INFORMATION AND COMPUTATION 89, 144–179 (1990)

Reduction and Covering of Infinite Reachability Trees*

ALAIN FINKEL

École Normale Supérieure de Cachan, Laboratoire d'Informatique, 61, avenue du Président Wilson, 94235 Cachan, France

We present a structure for transition systems with which the main decidability results on Petri nets can be generalized to structured transition systems. We define the reduced reachability tree of a structured transition system; it allows one to decide the finite reachability tree problem (also called the finite termination problem) and the finite reachability set problem. A general definition of the coverability set is given and the procedure of Karp and Miller is extended for well-structured transition systems. We show then that the coverability problem is a decidable problem in the framework of well-structured transition systems. Finally, we introduce structured set of terminal states and we show that the finite reachability tree problem and the finite reachability set problem are decidable. Coverability is an open problem for structured transition systems with a structured set of terminal states. © 1990 Academic Press, Inc.

CONTENT

1. INTRODUCTION

The concepts of states and state transitions are fundamental to many formalisms. State transition systems (or simply, transition systems) form a general model for specification and verification of many properties of

* A short and partial version of this paper was presented at ICALP'87 under the title "A Generalization of the Procedure of Karp and Miller to Well Structured Transition Systems" (Finkel, 1987). This paper was completed during a visiting year at the University of Montréal and at the Research Center in Computer Science of Montréal.

0890-5401/90 $3.00

systems. When the reachability set (i.e., the set of all the reachable states) is finite we can, at least theoretically, verify the traditional properties, such as existence of an infinite sequence, deadlock freedom, and mutual exclusion. But if the reachability set is infinite, verification of these properties with the study of the infinite reachability graph becomes impossible. Let us remark that there exist some transition systems, for example, the ones associated with **Petri nets** (Brauer *et al.*, 1986), with certain **Fifo nets** (Memmi and Finkel, 1985; Finkel, 1986; Choquet and Finkel, 1987), or with certain systems of Communicating Finite State Machines (*CFSMs*) (Brand and Zafiropulo, 1983; Pachl, 1986; Rosier and Yen, 1986) for which an infinite set of states does not preclude the analysis of typical properties.

In these models, the analysis is often made by associating with the infinite set of reachable states a finite set of states and "limits of states." This finite set (we called in the **coverability set**) allows the verification of usual properties. In the framework of Petri nets or equivalently **Vector Addition Systems**, the generation of the coverability set is achieved with the help of the **coverability tree** (Karp and Miller, 1969). The aim of this work is to illustrate and to generalize the fundamental concepts used in the construction of the coverability set of a Petri net. We define the general **structure** allowing the analysis of a transition system by reducing the number of states thanks to a coverability set.

One of the main properties of Petri nets is the existence of an **ordering** $\leqslant$ on the reachability set. This ordering gives a property of **monotonicity** to the net. Monotonicity of a net means that if from a state s, we can reach a state s_1, then from every state s' greater than or equal to s, we can reach a state s'_1 greater than or equal to s_1 (see, for example, Brams, 1983). A second property of Petri nets is that the reachability tree has a **finite degree** (because from every state, one may only reach a finite number of different states by firing one transition). This enables us to apply Koenig's Lemma. The fact that the ordering $\leqslant$ is a **well ordering** (Dickson, 1913; Kruskal, 1972) is the third important property of Petri nets. Recall that a well ordering is an ordering such that from every infinite sequence, one can extract an infinite increasing subsequence. For example, the usual ordering on the set of integers is a well ordering (Dickson, 1913). Finally, the ordering $\leqslant$ and the equality are **decidable** for states. By decidable we mean that for two vectors s, s', one can decide whether or not $s \leqslant s'$ and $s = s'$ (for orderings, the decidability of the equality is a consequence of the decidability of $\leqslant$; but it is not true for general quasi-orderings (a quasi-ordering is a reflexive and transitive relation)).

These four properties allow us to decide, for example, the finite reachability tree problem for Petri nets. The algorithm presented in (Karp and Miller, 1969) consists of constructing the reachability tree until we meet two comparable states s, s' on the same branch so that s' is reachable from s and $s \leqslant s'$; we stop if $s = s'$ or we continue the reachability tree during a finite delay if $s < s'$. As the reachability tree has a finite degree and as the ordering $\leqslant$ is a decidable well ordering, it follows that the reachability tree is infinite if and only if one eventually reaches two states s, s' such that s' is reachable from s and $s \leqslant s'$. This last property is decidable. In conclusion, we can decide the finite reachability tree problem for Petri nets.

The proof that the finiteness of the reachability set is a decidable problem for Petri nets is based on the following property (the strict monotonicity): if we can reach a state s_1 from a state s, then from any state $s' > s$ we can reach a state $s'_1 > s_1$. To prove the decidability of the finiteness of the reachability set, we apply the same reasoning as before; we find that the reachability set is infinite if and only if there exists an infinite strictly increasing sequence of reachable states. This last property is decidable, which implies that the finite reachability set problem is a decidable problem.

