manner while in the latter, P_2 is permanently true.

Observation. There is some similarity in our approach with the one followed when modal or temporal logic is used for the study of programs (see for example [23, 27, 21]). In fact, some reachability types correspond to modalities of a modal logic for which a (Kripke) frame is given by the doublet $(Q, \rightarrow^*)$. By using the terminology in [21] we can say that the assertion "P_2 is a-reachable from P_1" means that P_2 is true 'now' (at every state of P_1) and it will remain true during all the possible 'futures' (a 'future' corresponds to a computation in our model). The types of simple reachability give assertions that a statement is 'eventually" true: "P_2 is p-reachable" means that there exists some 'future' at some 'time' of which P_2 is reachable; "P_2 is o-reachable" means that for every non-terminating 'future' there is some 'time' at which P_2 is reached; "P_2 is i-reachable" means that for every possible 'future' there is some 'time' at which P_2 is reached.

Hereafter we introduce three more types of 'intermittent' reachability which will be called types of *systematic reachability*. They are used to qualify the situations where P_2 is reachable not only from every state which satisfies P_1 but also from every possible successor of it; i.e. "systematically reachable from P_1" means "simply reachable from $\text{post}^*(P_1)$" and it implies that P_2 is reachable an unbounded number of times if the system is initialized correctly.

Definitions. Let S be a transition system and $P_1, P_2 \in \mathcal{P}$.

- P_2 is *p-systematically reachable* or *p-s-reachable* from P_1 if P_2 is p-reachable from $\text{post}^*(P_1)$, i.e. $\text{post}^*(P_1) \sqsubseteq \text{pre}^*(P_2)$.
- P_2 is *o-systematically reachable* or *o-s-reachable* from P_1 if P_2 is o-reachable from $\text{post}^*(P_1)$, i.e. $\text{post}^*(P_1) \sqsubseteq (I \vee \widetilde{\text{pre}})^*(P_2)$.
- P_2 is *i-systematically reachable* or *i-s-reachable* from P_1 if P_2 is i-reachable from $\text{post}^*(P_1)$, i.e. $\text{post}^*(P_1) \sqsubseteq (I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_2)$.

Observation. Notice that the different types of systematic reachability can be considered as cases of a-reachability of $\mathrm{pre}^*(P_2)$, $(I \vee \widetilde{\mathrm{pre}})^*(P_2)$ and $(I \vee \mathrm{pre} \wedge \widetilde{\mathrm{pre}})^*(P_2)$ from P_1. For example, "$\mathrm{pre}^*(P_2)$ is always reachable from P_1" is equivalent to "P_2 is p-s-reachable from P_1" or to "P_2 is p-reachable from $\mathrm{post}^*(P_1)$".

Finally, we introduce the notion of *quasi-systematic reachability* in order to characterize the cases of simple but non-systematic reachability; the types of quasi-systematic reachability can be used for expressing properties such as partial liveness, partial deadlock and livelock.

Definitions. Let S be a transition system and P_1, $P_2 \in \mathscr{P}$. For $j \in \{p, o, i\}$, P_2 is *j-quasisystematically reachable* from P_1 or *j-qs-reachable* from P_1 if P_2 is j-reachable from P_1 and P_2 is not j-s-reachable from P_1.

3.2. The reachability types as relations involving invariants and trajectories

Proposition 19. *For a transition system S, P_1, $P_2 \in \mathscr{P}$,*

(a) *P_2 is p-reachable from P_1 in S iff for every invariant J of S less than or equal to $\neg P_2$, $P_1 \wedge J = \bot$;*

(b) *P_2 is o-reachable from P_1 in S iff for every non-terminating trajectory W of S less than or equal to $\neg P_2$, $P_1 \wedge W = \bot$;*

(c) *P_2 is i-reachable from P_1 in S iff from every trajectory W of S less than or equal to $\neg P_2$, $P_1 \wedge W = \bot$.*

Proof. (a) P_2 is p-reachable from $P_1 \Leftrightarrow P_1 \wedge \neg\mathrm{pre}^*(P_2) = \bot$. According to Proposition 12(a) $\neg\mathrm{pre}^*(P_2) = \widetilde{\mathrm{pre}}^\times(\neg P_2)$ is the greatest invariant less than or equal to $\neg P_2$. Thus, for every invariant J, $J \sqsubseteq \neg P_2$, we have, $P_1 \wedge J = \bot$.

(b) P_2 is o-reachable from $P_1 \Leftrightarrow P_1 \wedge \neg(I \vee \widetilde{\mathrm{pre}})^*(P_2) = \bot$. According to Proposition 15(b) $\neg(I \vee \widetilde{\mathrm{pre}})^*(P_2) = (I \wedge \mathrm{pre})^\times(\neg P_2)$ is the greatest non-terminating trajectory contained in $\neg P_2$. Thus for every non-terminating trajectory W contained in $\neg P_2$ we have, $W \wedge P_1 = \bot$.

(c) A similar proof can be carried out.

Proposition 20. *For a transition system S, P_1, $P_2 \in \mathscr{P}$, the following expressions are equivalent:*

(a) *P_2 is a-reachable from P_1;*

(b) $\mathrm{post}^*(P_1) \sqsubseteq P_2$;

(c) $P_1 \sqsubseteq \widetilde{\mathrm{pre}}^\times(P_2)$;

(d) *there exists an invariant J of S such that $P_1 \sqsubseteq J \sqsubseteq P_2$.*

Proof. Obviously, (a) is equivalent to (b).

(c) If $\mathrm{post}^*(P_1) \sqsubseteq P_2$, then, since $\widetilde{\mathrm{pre}}^\times$ is a monotonic operator, we have $\widetilde{\mathrm{pre}}^\times \circ \mathrm{post}^*(P_1) \sqsubseteq \widetilde{\mathrm{pre}}^\times(P_2)$. But $\mathrm{post}^*(P_1)$ is an invariant and $\widetilde{\mathrm{pre}}^\times \circ \mathrm{post}^*(P_1) =$

post*(P_1). Thus, post*$(P_1) \sqsubseteq \widetilde{\text{pre}}^{\times}(P_2)$ and from $P_1 \sqsubseteq$ post*(P_1) we have $P_1 \sqsubseteq \widetilde{\text{pre}}^{\times}(P_2)$. If $P_1 \sqsubseteq \widetilde{\text{pre}}^{\times}(P_2)$, then post*$(P_1) \sqsubseteq$ post* $\circ\, \widetilde{\text{pre}}^{\times}(P_2)$. But post* $\circ\, \widetilde{\text{pre}}^{\times}(P_2) = \widetilde{\text{pre}}^{\times}(P_2)$ since $\widetilde{\text{pre}}^{\times}(P_2)$ is an invariant. Thus, post*$(P_1) \sqsubseteq \widetilde{\text{pre}}^{\times}(P_2)$ and from $\widetilde{\text{pre}}^{\times}(P_2) \sqsubseteq P_2$ we have post*$(P_1) \sqsubseteq P_2$.

(d) If there exists an invariant J, $P_1 \sqsubseteq J \sqsubseteq P_2$, then post*$(P_1) \sqsubseteq J$ and $J \sqsubseteq P_2$ which implies post*$(P_1) \sqsubseteq P_2$. If post*$(P_1) \sqsubseteq P_2$, then by taking $J =$ post*(P_1) we have $P_1 \sqsubseteq J \sqsubseteq P_2$.