Let us consider now any transition system TS. We say that TS is **structured** when there exists a quasi-ordering $\leqslant$ on the reachability set so that $(TS, \leqslant)$ has the monotonicity property, $\leqslant$ is decidable, $\leqslant$ is a well quasi-ordering, and the reachability tree has a finite degree. We define two types of structured transition systems according to the monotonicity induced by the quasi-ordering. We show that for the most general structured transition systems, the finite reachability tree problem is a decidable problem. Finiteness of the reachability set is also a decidable problem except for the first type of structured transition systems.

The coverability problem (Given a state s, is it possible to reach a state $s' \geqslant s$?) arises naturally in the framework of structured transition systems; let us remark that the quasi-liveness problem for Petri nets (Karp and Miller, 1969) is reducible to the coverability problem. The method of solving this problem in the framework of Petri nets is by the use of a coverability tree. This finite tree "covers" (for the usual quasi-ordering on integers) all the states of the reachability set. Now, what allows us to construct a finite coverability tree?

First, Petri nets are strictly structured for the usual ordering on integers. Second, this ordering, naturally extended on limits of sequences of states, is still a well quasi-ordering. Third, we know how to compute the limit of an infinite increasing sequence of states, which means that for every integer n, we may compute the nth term of the infinite sequence; moreover, from every limit of sequence of states, we can decide whether or not the system is blocked. Fourth, the quasi-ordering $\leqslant$ and the equality are both decidable on the limits of sequences of states. Finally, there exists an integer k (k is the number of places of the Petri net) such that the kth "limit" of the reachability set is finite.

We extend the procedure of Karp and Miller to a class of structured labelled transition systems, called **well-structured labelled transition systems**. For these systems, we show that the coverability problem is decidable.

Sometimes we need to refine the behaviour of a transition system in considering a set of terminal states. If we look at a general set of terminal states, we increase the power of the Petri net model to that of a Turing machine (Hopcroft and Ullman, 1979). Finite sets of terminal states have been studied for Petri nets (Hack, 1976; Peterson, 1981); here, we consider infinite sets of terminal states which possess a structure allowing us to resolve some of our problems. Structured sets of terminal markings of a Petri net were introduced for the first time in (Choquet and Finkel, 1987). Here, we generalize this notion to general structured transition systems having a **structured set of terminal states**.

We decide the finite reachability tree and the finite reachability set problems (with respect to the set of terminal states) using another finite reduced reachability tree. We also extend the procedure of Karp and Miller to well-structured transition systems having a structured set of terminal states. However, Karp and Miller's tree is not a coverability tree and hence it does not allow us to decide the coverability set problem.

2. Transition systems

2.1. *Preliminaries*

If X and Y are two sets, we denote by $X \cup Y$ the union of X and Y and by $X \cap Y$ the intersection of X and Y. We denote by $\mathbf{N}$ the set of positive integers and by $\mathbf{N} \cup \{\omega\}$ the classical completion of $\mathbf{N}$. Let X be an *alphabet* (i.e., a finite set) whose elements are called *letters*. The concatenation operator "$\cdot$" allows us to construct words on X. A *word* x on X is a sequence of letters of X. The *empty word* is denoted by λ. A *language* L is a set of words. X^* is the set of finite words on X (X^* contains the empty word) and X^+ is equal to X^*-$\{\lambda\}$; χ^ω is the set of infinite words on X and $X^\infty = X^* \cup X^\omega$. We write $|x|$ for the length of x. We have $|x \cdot x'| = |x| + |x'|$, for every pair of words $x, x' \in X^*$, and $|\lambda| = 0$. We also use this notation for the cardinality $|A|$ of the set A. A *morphism* m is a function from X^* into Y^* such that $m(x \cdot x') = m(x) \cdot m(x')$ for any two words x, x', and moreover we have $m(\lambda) = \lambda$ and hence a morphism only has to be described with respect to X. A word x is a *left factor* of a word y if there exists a word z such that $y = x \cdot z$. We write $x[n]$ for the left factor of the word x whose length is equal to n. Given a language $L \subseteq X^*$ and a sub-alphabet $Y \subseteq X$, the *projection* of L on Y^* (or Y) is written $\mathrm{proj}_Y(L)$ and it is defined by the following morphism: $\mathrm{proj}_Y(t) = \text{if } t \in Y \text{ then } t \text{ else } \lambda$. Let