Proposition 21. *For a transition system S, P_1, $P_2 \in \mathcal{P}$, the following expressions are equivalent:*

(a) *P_2 is p-s-reachable from P_1 in S;*

(b) $P_1 \sqsubseteq \widetilde{\text{pre}}^{\times}(\text{pre}^*(P_2))$,

(c) *for every invariant J of S, $J \sqsubseteq \neg P_2$, post*$(P_1) \wedge J = \bot$;*

(d) *there is no invariant J of S, $J \neq \bot$, such that $J \sqsubseteq$ post*$(P_1) \wedge \neg P_2$.*

Proof. By Proposition 20, (a) is equivalent to (b). Also, by Proposition 19(a) and the definition of p-s-reachability (a) is equivalent to (c). Hereafter we prove that (a) is equivalent to (d). Suppose that post*$(P_1) \sqsubseteq$ pre*(P_2) and that there exists an invariant J, $J \neq \bot$, $J \sqsubseteq$ post*$(P_1) \wedge \neg P_2$. Then, $J \sqsubseteq$ pre*(P_2) which implies that $J \not\sqsubseteq \neg P_2$ (contradiction).

Conversely, suppose that P_2 is not 1-s-reachable from P_1. Then, post*$(P_1) \wedge \neg$pre*$(P_2) \neq \bot$. The predicate post*$(P_1) \wedge \neg$pre*(P_2), being the conjunction of two invariants, is an invariant less than or equal to both post*(P_1) and $\neg P_2$.

Proposition 22. *For a transition system S, P_1, $P_2 \in \mathcal{P}$, the following expressions are equivalent:*

(a) *P_2 is o-s-reachable from P_1 in S;*

(b) $P_1 \sqsubseteq \widetilde{\text{pre}}^{\times}(I \vee \widetilde{\text{pre}})^*(P_2)$;

(c) *for every non-terminating trajectory W of S less than or equal to $\neg P_2$, post*$(P_1) \wedge W = \bot$;*

(d) *there is no non-terminating trajectory W of S, $W \neq \bot$, such that $W \sqsubseteq$ post*$(P_1) \wedge \neg P_2$.*

Proof. Similar to the preceding one. Notice that the conjunction of an invariant and of a non-terminating trajectory is a non-terminating trajectory.

Proposition 23. *For a transition system S, P_1, $P_2 \in \mathcal{P}$, the following propositions are equivalent:*

(a) *P_2 is i-s-reachable from P_1 in S;*

(b) $P_1 \sqsubseteq \widetilde{\text{pre}}^{\times}(I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_2)$.

(c) *for every trajectory W of S, $W \sqsubseteq \neg P_2$, post*$(P_1) \wedge W = \bot$;*

(d) *there is no trajectory W of S, $W \neq \bot$, such that $W \sqsubseteq$ post*$(P_1) \wedge \neg P_2$.*

Proof. Omitted.

Observations. (a) According to Proposition 19 if P_2 is such that $\widetilde{\text{pre}}^{\times}(\neg P_2) = \bot$ (respectively $(I \wedge \text{pre})^{\times}(\neg P_2) = \bot$, $(I \wedge (\text{pre} \vee \widetilde{\text{pre}}))^{\times}(\neg P_2) = \bot$), then P_2 is p-reachable (respectively o-reachable, i-reachable) from every possible initial state.

(b) The greatest invariant from which a predicate P_2 is p-s-reachable (respectively o-s-reachable, i-s-reachable) is $\widetilde{\text{pre}}^{\times}(\text{pre}^*(P_2))$ (respectively, $\widetilde{\text{pre}}^{\times}(I \vee \widetilde{\text{pre}})^*(P_2)$, $\widetilde{\text{pre}}^{\times}(I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_2)$).

For an overview of simple and systematic reachabilities see Table 1.

The main result of this subsection is that it is possible to express the ten defined reachability types in terms of relations between on the one hand, the 'input predicate' P_1 or the least invariant containing P_1 and on the other hand, the greatest invariant, non-terminating trajectory, trajectory less than or equal to $\neg P_2$. These relations are illustrated in Fig. 1.

3.3. Study of the properties

In this subsection, we show how the results of 3.1 and 3.2 can be applied to prove system properties.

Definition. A *property* of a given system (S, P_1) is specified as a pair (P_2, RT) where P_2 is an arbitrary predicate of $\mathcal{P}$ and RT one of the ten reachability types defined in 3.1. We say that the system (S, P_1) *satisfies* the property (P_2, RT) if P_2 is RT from P_1 in S.

3.3.1. Invariant properties

For a system (S, P_1) an *invariant property* is any property that is specified as a pair $(P_2, \text{a-reachable})$, where P_2 is an arbitrary predicate of $\mathcal{P}$.

According to Proposition 20, in order to prove that a system satisfies an invariant property one has to find some invariant J of S such that $P_1 \sqsubseteq J \sqsubseteq P_2$ and such an invariant exists iff $\text{post}^*(P_1) \sqsubseteq P_2$ or $\widetilde{\text{pre}}^{\times}(P_2) \sqsubseteq P_1$. Thus the validity of an invariant property can be established by computing either $\text{post}^*(P_1)$ or $\widetilde{\text{pre}}^{\times}(P_2)$.

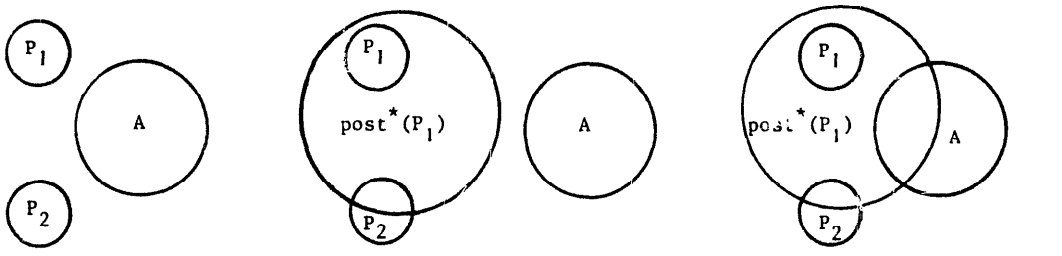
Notice that if a system satisfies an invariant property with target predicate P_2, then it satisfies every invariant property with target predicate P_2', $P_2 \sqsubseteq P_2'$. Also, in this case, $(J, \text{a-reachable})$ is a valid invariant property for every invariant J, $P_1 \sqsubseteq J \sqsubseteq P_2$.

We believe that the distinction between the notions of invariant and of invariant property is important since both of them have to be used in order to prove that an assertion about a system 'always' holds. Suppose for instance that we want to prove that a mutual exclusion constraint always holds in a given system (S, P_1). According to our approach, one has to prove that (S, P_1) satisfies the invariant property $(P_2, \text{a-reachable})$ where P_2 is a predicate characterizing all the states for which mutual exclusion holds. It is important to note that P_2 is given in the specification (or can be deduced from the specifications) of (S, P_1) and, as a rule,

Table 1
Simple and systematic reachability

Notions of reachability Y / Modalities X	Reachable	Systematically reachable	
Potentially	$P_1 \sqsubseteq \text{pre}^*(P_2)$	$\text{post}^*(P_1) \sqsubseteq \text{pre}^*(P_2)$	$\Leftrightarrow P_1 \sqsubseteq \widetilde{\text{pre}}^{\times} \circ \text{pre}^*(P_2)$
	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$
	$P_1 \wedge \widetilde{\text{pre}}^{\times}(\neg P_2) = \bot$	$\text{post}^*(P_1) \wedge \widetilde{\text{pre}}^{\times}(\neg P_2) = \bot$	$\Leftrightarrow P_1 \wedge \text{pre}^* \circ \widetilde{\text{pre}}^{\times}(\neg P_2) = \bot$
Obligatorily	$P_1 \sqsubseteq (I \vee \widetilde{\text{pre}})^*(P_2)$	$\text{post}^*(P_1) \sqsubseteq (I \vee \widetilde{\text{pre}})^*(P_2)$	$\Leftrightarrow P_1 \sqsubseteq \widetilde{\text{pre}}^{\times} \circ (I \vee \widetilde{\text{pre}})^*(P_2)$
	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$
	$P_1 \wedge (I \wedge \text{pre})^{\times}(\neg P_2) = \bot$	$\text{post}^*(P_1) \wedge (I \wedge \widetilde{\text{pre}})^{\times}(\neg P_2) = \bot$	$\Leftrightarrow P_1 \wedge \text{pre}^* \circ (I \wedge \text{pre})^{\times}(\neg P_2) = \bot$
Inevitably	$P_1 \sqsubseteq (I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(P_2)$	$\text{post}^*(P_1) \sqsubseteq (I \vee \widetilde{\text{pre}} \wedge \text{pre})^*(P_2)$	$\Leftrightarrow P_1 \sqsubseteq \widetilde{\text{pre}}^{\times} \circ (I \vee \widetilde{\text{pre}} \wedge \text{pre})^*(P_2)$
	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$
	$P_1 \wedge (I \wedge (\text{pre} \vee \widetilde{\text{pre}}))^{\times}(\neg P_2) = \bot$	$\text{post}^*(P_1) \wedge (I \wedge (\text{pre} \vee \widetilde{\text{pre}}))^{\times}(\neg P_2) = \bot$	$\Leftrightarrow P_1 \wedge \text{pre}^*(I \wedge (\text{pre} \vee \widetilde{\text{pre}}))^{\times}(\neg P_2) = \bot$

P_2 is 'X' 'Y' from P_1



simple reachability systematic reachability quasi-systematic reachability

Definition of A :

p -**-reachable : A is the greatest invariant contained in $\neg P_2$

o -**-reachable : A is the greatest non-terminating trajectory contained in $\neg P_2$

i -**-reachable : A is the greatest trajectory contained in $\neg P_2$

(** stands for the prefixes "s", "qs" or the absence of prefixe).

Fig. 1.

P_2 is not an invariant of S. In order to establish that (P_2, a-reachable) is satisfied one has to find some invariant J of S such that it implies the mutual exclusion constraint ($J \subseteq P_2$) and all the possible initial states satisfy this invariant ($P_1 \subseteq J$).

3.3.2. Non-invariant properties

For a system (S, P_1), a *non-invariant property* is any property which is specified as a pair (P_2, RT), where P_2 is an arbitrary predicate of $\mathcal{P}$ and RT one of the nine types of simple, systematic, quasi-systematic reachability defined in 3.1.

According to the suggested approach, a *family* of non-invariant properties is defined by fixing the target predicate P_2 and it potentially contains nine different properties corresponding to the different reachability types. The results of subsection 3.2 applied for a given family provide proof methods for each one of its properties.

Apart from the two families of properties studied hereafter, other families which are of interest in practice can be defined by appropriately choosing the target predicate P_2. The reader can find an application of this idea in [32] where the properties relative to the presence of deadlock and livelock are studied.

3.3.2.1. Blockability properties

Definition. A transition system S is said to be *'x'-blockable* from P_1 if (S, P_1) satisfies the non-invariant property ($\bigwedge \neg c_i$, 'x'-reachable) where 'x' stands for one of the nine possible prefixes of '-reachable'.

Lemma 2. *If P_2 is a predicate such that $\bigwedge \neg c_i \subseteq P_2$, then*

$$P_2 \vee \widetilde{\text{pre}}(P_2) = P_2 \vee \widetilde{\text{pre}}(P_2) \wedge \text{pre}(P_2).$$

Proof. Omitted.

Proposition 24. *For a transition system S the following expressions are equivalent:*
(a) *S is o-blockable from* P_1;
(b) *S is i-blockable from* P_1;
(c) *S is o-s-blockable from* P_1;
(d) *S is i-s-blockable from* P_1.

Proof. By Lemma 2, (a) is equivalent to (b) and (c) is equivalent to (d). Furthermore, if $P_1 \sqsubseteq (I \vee \widetilde{\text{pre}})^*(\bigwedge \neg c_i)$, then every possible successor of a state of P_1 verifies the predicate $(I \vee \widetilde{\text{pre}})^*(\bigwedge \neg c_i)$. Thus, $\text{post}^*(P_1) \sqsubseteq (I \vee \widetilde{\text{pre}})^*(\bigwedge \neg c_i)$, i.e. (a) is equivalent to (c).

Corollary. *For a transition system S the following propositions are contradictions:*
(a) *S is o-qs-blockabie from* P_1;
(b) *S is i-qs-blockable from* P_1.

The following three propositions are direct consequences of the Propositions 19(a), 21 and 19(b) respectively.

Proposition 25. *A transition system S is p-blockable from* P_1 *iff for every deadlock-free invariant J of S,* $P_1 \wedge J = \bot$.

Proposition 26. *For a transition system S the following expressions are equivalent:*
(a) *S is p-s-blockable from* P_1;
(b) $P_1 \sqsubseteq \widetilde{\text{pre}}^{\times}(\neg J)$, *where J is the greatest deadlock-free invariant of S.*
(c) *for every deadlock-free invariant J of S,* $J \wedge \text{post}^*(P_1) = \bot$;
(d) *there is no invariant J of S,* $J \neq \bot$, *such that* $J \sqsubseteq \text{post}^*(P_1) \wedge (\bigvee c_i)$.

Proposition 27. *A transition system S is o-s-blockable from* P_1 *iff for every non-terminating trajectory W,* $P_1 \wedge W = \bot$.

Observations. (a) If *S* is such that the greatest deadlock-free invariant $\widetilde{\text{pre}}^{\times}(\bigvee c_i) = \bot$, then *S* is potentially blockable from every possible initial state.

(b) If *S* is such that the greatest non-terminating trajectory $\text{pre}^{\times}(\bigvee c_i) = \bot$, then *S* is inevitably blockable from every possible initial state.

(c) The greatest invariant *J* such that *S* be p-s-blockable from *J* is equal to $\widetilde{\text{pre}}^{\times}(\text{pre}^*(\bigwedge \neg c_i)) = \neg\text{pre}^*(\neg\text{pre}^*(\bigwedge \neg c_i)) = \neg\text{pre}^*(\widetilde{\text{pre}}^{\times}(\bigvee c_i))$. $\widetilde{\text{pre}}^{\times}(\bigvee c_i)$ is the greatest deadlock-free invariant and $\neg\text{pre}^*(\text{pre}^{\times}(\bigvee c_i))$ represents the set of the states from which this invariant is not reachable.