$\leqslant$ be a *quasi-ordering* (a reflexive and transitive relation) on a set S and let A and B be two subsets of S. We define a transitive relation $\leqslant_s$ from $\leqslant$ between subsets of S in the following way: $A \leqslant_s B$ (B is a *cover* of A) iff for every element $a \in A$, there exists an element $b \in B$ such that $a \leqslant b$; $\leqslant_s$ is not reflexive (consider, for example, $\mathbf{N}$ and $\mathbf{N}$-$\{25\}$). We sometimes denote $\leqslant_s$ by $\leqslant$. One can associate two non-reflexive and transitive relations defined by: $s < s'$ iff $s \leqslant s'$ and $s \neq s'$; $A < B$ (B is a *strict cover* of A) iff for every element $a \in A$, there exists an element $b \in B$ such that $a < b$. We sometimes denote $<_s$ by $<$. A quasi-ordering $\leqslant$ on a set S is a *well quasi-ordering* if from every infinite sequence $\{s_n\}$ of elements $s_n \in S$, one can always extract an infinite increasing (for $\leqslant$) subsequence $\{s_{ni}\}$. An ordered set is *directed* when it contains a least element and every increasing sequence has a upper bound. A function f: $A \rightarrow B$, where A and B are two directed ordered sets is *continuous* when it commutes with the upper bounds of increasing sequences; i.e., $f(\lim a_n) = \lim f(a_n)$, where $\{a_n\}$ is an increasing sequence of elements of A.

We denote $|$(read "*is a subword of*") the quasi-ordering (Higman, 1952) on finite words defined as follows: for two words u and v of A^*, $u \mid v$ if the word v can be written as $w_1 \cdot u_1 \cdot w_2 \cdot u_2 \cdots w_n \cdot u_n \cdot w_{n+1}$, where $u = u_1 \cdot u_2 \cdots u_n$ and $w_1, w_2, ..., w_{n+1}$ are words of A^*. A *tree* is a connected acyclic labelled directed graph such there exists a unique node r, called the root, from which every node is reachable. A tree T has a *finite degree* or is *finitely branching* if every node has only a finite number of direct successors; when T is infinite and is finitely branching, T contains at least an infinite branch (Koenig, 1936). We denote by *Labels*(T) the set of labels of nodes of T.

2.2. *Reachability Properties for Transition Systems*

Communication protocols are often modelled by systems of finite state machines communicating by Fifo channels (Bochmann, 1978; and Zafiropulo, 1983). In a system of *CFSMs*, the finite state machines (or finite transition systems) communicate exclusively by exchanging messages via connecting channels. There are generally two one-directional Fifo channels between each pair of machines in the system. Each state transition rule is accompanied by either sending (denoted by $-a$) or receiving (denoted by $+a$) one message to or from one of the output or the input channels of the machine (a basic extension of this model allows a machine to send and to receive a word instead of only one message). Consider the protocol described in (Finkel, 1988) as a system of two *CFSMs* (Fig. 2.1). The two machines are M_1 and M_2. Circles represent the state of the machines. The transition, from state 1 to state 2, labelled "$-ab$" (in machine M_1) indicates that the transition is accompanied by sending the word "*ab*" to the output channel of the machine. (Channel destinations are not explicitly

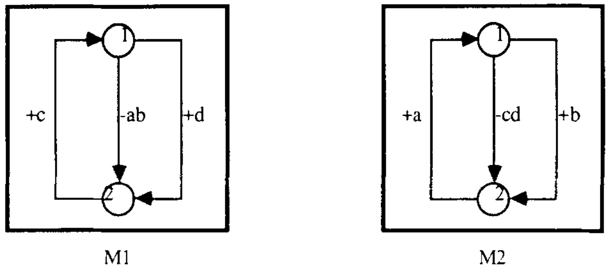


FIG. 2.1. Protocol $P1$.

given here as there is only a single input and output channel for each machine.) The label $+c$ (in machine M_1) indicates that the message "c" is to be received. The starting state for M_1 (and M_2) is the state labelled 1.

We want to know if this protocol contains a deadlock, if it has an infinite computation, if there is a bound on the lengths of the words which the channel can contain in any computation, and many other questions. To be able to automatically answer these questions, we introduce the general model of transition systems. We give a definition of a transition system which is equivalent to those in (Keller, 1972, 1976). Let us remark that the following definition does not contain any reference to a labelling function of transitions as is generally the case.

DEFINITION 2.1. A **transition system** TS is a triplet $TS = \langle S, R, S_0 \rangle$, where S is the set of states, R is the transition relation on S (R is a binary relation included in $S \times S$), and S_0 is a finite subset of S whose elements are called initial states. A transition system is **finite** if and only if S is finite; a finite transition system is often called a finite state machine or a finite automaton.