(d) The greatest invariant *J* such that *S* be o-s-blockable from *J* is equal to $\widetilde{\text{pre}}^{\times}((I \vee \widetilde{\text{pre}})^*(\bigwedge \neg c_i)) = \neg\text{pre}^*(\neg(I \vee \widetilde{\text{pre}})^*(\bigwedge \neg c_i)) = \neg\text{pre}^*((I \wedge \text{pre})^{\times}(\bigvee c_i))$. This

predicate represents the set of the states from which the greatest non-terminating trajectory is not reachable.

Fig. 2 represents the results of this subsection.

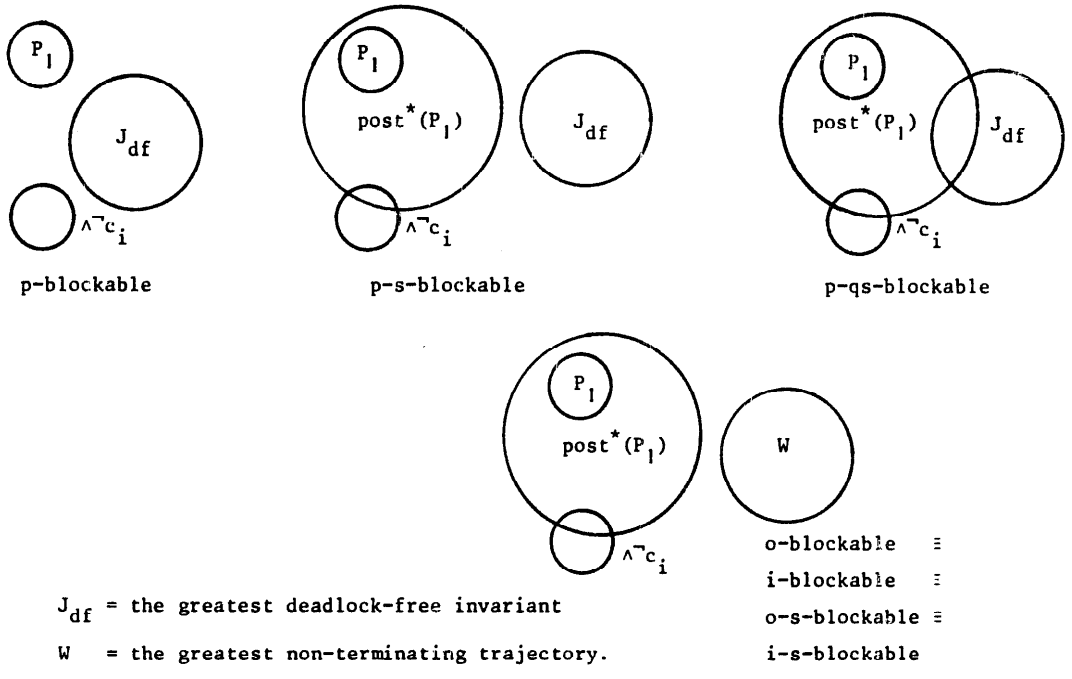


Fig. 2.

3.3.2.2. Activability properties

Activability of a set of transitions

Definition. Let $S=(Q, T, \{R_i\}_{i=1}^{m})$ be a transition system, $P_1 \in \mathcal{P}$ and $L \subseteq \{1, 2, \ldots, m\}$, $L \neq \emptyset$.

A set of transitions $\{t_i\}_{i\in L}$ is said to be 'x'-*activable* from P_1 in S if (S, P_1) satisfies the non-invariant property ($\bigvee_{i\in L} c_i$, 'x'-reachable) where 'x' stands for one of the nine possible prefixes of '-reachable'.

Obviously, the properties prefixed by 'o' do not represent any practical interest. Fig. 3 shows how the other types of activability can be characterized by using Propositions 19(a), 19(c), 21 and 23.

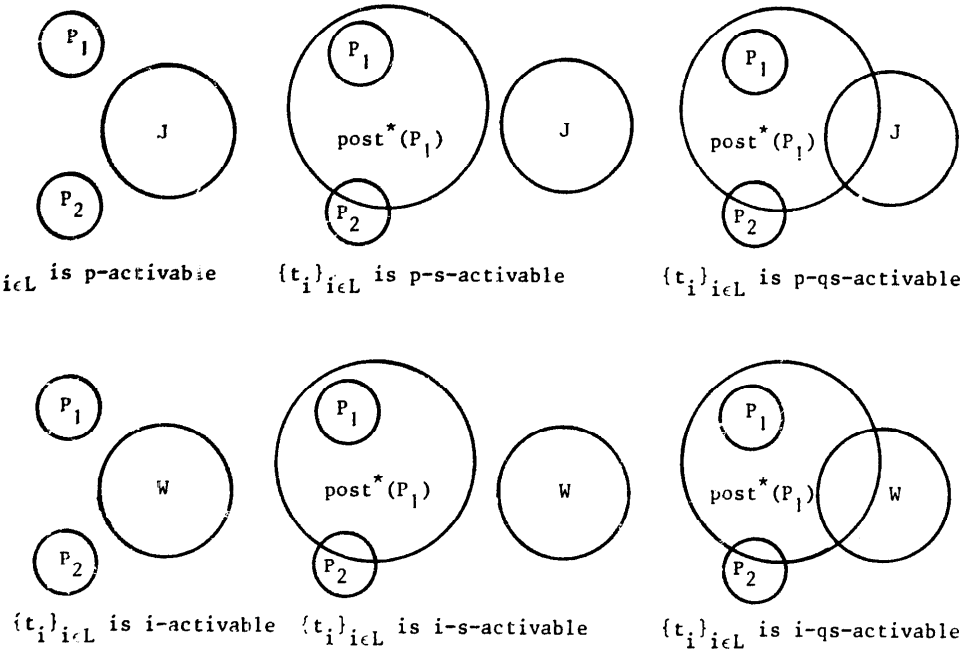
If $|L|=1$, then the properties prefixed by 'p' correspond to liveness properties for a transition [16]: these properties express the *possibility* of activating a transition without however guaranteeing that it will be effectively enabled. The properties prefixed by 'i' express to which extent the enabling of a transition will inevitably take place and they can characterize the absence or presence of a certain type of livelock [19, 32].

Observations. (a) If the greatest invariant (trajectory) contained in $\bigwedge_{i\in L}\neg c_i$ is equal to $\bot$, then $\{t_i\}_{i\in L}$ is p-s-activable (i-s-activable) from every possible initial state.

(b) The greatest invariant from which $\{t_i\}_{i\in L}$ is p-s-activable is equal to

$$\widetilde{\text{pre}}^{\times}\left(\text{pre}^*\left(\bigvee_{i\in L} c_i\right)\right) = \neg\text{pre}^*\left(\neg\text{pre}^*\left(\bigvee_{i\in L} c_i\right)\right) = \neg\text{pre}^*\left(\widetilde{\text{pre}}^{\times}\left(\bigwedge_{i\in L}\neg c_i\right)\right).$$

(c) The greatest invariant from which $\{t_i\}_{i\in L}$ is i-s-activable is equal to $\widetilde{\text{pre}}^{\times}((I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(\bigvee_{i\in L} c_i)) = \neg\text{pre}^*(W)$, where W is the greatest trajectory less than or equal to $\bigwedge_{i\in L}\neg c_i$.



$P_2 = \bigvee_{i\in L} c_i$

J = the greatest invariant contained in $\bigwedge_{i\in L}\neg c_i$

W = the greatest trajectory contained in $\bigwedge_{i\in L}\neg c_i$

Fig. 3.

Activability of a transition system

Definition. A transition system $S = (Q, T, \{R_i\}_{i=1}^m)$ is said to be 'x'-*activable* from $P_1 \in \mathcal{P}$, iff T is 'x'-activable from P_1 in S.

Proposition 28. *Let S be a transition system and $P_1 \in \mathcal{P}$.*

(a) *S is p-activable from P_1 iff S is i-activable from P_1.*

(b) *S is o-activable from P_1 is a tautology.*

(c) *S is p-s-activable from P_1 iff S is i-s-activable from P_1.*
(d) *S is o-s-activable from P_1 is a tautology.*
(e) *S is p-qs-activable from P_1 iff S is i-qs-activable from P_1.*
(f) *S is o-qs-activable from P_1 is a contradiction.*

Proof. By direct application of Propositions 19, 21, 22 and 23 and by taking into account the relations: $\text{pre}^*(\bigvee c_i) = \bigvee c_i$, $\bigvee c_i \vee \widetilde{\text{pre}}(\bigvee c_i) = \top$, $(I \vee \text{pre} \wedge \widetilde{\text{pre}})^*(\bigvee c_i) = \bigvee c_i$.

Fig. 4 represents the main results of this paragraph.

Observation. The greatest invariant under which S is p-s-activable is equal to

$$\widetilde{\text{pre}}^{\times}\left(\text{pre}^*\left(\bigvee c_i\right)\right) = \widetilde{\text{pre}}^{\times}\left(\bigvee c_i\right) = \text{the greatest deadlock-free invariant.}$$

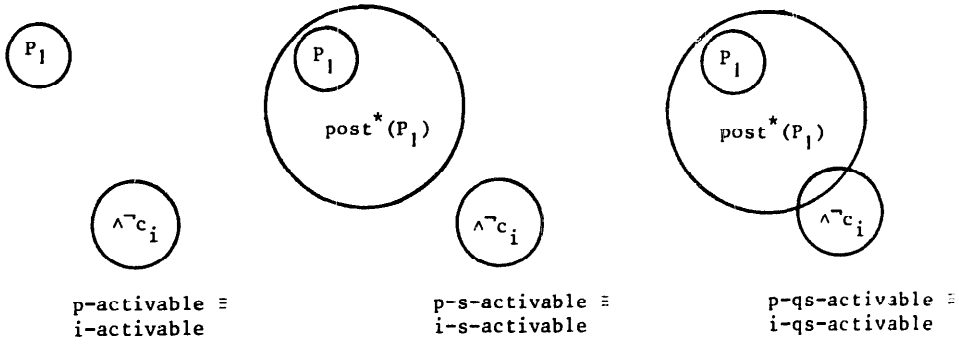


Fig. 4.

4. Applications

The results presented can be applied to the verification of a system described in any discrete model provided that a semantics of this model is given in terms of transition systems. In this case it is possible to compute the function pre associated to the given system and verifying a property amounts to computing fixed points of monotonic functions constructed from pre.

In order to illustrate this idea we consider a class of programs with guarded commands [8] for which the function pre can be obtained in a direct manner. These programs are of the type

$$S = \textbf{do } c_1 \to a_1 \,\square\, c_2 \to a_2 \,\square \cdots \square\, c_m \to a_m \textbf{ od}$$

where,
$-\{c_i\}_{i=1}^m$ is a set of total computable predicates,
$-\{a_i\}_{i=1}^m$ is a set of 'simultaneous' assignments, $a_i = (X = \alpha_i(X))$, where $X = (x_1, \ldots, x_n)$ is the vector of program variables and α_i an arbitrary total computable function.

If we assume that the state space of S is Q, then we can associate to S a transition system $S' = (Q, T, \{R_i\}_{i=1}^m)$ such that its transitions are in bijection with the guarded commands and $\forall q, q' \in Q$

$$((q, q') \in R_i \Leftrightarrow c_i(q) \textbf{ and } \alpha_i(q) = q').$$

Obviously, for a given predicate P, $\text{pre}[S](P) = \bigvee_{i=1}^m c_i \wedge P \circ \alpha_i$, where $P \circ \alpha_i$ represents the predicate $P \circ \alpha_i(q) = P(\alpha_i(q))$.

This class of programs, besides the nice possibilities for analysis that it provides, has good description capabilities due to the iterative non-deterministic construct **do od** [9]. In fact, for every iterative sequential program (deterministic or not), it is possible to find an equivalent program of this class by adding control variables (see for example [14]). Furthermore, if it is accepted, as in [20, 10, 31, 12], that concurrent execution can be 'represented' by non-deterministic sequential models, then these results can be applied to the verification of the properties of concurrent systems; in particular it is given in [10] a method for obtaining from a given parallel program with conditional critical regions [2] an 'equivalent' program of this class.

Finally, notice that under the aforementioned assumption, several models used to describe the flow of control in concurrent systems such as vector replacement systems, vector addition systems, Petri nets and their extensions, can be represented by such programs and consequently the methods given are directly applicable to these models [28].

Example 1. Consider the system described by:

$$\begin{aligned} S = \textbf{do } & x \geq 0 \rightarrow x := x + 1 \\ \square\ & x > 0 \rightarrow x := x - 1 \\ \square\ & x = 0 \rightarrow x := x - 1 \\ \square\ & x = -1 \rightarrow x := x + 1 \\ \textbf{od} & \end{aligned}$$

S may be considered to represent the coordination of a 'reader/writer' system: the domain of the synchronization variable x is Z; $x = 0$, $x > 0$, $x = -1$, correspond respectively to the situations where, the shared resource is free, x readers use the resource, a writer uses the resource.

(a) In order to find the greatest deadlock-free invariant, one has to compute iteratively:

$$P_{k+1} = P_k \wedge \widetilde{\text{pre}}(P_k), \quad \text{with } P_0 = \bigvee_{i=1}^4 c_i = x \geq -1$$

(We denote by $c_i \rightarrow a_i$ the ith guarded command of S, where $a_i = (x := \alpha_i(x))$)

$$\widetilde{\text{pre}}(P_0) = \bigwedge_{i=1}^4 (\neg c_i \vee P_0 \circ \alpha_i) = [x < 0 \vee x \geq 0] \wedge [x < -1 \vee x \geq -2] = \top.$$

Thus, $P_1 = P_0 = x \geq -1$ is the greatest deadlock-free invariant and S never blocks if it is initialized at a state verifying this invariant.

(b) Liveness of the guarded command $(x \geq 0 \rightarrow x := x + 1)$:

$$\text{pre}^*(x \geq 0) = \bigvee_{i=0}^{\infty} \text{pre}^i(x \geq 0) = x \geq -1.$$

The greatest invariant under which $(x \geq 0 \rightarrow x := x + 1)$ is live (p-s-activable) is $\widetilde{\text{pre}}^{\times}(x \geq -1) = x \geq -1$; thus, this guarded command is live from every initial state verifying $x \geq -1$.