EXAMPLE 2.2. The following transition system $TS1 = \langle S1, R1, s_{01} \rangle$,

$$S_1 = \{(i, j, w_1, w_2) : i, j = 1, 2 \text{ and } w_1 \in \{a, b\}^*, w_2 \in \{c, d\}^*\},$$

$$\begin{aligned} R1 = \{&((1, j, w_1, w_2), (2, j, w_1 ab, w_2)), ((1, j, w_1, dw_2), (2, j, w_1, w_2)),\\ &((2, j, w_1, cw_2), (1, j, w_1, w_2)), ((i, 1, w_1, w_2),\\ &(i, 2, w_1, w_2 cd)), ((i, 1, bw_1, w_2),\\ &(i, 2, w_1, w_2)), ((i, 2, aw_1, w_2), (i, 1, w_1, w_2)) : w_1, w_2 \in A^*\},\end{aligned}$$

$$s_{01} = (1, 1, -, -),$$

is naturally associated to the protocol described in Fig. 2.1. In the following, we often confuse a protocol described by a system of (two) finite state machines with its associated transition system.

Sometimes we want to describe the behaviour of a system. A way to achieve this is to label the elements of R (the transitions) of a transition system.

DEFINITION 2.3. A **labelled transition system** is a couple $LTS = \langle TS, L\rangle$, where TS is a transition system $TS = \langle S, R, S_0\rangle$ and L is a labelling morphism from R into A (A is a finite alphabet of actions).

We may represent a labelled transition system $LTS = \langle S, R, S_0, L\rangle$ by the following notation: $\langle S, A, h, S_0\rangle$, where h is the partial transition function from $S \times A$ into 2^S (2^S is the set of all subsets of S) defined by $s' \in h(s, a)$ iff $(s, s') \in R$ (also written $s' \in R(s)$) and $L(s, s') = a$. To simplify the notation, we denote $L((s, s'))$ by $L(s, s')$. The labelling allows non-deterministic "machines." As in (Kasai and Miller, 1982), the partial function h can be changed into a total function h_0; we add a new element to S, denoted by $\perp$. We suppose now that S contains $\perp$. We define $h_0: S \times A \to 2^S$ in the following way: $h_0(s, a) =$ **if** $h(s, a)$ is defined **then** $h(s, a)$ **else** $\perp$. We extend, in a natural way, the function h_0 to a morphism $h_{0m}: 2^S \times A^+ \to 2^S$ such that:

for $S' \subseteq S$, $a \in A$, $h(S', a)$ is the set of states defined by $h(S', a) = \bigcup_{s' \in S'} h(s', a)$ and, the function h is naturally extended to words on A^+,

for $S' \subseteq S$, $x \in A^+$, and $a \in A$, we put $h(S', xa) = h(h(S', x), a)$.

We abuse the notation by denoting h_{0m} by h.

Let us remark that all the results presented in this paper for general transition systems (i.e., non-labelled transition systems), in Section 3, are obviously still true for labelled transition systems.

DEFINITION 2.4. Given a transition system TS, we define the sets $R^n(TS, s)$, shortly $R^n(s)$, where s is a state and n is an integer, by the following: $R^0(s) = \{s\}$, $R^1(s) = R(s) = \{s' \in S : (s, s') \in R\}$, the relation R is extended on sets of states by $R(\{s, s'\}) = R(s) \cup R(s')$ and $R^{n+1}(s) = R(R^n(s))$. The transitive closure R^* of R is defined by $R^*(s) = R^0(s) \cup R^1(s) \cup \cdots \cup R^n(s) \cup \cdots$. A state s is a **deadlock state** (or simply a **deadlock**) when $R(s) = \varnothing$. The **reachability set** of a transition system TS is denoted by $RS(TS)$ and it is equal to $R^*(S_0)$. The set $R^+(s)$ is defined by $R^+(s) = R^1(s) \cup \cdots \cup R^n(s) \cup \cdots$.

Remark 2.5. The set of states S always contains the reachability set but it can be greater than it.

In the following, we state **two assumptions** about transitions systems: Given a transition system $TS = \langle S, R, S_0 \rangle$,

A.1. For every state $s \in S$, the set $R(s)$ is finite and it is effectively computable.

A.2. S_0 consists of a single element: $S_0 = \{s_0\}$ and $s_0 \in S$.

Remark 2.6. Assumption A.1 is a classical restriction: the reachability tree then has a finite degree, hence it allows us to apply Koenig's Lemma; moreover, the reachability tree can be effectively constructed. Assumption A.2 simplifies the notation but it is not a real restriction because every transition system with a finite number of initial states is equivalent to another transition system having a unique initial state.

For labelled transition systems, we add a third condition to be able to only consider functions h into S (instead of 2^S).

A.3. The function h maps from $S \times A$ into S.

DEFINITION 2.7. The **reachability tree** of a transition system $TS = \langle S, R, s_0 \rangle$ is denoted by $RT(TS)$ and it is a rooted tree defined in the following way.

(1) The root r is labelled by the initial state s_0.

(2) A node m, labelled by s, $s \in S$, has no successor if and only if $R(s) = \varnothing$ (s is a deadlock).

(3) If m is a node labelled by $s \in S$, which does not satisfy condition (2) then for every state $s' \in R(s)$, there exists a node m' in $RT(TS)$, the successor of m, labelled by s'.

The **Reachability Graph**, $RG(TS)$, is obtained from the reachability tree by identifying nodes which have the same label.

EXAMPLE 2.8. The reachability graph of $TS1$ is described in Fig. 2.2. (One may note that, here, all arcs are labelled by the message sent or received.)

Remark 2.9. A transition system can be represented by its reachability tree in the following way: the relation R is represented by the arcs between states in the reachability tree, the initial state is the root of the tree, and the set of states S is equal to the set (or only contains the set) of labels of all nodes of the reachability graph. A labelled reachability graph may represent a labelled transition system (Fig. 2.2). In this case, the language of a labelled transition system is defined by $L(LTS) = \{x \in A^*: h(s_0, x) \neq \perp\}$. The reachability graph, sometimes, gives a finite representation of an infinite reachability tree and always gives a more concise

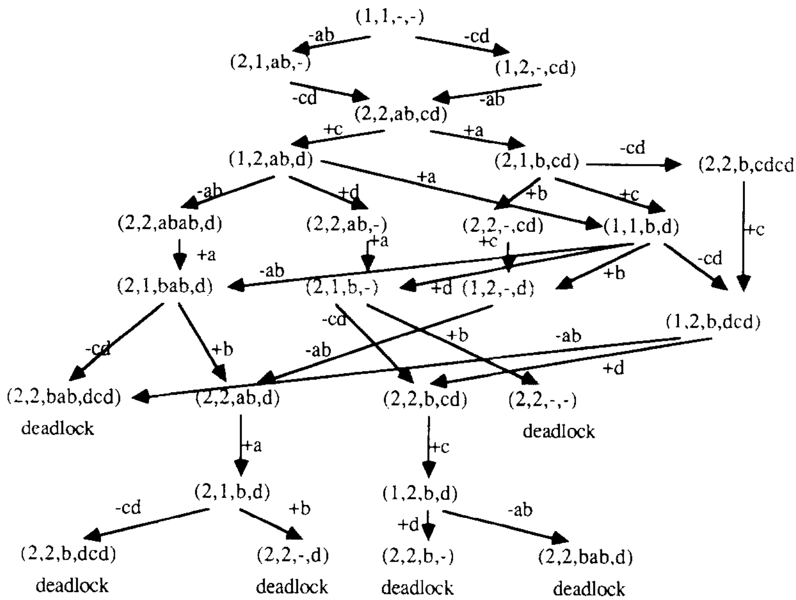


FIG. 2.2. The reachability graph of $TS1$.

representation. In what follows we often confuse a node and its label when there is no ambiguity.

The analysis of a transition system consists of verifying some properties of the system; for example, we may want to know if the state $(2, 1, b, d)$ in Fig. 2.1 is reachable from the initial state and if it is a deadlock. We are interested in the following reachability problems for transition systems $TS = \langle S, R, s_0 \rangle$.

1. The **Finite Reachability Tree Problem (FRTP)**: Is the reachability tree $RT(TS)$ finite?

2. The **Finite Reachability Set Problem (FRSP)**: Is the reachability set $RS(TS)$ finite?

3. The **Coverability Problem (CP)**: Suppose there exists a quasi-ordering $\leqslant$ on the set of states S which has "good" properties like "monotonicity" (it means, if $s' \geqslant s$ then $R^*(s) \subseteq R^*(s')$; a formal definition of monotonicity is given in Section 3) and "well quasi-ordering." Given a state $s \in S$, is there a reachable state $s' \in RS(TS)$ such that $s' \geqslant s$?

4. The **Reachability Problem (RP)**: Given a state $s \in S$, does s belong to the reachability set $RS(TS)$?

5. The **Reachability Sets Inclusion Problem (RSIP)**: Given two states

$s, s' \in S$, is the reachability set from s included in the reachability set from s': $R^*(s) \subseteq R^*(s')$?

6. The **Deadlock Problem (DP)**: Is there a deadlock state?

Analysis of Protocol P1. The reachability tree is finite and the reachability set contains 24 elements. The *CP*, the *RP*, the *RSIP*, and the *DP* are all decidable because the reachability set is finite. The two Fifo channels are bounded by 4 and the global bound (the maximal sum of the lengths of the two words in the channels) for the two channels is equal to 6. There is no infinite behaviour because there is no circuit in the reachability graph.

Remark 2.10. The Finite Reachability Tree Problem is equivalent to the Finite Termination Problem often studied in rewritten systems. The Finite Reachability Set Problem is equivalent to the Boundedness Problem in the framework of Petri nets or Vector Addition Systems (Karp and Miller, 1969), Fifo nets (Memmi and Finkel, 1985; Brauer *et al.*, 1986; Finkel, 1986), and systems of finite state machines communicating by Fifo channels (Brand and Zafiropulo, 1983; Rosier and Yen, 1986). In general, the Finite Reachability Set Problem is more complex than the Finite Reachability Tree Problem (Rosier and Yen, 1986). The Quasi-Liveness Problem (Karp and Miller, 1969) is reducible to the Coverability Set Problem; the Coverability Set Problem is called the Exceedability Problem in (Kasai and Miller, 1982). The Reachability Problem was shown to be decidable for Petri nets and Vector Addition Systems by (Mayr, 1984). The Reachability Sets Inclusion Problem is undecidable for Petri nets (Hack, 1976), hence it is also undecidable for transition systems. The Deadlock Problem is reducible to the Reachability Problem for Petri nets (Mayr, 1984; Valk and Jantzen, 1985), but in general, this problem is more complex than the Reachability Problem, hence it is often undecidable (Brand and Zafiropulo, 1983).