(c) Since S does never block if it is initialized properly, in order to verify if there is a possible livelock for $(x \geq 0 \rightarrow x := x + 1)$ one has to compute the greatest non-terminating trajectory contained in $\neg(x \geq 0)$

$$P_{k+1} = P_k \wedge \text{pre}(P_k), \quad \text{with } P_0 = x < 0,$$

$$P_1 = (x < 0) \wedge [(x \geq 0) \wedge (x < 1) \vee (x \geq -1) \wedge (x < -1)] = \bot.$$

Thus, there is no possible livelock for this guarded command. On the contrary compute the greatest non-terminating trajectory contained in $\neg c_3 = (x \neq 0)$. We have

$$P_0 = (x \neq 0),$$

$$P_1 = (x \neq 0) \wedge [(x \geq 0) \wedge (x \neq 1) \vee (x \geq -1) \wedge (x \neq -1)] = x \geq 1,$$

$$P_2 = (x \geq 1) \wedge [(x \geq 0) \wedge (x \geq 2) \vee (x \geq -1) \wedge (x \geq 0)] = x \geq 1.$$

Thus, there is possibility of livelock for $c_3 \rightarrow a_3$.

Example 2. The following parallel program, given in [10], is a solution to the mutual exclusion problem discussed in [6] for two processes A and B.

```
var inA, inB: boolean initially false,
    prty: (A, B) initially A;

processA:while true do
 ⟨think⟩
 inA ← true;
 while inB do
  if prty = B then
   inA ← false;
   while prty = B do skip od;
   inA ← true
   fi
  od;
⟨critical section⟩
inA ← false;
prty ← B
od
```

```
processB:while true do
 ⟨think⟩
 inB ← true;
 while inA do
  if prty = A then
   inB ← false;
   while prty = A do skip od;
   inB ← true
   fi
  od;
⟨critical section⟩
inB ← false;
prty ← A
od
```

A 'non-deterministic version' of this program is [10]:

$p1 \leftarrow p2 \leftarrow 1$; inA $\leftarrow$ inB $\leftarrow$ false; prty $\leftarrow A$;

do $p1 = 1 \rightarrow$ inA $\leftarrow$ true; $p1 \leftarrow 2$
$\square\ p1 = 2 \wedge$ inB $\rightarrow p1 \leftarrow 3$
$\square\ p1 = 3 \wedge$ prty $= B \rightarrow$ inA $\leftarrow$ false; $p1 \leftarrow 4$
$\square\ p1 = 4 \wedge$ prty $= B \rightarrow$ skip
$\square\ p1 = 4 \wedge$ prty $\neq B \rightarrow$ inA $\leftarrow$ true; $p1 \leftarrow 2$
$\square\ p1 = 3 \wedge$ prty $\neq B \rightarrow p1 \leftarrow 2$
$\square\ p1 = 2 \wedge \neg$inB $\rightarrow$ ⟨critical section⟩; $p1 \leftarrow 5$
$\square\ p1 = 5 \rightarrow$ inA $\leftarrow$ false; $p1 \leftarrow 6$
$\square\ p1 = 6 \rightarrow$ prty $\leftarrow B$; $p1 \leftarrow 1$

$\square\ p2 = 1 \rightarrow$ inB $\leftarrow$ true; $p2 \leftarrow 2$
$\square\ p2 = 2 \wedge$ inA $\rightarrow p2 \leftarrow 3$
$\square\ p2 = 3 \wedge$ prty $= A \rightarrow$ inB $\leftarrow$ false; $p2 \leftarrow 4$
$\square\ p2 = 4 \wedge$ prty $= A \rightarrow$ skip
$\square\ p2 = 4 \wedge$ prty $\neq A \rightarrow$ inB $\leftarrow$ true; $p2 \leftarrow 2$
$\square\ p2 = 3 \wedge$ prty $\neq A \rightarrow p2 \leftarrow 2$
$\square\ p2 = 2 \wedge \neg$inA $\rightarrow$ ⟨critical section⟩; $p2 \leftarrow 5$
$\square\ p2 = 5 \rightarrow$ inB $\leftarrow$ false; $p2 \leftarrow 6$
$\square\ p2 = 6 \rightarrow$ prty $\leftarrow A$; $p2 \leftarrow 1$
od

Showing that the mutual exclusion constraint is respected in the concurrent system amounts to proving that the non-deterministic system never reaches a state for which the guards protecting the critical section are both true, i.e. that it satisfies the invariant property having as target predicate J_0:

$$J_0 = \neg[(p1 = 2) \wedge \neg\text{inB} \wedge (p2 = 2) \wedge \neg\text{inA}]$$

(J_0 is the negation of the conjunction of the guards protecting the critical section).

Computation of $J = \widetilde{\text{pre}}^{\times}(J_0)$:

$$J_{k+1} = J_k \wedge \widetilde{\text{pre}}(J_k), \quad \text{with } J_0 = p1 \neq 2 \vee \text{inB} \vee p2 \neq 2 \vee \text{inA}.$$

We have $J_1 = J_0 \wedge \widetilde{\text{pre}}(J_0) = J_0 \wedge \bigwedge_{i=1}^{18} [\neg c_i \vee J_0 \circ \alpha_i]$ where c_i and $a_i = (X \leftarrow \alpha_i(X))$ are respectively the ith guard and the ith command of the non-deterministic program (for example $c_{10} = (p_2 = 1)$).

The proposed solution being symmetric, in order to evaluate $\bigwedge_{i=1}^{18} (\neg c_i \vee J_0 \circ \alpha_i)$ it is sufficient to evaluate $\bigwedge_{i=1}^{9} (\neg c_i \vee J_0 \circ \alpha_i)$.

Remark that for every command a_i containing an assignment of the type $p1 \leftarrow j$ with $j \neq 2$, $J_0 \circ \alpha_i = \top$ because $J_0 = p1 \neq 2 \vee \text{inB} \vee p2 \neq 2 \vee \text{inA}$.

Thus, it remains to compute $\neg c_i \vee J_0 \circ \alpha_i$ for $i = 1, 4, 5, 6$.

- $i = 1$: $\neg c_1 \vee J_0 \circ \alpha_1 = p1 \neq 1 \vee (2 \neq 2) \vee \text{inB} \vee p2 \neq 2 \vee \text{inA} \vee \top = \top$;
- $i = 4$: $\neg c_4 \vee J_0 \circ \alpha_4 = \neg c_4 \vee J_0$;

$-i=5$: $\neg c_5 \vee J_0 \circ \alpha_5 = p1 \neq 4 \vee \text{prty} = B \vee \text{inB} \vee p2 \neq 2 \vee \top = \top$;
$-i=6$: $\neg c_6 \vee J_0 \circ \alpha_6 = p1 \neq 3 \vee \text{prty} = B \vee \text{inB} \vee p2 \neq 2 \vee \text{inA}$.