3. Decidability of Two Reachability Problems

In the general case (when the reachability tree is not known to be finite) no algorithm exists which can verify traditional properties using the (infinite) reachability tree. But some of these algorithms exist in the framework of particular transition systems with infinite reachability trees and/or infinite reachability sets. For example:

— Petri nets and Coloured Petri nets (Hack, 1975; Jensen, 1981; Brams, 1983; Valk and Jantzen, 1985; Brauer et al., 1986).

— Monogeneous Communicating Finite State Machines and monogeneous Fifo nets [Memmi and Finkel, 1985; Finkel, 1986, 1988).

— Free choice Fifo nets (Finkel, 1986, 1988; Choquet, 1987).

— Linear Communicating Finite State Machines and linear Fifo nets (Choquet and Finkel, 1987; Gouda *et al.*, 1987).

— Communicating Finite State Machines having the recognizable channel property (Pachl, 1986).

One of the nice properties of Petri nets (and in fact of almost all of the previous models), which allows us to decide the Finite Reachability Tree Problem and the Finite Reachability Set Problem, is that the usual ordering $\leqslant$ on vectors of integers (then on the reachability set) has the monotonicity property. This means that for every state s_1 reachable from a state s, if $s' \geqslant s$ then from s', one can reach at least one state $s'_1 \geqslant s_1$. This notion of monotonicity has never been given in general, but only in the particular case of Petri nets (for example, in Brams, 1983). We propose to state now the general notion of monotonous transition systems.

DEFINITION 3.1. A **quasi-ordered transition system** is a couple $(TS, \leqslant)$, where $TS = \langle S, R, s_0 \rangle$ is a transition system and $\leqslant$ is a quasi-ordering on the reachability set. A **well-quasi-ordered transition system** is a quasi-ordered transition system such that $\leqslant$ is a well quasi-ordering on the reachability set.

Let us remark that if the quasi-ordering $\leqslant$ is a well quasi-ordering on X then for every subset $X' \subseteq X$, the quasi-ordering $\leqslant$ is still a well quasi-ordering on X': as the reachability set $RS(TS)$ is always included in S, every well quasi-ordering on S is still a well quasi-ordering on $RS(TS)$. This is the case for Petri nets because the usual ordering $\leqslant$ on vectors of integers is a well ordering.

DEFINITION 3.2. Let $(TS, \leqslant) = (\langle S, R, s_0 \rangle, \leqslant)$ be a quasi-ordered transition system and k be an integer such that $k \geqslant 1$. We say that the relation R (or the transition system TS) is **k-compatible with** $\leqslant$ if and only if

$$(\forall s \in S)\,(\forall s' \in R^+(s)), \qquad s \leqslant s' \Rightarrow R(s) \leqslant R^k(s'). \tag{k}$$

We say that the relation R (or the transition system TS) is **strictly k-compatible (or k′-compatible) with** $\leqslant$ if and only if

$$(\forall s \in S)\,(\forall s' \in R^+(s)), \qquad s < s' \Rightarrow R(s) < R^k(s'). \tag{k'}$$

We say that the relation R (or the transition system TS) is **compatible with** $\leqslant$ if and only if

$$(\forall s \in S)\,(\forall s' \in R^+(s)), \qquad s \leqslant s' \Rightarrow R(s) \leqslant R^+(s'). \tag{∞}$$

We say that the relation R is **strictly compatible with** $\leqslant$ if and only if

$$(\forall s \in S)\,(\forall s' \in R^+(s)), \qquad s < s' \Rightarrow R(s) < R^+(s'). \tag{∞'}$$

PROPOSITION 3.3. (∞) *is equivalent to* "$\forall s \in S\ \ \forall s' \in R^+(s),\ s \leqslant s' \Rightarrow R^*(s) \leqslant R^+(s')$."

(∞') *is equivalent to* "$(\forall s \in S)\,(\forall s' \in R^+(s)),\ s < s' \Rightarrow R^*(s) < R^+(s')$."

We show the two non-trivial implications by induction on $n \geqslant 0$. ($R^*(s) = \bigcup_{n \geqslant 0} R^n(s)$).

PROPOSITION 3.4 (Finkel, 1986). *Petri nets have the* 1'*-compatibility property.*

CONJECTURE 3.5. *Let* $(TS, \leqslant) = (\langle S, R, s_0 \rangle, \leqslant)$ *be a quasi-ordered transition system. Given two states s, s', the two problems*

$$R(s) \leqslant R^+(s')? \qquad \textit{and} \qquad R(s) < R^+(s')?$$

are both undecidable.

As a consequence, we also think that it is undecidable to know whether R is compatible (strictly compatible, respectively) with $\leqslant$. Moreover, given a transition system $TS = \langle S, R, s_0 \rangle$, finding a quasi-ordering $\leqslant$ such that R is compatible with $\leqslant$ (strictly compatible, respectively) seems to be very difficult in the general case. We introduce some particular cases where we can find such quasi-orderings.

Following the notations introduced in (Finkel, 1987), we say, sometimes, that $(TS, \leqslant)$ is **monotonous (strictly monotonous**, respectively) when R is compatible with $\leqslant$ (strictly compatible with $\leqslant$, respectively). When R is 1-compatible with $\leqslant$ (1'-compatibility, respectively) we also say (Finkel, 1987) that R is 1-monotonous (1'-monotonous, respectively).

We introduce a new tree, called the reduced reachability tree, associated with a quasi-ordered transition system. This reduced reachability tree allows us to decide the *FRTP* and the *FRSP*.

DEFINITION 3.6. Let $(TS, \leqslant)$ be a quasi-ordered transition system. The **reduced reachability tree** of $(TS, \leqslant)$, denoted by $RRT(TS, \leqslant)$ or by $RRT(TS)$ for short, is a rooted tree defined in the following way.

(1) The root r is labelled by the initial state s_0.

(2) A node m, labelled by s, $s \in S$, has no successor if and only if either $R(s) = \emptyset$, or there exists a node m_1, different from m, on the branch from r to m, labelled by s_1 with $s_1 \leqslant s$.

(3) If m is a node labelled by $s \in S$, which does not satisfy condition (2) then for every state $s' \in R(s)$, there exists a node m' in $RRT(TS)$, the successor of m, labelled by s'.

The **reduced reachability graph** denoted by $RRG(TS)$ is obtained from the reduced reachability tree by identifying nodes which have the same label.

When the reachability graph is finite, the reduced reachability graph is not necessarily equal to the reachability graph (Fig. 3.4.).

EXAMPLE 3.7. We modify the system $TS1$ to obtain a new system without any deadlock; we choose to change machine $M2$ in confusing state 1 and state 2. Let $TS2$ be the transition system (or protocol) shown in Fig. 3.1.

As machine $M2$ has only one state we can consider a 3-tuple $s = (n_1, w_1, w_2)$ instead of the previous 4-tuple. A state s of this net is, here, a 3-tuple $s = (n_1, w_1, w_2)$, where n_1 is an integer and w_1, w_2 are finite words. Let $\leqslant_2$ be the following quasi-ordering defined by

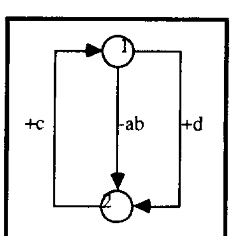
$$s = (n_1, w_1, w_2) \leqslant_2 s' = (n'_1, w'_1, w'_2)$$

if and only if

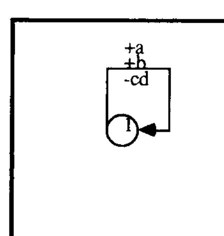
$$n_1 = n'_1, \qquad w'_1 \in w_1(ab)^*, \qquad w'_2 \in w_2(cd)^*.$$

Let us draw its reduced reachability graph (Fig. 3.2). We can show that this system has the monotonicity property for this quasi-ordering; moreover $\leqslant_2$ is a well quasi-ordering (Finkel, 1986).

For general quasi-ordered transition systems, the reduced reachability tree may not be finite. The importance of well quasi-orderings is stated by the following result.



M1



M2

FIG. 3.1. Protocol $P2$.

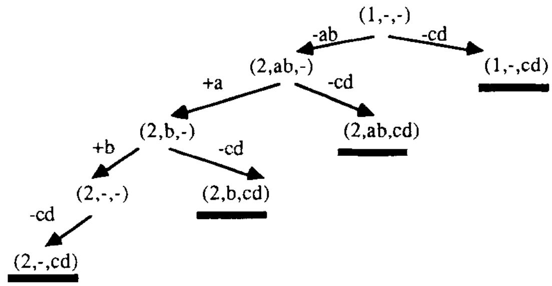


FIG. 3.2. The reduced reachability graph of *TS*2.