Let us put

$$K = p1 \neq 3 \vee \text{prty} = B \vee \text{inB} \vee p2 \neq 2 \vee \text{inA}$$

and

$$L = p2 \neq 3 \vee \text{prty} = A \vee \text{inA} \vee \text{p1} \neq 2 \vee \text{inB}.$$

We have $J_1 = J_0 \wedge \widetilde{\text{pre}}(J_0) = J_0 \wedge K \wedge L$. Compute now $J_2 = J_1 \wedge \widetilde{\text{pre}}(J_1) = J_0 \wedge \widetilde{\text{pre}}(J_0) \wedge \widetilde{\text{pre}}(K)\widetilde{\text{pre}}(L)$.

Computation of $\widetilde{\text{pre}}(K) = \bigwedge_{i=1}^{18} (\neg c_i \vee K \circ \alpha_i)$: Remark that for every i, $1 \leq i \leq 18$ such that a_i contains an assignment of one of the following types, $p1 \leftarrow j$ with $j \neq 3$, $\text{prty} \leftarrow B$, $\text{inB} \leftarrow \text{true}$, $p2 \leftarrow j$ with $j \neq 2$, $\text{inA} \leftarrow \text{true}$, we have $\neg c_i \vee K \circ \alpha_i = \top$. Furthermore, if $a_i = \text{skip}$, then $\neg c_i \vee K \circ \alpha_i = \neg c_i \vee K$ and this term is absorbed by $\widetilde{\text{pre}}(J_0)$. Thus, it remains to evaluate $\neg c_i \vee K \circ \alpha_i$ for $i = 2$ and $i = 15$.
$-i=2$: $\neg c_2 \vee K \circ \alpha_2 = p1 \neq 2 \vee \neg \text{inB} \vee \text{prty} = B \vee \text{inB} \vee p2 \neq 2 \vee \text{inA} = \top$;
$-i=15$: $\neg c_{15} \vee K \circ \alpha_{15} = p2 \neq 3 \vee \text{prty} = A \vee p1 \neq 3 \vee \text{prty} = B \vee \text{inB} \vee 2 \neq 2 \vee \text{inA} = \top$.

Thus, $\widetilde{\text{pre}}(K) = \neg c_4 \wedge \neg c_{13} \vee K$ and by symmetry $\widetilde{\text{pre}}(L) = \neg c_4 \wedge \neg c_{13} \vee L$. Consequently, the greatest invariant under which the mutual exclusion is respected is equal to

$$J = J_1 = J_0 \wedge K \wedge L,$$

$$J = \text{inA} \vee \text{inB} \vee (p1 \neq 2) \wedge (p2 \neq 2) \vee (p1 \neq 2) \wedge (p1 \neq 3)$$
$$\vee (p2 \neq 2) \wedge (p2 \neq 3) \vee (p2 \neq 2) \wedge (\text{prty} = A) \vee (p1 \neq 2) \wedge (\text{prty} = B).$$

This invariant is verified for the initial values of the variables and consequently the mutual exclusion constraint is respected by the two processes.

Example 3. Consider the problem of constructing a self-stabilizing ring of machines discussed in [7]. The first solution proposed in that paper can be described by the program:

$$S = \textbf{do } S_1 \,\square\, S_2 \,\square \cdots \square\, S_n \textbf{ od}$$

where,
$-S_0 = (x_0 = x_n \rightarrow x_0 := (x_0 + 1) \bmod n)$,
$-S_1 = (x_i \neq x_{i-1} \rightarrow x_i := x_{i-1})$, $1 \leq i \leq n$,
– each guarded command corresponds to a machine. The x_i's represent the state variables of the $n+1$ machines and the operations on the subscripts are done mod$(n+1)$. (We admit the existence of a 'central deamon' selecting one privilege at a time).

Let $S_i = c_i \rightarrow a_i$ with $a_i = (x_i := \alpha_i(X))$ the ith guarded command and represent by B_i the predicate

$$B_i = \neg c_0 \wedge \neg c_1 \wedge \cdots \wedge \neg c_{i-1} \wedge c_i \wedge \neg c_{i+1} \wedge \cdots \wedge \neg c_n.$$

Proving that this solution conforms to the specifications of the problem amounts to proving that

(a) $J = \bigvee_{i=1}^{n} B_i$ is a (deadlock-free) invariant, i.e. every possible successor of a legitimate state is a legitimate state;

(b) For every couple B_r, B_s, $0 \leq r, s \leq n$, B_r is a i-s-reachable from B_s;

(c) The system is self-stabilizing: from every possible initial state it will finally reach a legitimate state after execution of a finite number of transitions, i.e. J is i-reachable from $\neg J$.

We have

$$B_0 = (x_0 = x_1 = \cdots = x_{k-1} = x_k = \cdots = x_n),$$

$$B_k = (x_0 = x_1 = \cdots = x_{k-1} \neq x_k = \cdots = x_n), \quad 1 \leq k \leq n.$$

Let's prove that $B_i \sqsubseteq (\text{pre}[S_i] \wedge \widetilde{\text{pre}}[S_i])(B_{i+1})$, i.e. every time S_i is executed from a legitimate state, a state verifying B_{i+1} is reached. This is equivalent to

$$B_i \sqsubseteq (c_i \wedge B_{i+1} \circ \alpha_i) \wedge (\neg c_i \vee B_{i+1} \circ \alpha_i) = c_i \wedge B_{i+1} \circ \alpha_i.$$

By substituting the B_i's in this inequality one obtains the trivially verified relations.

For $i = 0$:

$$(x_0 = x_1 = x_2 = \cdots = x_n) \sqsubseteq (x_0 = x_1 = x_2 = \cdots = x_n).$$

For $i = k$, $1 \leq k \leq n$:

$$(x_0 = x_1 = \cdots = x_{k-1} \neq x_k = \cdots = x_n) \sqsubseteq (x_{k-1} \neq x_k) \wedge [x_0 = \cdots = x_{k-1} \neq x_{k+1} = \cdots = x_n]$$

The proved relation $B_i \sqsubseteq (\text{pre}[S_i] \wedge \widetilde{\text{pre}}[S_i])(B_{i+1})$ implies that $B_i \sqsubseteq (\text{pre}[S] \wedge \widetilde{\text{pre}}[S])(B_{i+1})$. By taking the disjunction of all the relations of this type we have:

$$\bigvee_{i=0}^{n} B_i \sqsubseteq \bigvee_{i=0}^{n} (\widetilde{\text{pre}}[S] \wedge \text{pre}[S])(B_i)$$

and since $\text{pre}[S] \wedge \widetilde{\text{pre}}[S]$ is a monotonic function,

$$\bigvee_{i=0}^{n} B_i \sqsubseteq (\widetilde{\text{pre}}[S] \wedge \text{pre}[S])\left(\bigvee_{i=0}^{n} B_i\right).$$

The latest relation shows that J is a deadlock-free invariant. Furthermore, the relations $B_i \sqsubseteq (\text{pre}[S] \wedge \widetilde{\text{pre}}[S])(B_{i+1})$ for $0 \leq i \leq n$ imply that if the system is initialized at a state verifying B_i, then it goes through a sequence of states verifying successively $B_{i+1}, B_{i+2}, \ldots, B_0, B_1, \ldots$ and it reaches a state verifying B_i after $n+1$ transitions. Thus, B_i is i-s-reachable from every B_j, $1 \leq j \leq n$.

Finally, in order to establish (c) one has to prove that the greatest trajectory contained in $\neg J$ is equal to $\perp$. The computation of $(I \wedge (\text{pre}[S] \vee \widetilde{\text{pre}}[S])^{\times}(\neg J)$ raises non-trivial problems of manipulation and simplification of predicates.

5. Conclusion

This paper proposes a very general framework for tackling the problem of system verification. The presented results can be applied to any discrete model provided that a semantics of this model can be given in terms of transition systems.

The method used for the study of properties, namely their definition by giving a 'target' predicate and a reachability type, seems to be sufficiently general for being applicable to a great variety of cases. This method is the more interesting as it allows a systematic study of the properties in terms of two fundamental concepts, the different reachability types being expressed by simple relations involving invariants and trajectories.

Computing fixed points of monotonic functions is, from a practical point of view, the central problem and it determines the limitations of our approach in the domain of verification. Apart from the limitations of theoretical nature (non-decidability of the 'interesting' system properties, non-continuity of the functions) serious problems appear when applying iterative solution methods which require the manipulation, simplification and comparison of predicates on several variables.

Superposed on these difficulties is the lack both of any general criterion guaranteeing the convergence of the iterations and of any notion allowing to measure the 'distance' between the result of the ith iteration and the approached fixed point.

For all these reasons, it is not realistic to expect that the presented results can be applied *directly* to the analysis of systems of non-trivial complexity. However, we believe that it is possible to obtain mechanizable proof methods by applying techniques for approximating fixed points as in [3–5] or by working with finite state models which represent some adequately chosen 'abstraction' of a complex system under study.

How to exploit in practice the given theoretical results is an open problem to which we are not supposed to answer; the examples with 'condition-action' systems in Section 4 are an illustration of what can be done with other models too. The main contribution of this paper is to propose a methodology for system verification by giving a unified approach for generating, comparing and proving system properties.

Acknowledgment

I am indebted to Pedro Guerreiro for interesting discussions and critiques. Also, thanks go to Willem P. de Roever and an anonymous referee whose suggestions have influenced and improved this work.

References

[1] J.W. de Bakker, Semantics and termination of non-deterministic recursive programs, *3rd International Colloquium Automata Languages and Programming* (1976) 435–477.

[2] P. Brinch Hansen, *Operating Systems Principles* (Prentice-Hall, Englewood Cliffs, NJ, 1973).
[3] E.M. Clarke Jr., Synthesis of resource invariants for concurrent programs, *ACM Trans. Progr. Languages and Systems* **2** (3) (1980) 338–358.
[4] P. Cousot and N. Halbwachs, Automatic discovery of linear restraints among variables of a program, *Proc. 5th ACM Symposium on Principles of Programming Languages*, Tucson, AZ (1978) 84–96.
[5] P. Cousot and R. Cousot, Systematic design of program analysis frameworks, *Proc. 6th ACM Symposium on Principles of Programming Languages*, San Antonio, TX (1979) 269–282.
[6] E.W. Dijkstra, Solution of a problem in concurrent programming control, *Comm. ACM* **8** (9) (1965) 569.
[7] E.W. Dijkstra, Self stabilizing system in spite of distributed control, *Comm. ACM* **17** (11) (1974) 643–644.
[8] E.W. Dijkstra, Guarded commands, non determinacy and formal derivation of programs, *Comm. ACM* **18** (8) (1975) 453–457.
[9] E.W. Dijkstra, *A Discipline of Programming* (Prentice-Hall, Englewood Cliffs, NJ, 1976).
[10] L. Flon and N. Suzuki, Non determinism and the correctness of parallel programs, in: E.J. Neuhold, Ed., *Formal description of programming concepts* (North-Holland, Amsterdam, 1978) 589–608.
[11] P. Guerreiro, A relational model for non-deterministic programs and predicate transformers, Lecture Notes in Computer Science **83** (Springer, Berlin, 1980) 136–146.
[12] P. Guerreiro, Relational semantics of strongly communicating sequential processes, Research Report No. 200, IMAG, Grenoble (1980).
[13] M. Hack, Analysis of production schemata by Petri nets, Project MAC, M.I.T. (1972).
[14] D. Harel, On folk theorems, *Comm. ACM* **23** (7) (1980) 379–389.
[15] C.A.R. Hoare, Some properties of predicate transformers, *JACM* **25** (3) (1978) 461–480.
[16] R.M. Keller, Vector replacement systems: a formalism for modeling asynchronous systems, Princeton University, Technical Report No. 117 (1972).
[17] R.M. Keller, Formal verification of parallel programs, *Comm. ACM* **19** (7) (1976) 371–384.
[18] Y.S. Kwong, On reduction of asynchronous systems, *Theoret. Comput. Sci.* **5** (1977) 25–50.
[19] Y.S. Kwong, On the absence of livelocks in parallel programs, in: *Semantics of Concurrent computation*, Lecture Notes in Computer Science **70** (Springer, Berlin, 1979)
[20] A. van Lamsweerde and M. Sintzoff, Formal derivation of strongly correct parallel programs, MBLE Research Lab., Report R338 (1976); and *Acta Informat.* **12** (1) (1979) 1–31.
[21] L. Lamport, 'Sometime' is sometimes 'not never' — On the temporal logic of programs, *Proc. 7th Annual ACM Symposium on Principles of Programming Languages*, Las Vegas (1980) 174–185.
[22] K. Lautenbach, Liveness in Petri nets, GMD Internal report, ISF 75-02-1, Bonn (1975).
[23] Z. Manna and A. Pnueli, The modal logic of programs, Lecture Notes in Computer Science **71** (Springer, Berlin, 1979) 385–406.
[24] A. Mazurkiewicz, Proving properties of processes, *Algorytmy* **XI** (19) (1974) 5–22.
[25] D. Park, Fixpoint induction and proofs of program properties, in: *Machine Intelligence 5* (1969) 59–78.
[26] J.L. Peterson, Petri-nets, *Comput. Surveys ACM* **9** (3) (1977) 223–252.
[27] A. Pnueli, The temporal semantics of concurrent programs, Lecture Notes in Computer Science **70** (Springer, Berlin, 1979) 1–20.
[28] J.P. Queille and J. Sifakis, Iterative methods for the analysis of Petri nets, First European Workshop on Applications and Theory of Petri Nets, Strasbourg (1980).
[29] W.P. de Roever, Dijkstra's predicate transformer, non-determinism, recursion and termination, in: *MFCS '76*, Lecture Notes in Computer Science **45** (Springer, Berlin, 1976) 472–481.
[30] B.K. Rosen, Correctness of parallel programs: the Church–Rosser approach, *Theoret. Comput. Sci.* **2** (1976) 183–207.
[31] J. Sifakis, Le contrôle des systèmes asynchrones: concepts, propriétés, analyse statique, Thèse d'Etat, Université de Grenoble (1979).
[32] J. Sifakis, Deadlocks and livelocks in transition systems, Lecture Notes in Computer Science **88** (Springer, Berlin, 1980) 587–600.
[33] A. Tarski, On the calculus of relations, *J. Symbolic Logic* **6** (3) (1941) 73–89.
[34] A. Tarski, A lattice-theoretical fixpoint theorem and its applications, *Pacific J. Math.* **5** (1955) 285–309.