PROPOSITION 3.8. *The reduced reachability tree of a well-quasi-ordered transition system is finite.*

Proof. Suppose the reduced reachability tree is infinite. Since all nodes of the reachability tree of *TS* are of finite degree then so are the nodes of the reduced reachability tree. We can apply Koenig's Lemma. Let $r, m_1, m_2, ..., m_n, ...$ be the nodes of this infinite branch and let $s_0, s_1, s_2, ...,$ $s_n, ...$ be the sequence of corresponding labels. By hypothesis $\leqslant$ is a well quasi-ordering on $RS(TS)$, so there exists an infinite increasing subsequence $\{s_{np}\}$ such that for each $i \geqslant 0$, $s_{ni} \leqslant s_{ni+1}$. But this is in contradiction with the definition of the reduced reachability tree of *TS*; so the reduced reachability tree is finite. ∎

Remark 3.9. The usual ordering on $\mathbf{N}^p$ ($\mathbf{N} \times \cdots \times \mathbf{N}$, p times), $p \geqslant 1$, is a decidable well ordering (Dickson, 1913). The subword relation | is a well ordering on the set A^* of finite words (Higman, 1952). However, the left factor relation is not a well ordering on A^* (if $|A| \geqslant 2$). For example, the sequence $(a^nb)_{n \geqslant 0}$ does not contain any increasing subsequence for the left factor ordering.

In order to terminate the computation of the finite reduced reachability tree, we must be able to decide if a state s' is larger than a state s or if they are equal. We remark that equality is decidable if the quasi-ordering is decidable and if it is an ordering (reflexive, transitive, and antisymmetric relation); the converse is false.

We can now define a new structure on transition systems.

DEFINITION 3.10. A **structured transition system** is a quasi-ordered transition system $(S, R, s_0, \leqslant)$ such that:

(1) R is compatible with $\leqslant$,

(2) $\leqslant$ is a well quasi-ordering on the reachability set,

(3) $\leqslant$ is decidable.

A **strictly structured transition system** is a structured transition system such that R is strictly compatible with $\leqslant$ and such that the equality is decidable on the reachability set.

Sometimes we say that a class of transition systems is structured: it means that there exists a quasi-ordering such that each transition system of this class is structured for this quasi-ordering.

PROPOSITION 3.11. *Petri nets are 1′-structured for the usual ordering on vectors of integers.*

Proof. The 1′-monotonicity (Proposition 3.4) is proved, for example, in (Brams, 1983) or (Finkel, 1986). Dickson's Theorem (Dickson, 1913) says that the usual ordering on vectors of integers is a well ordering; then it is also a well ordering on the reachability set. At last, these two orderings $\leqslant$ and $=$ are decidable. ∎

The "miracle" of *PNs* is that the usual ordering on vectors of integers always gives the monotonicity property to every *PN* and it is also a well ordering.

The reduced reachability tree allows us to decide the finiteness of the reachability tree of a structured transition system and also the finiteness of the reachability set of a strictly structured transition system.

THEOREM 3.12. *The FRTP is decidable for structured transition systems.*

Proof. Let us first show the following equivalence:

$$RT(TS) \text{ infinite} \Leftrightarrow \exists s,\ s' \in RS(TS),\ s' \in R^+(s), \text{ and } s \leqslant s'.$$

That the right side implies the left side is a consequence of monotonicity. For the converse, let us suppose that $RT(TS)$ is infinite. Since $RT(TS)$ has a finite degree, there is an infinite branch issued from the root r, by Koenig's Lemma. By hypothesis, $\leqslant$ is a well quasi-ordering on $RS(TS)$, hence there exist two states s_n and s_p such that $s_p \in R^+(s_n)$ and $s_n \leqslant s_p$. By definition of the reduced reachability tree we have

$$\exists s,\ \exists s \in RS(TS)\ \exists s' \in RS(TS)\ s' \in RS(TS),\ s' \in R^+(s), \text{ and } s \leqslant s'$$

which is equivalent to

$$\exists s,\ \exists s,\ s' \in RS(TS)\ s' \in RRT(TS)\ s' \in R^+(s), \text{ and } s \leqslant s'.$$

The last equivalence is decidable and hence the finiteness of the reachability tree $RT(TS)$ is a decidable problem. ∎

THEOREM 3.13. *The FRSP is decidable for strictly structured transition systems.*

Proof. We use the same reasoning as in Theorem 3.12. We show the following equivalence:

$$RS(TS) \text{ infinite} \Leftrightarrow \exists s, s' \in RS(TS), s' \in R^+(s), \text{ and } s < s'.$$

That the right-hand side implies the left-hand side is a consequence of the strict monotonicity. For the converse, let us suppose that $RS(TS)$ is infinite. Then the reachability tree of TS ($RT(TS)$) is also infinite. Since $RT(TS)$ has a finite degree and $RS(TS)$ is infinite, by Koenig's Lemma, there is an infinite branch containing an infinity of differents states, issued from the root r. By hypothesis, $\leqslant$ is a well quasi-ordering. Hence there exists two states s_n and s_p such that $s_p \in R(s_n)$ and $s_n < s_p$. By definition of the reduced reachability tree, we have

$$\exists s, \exists s \in RS(TS)\ \exists s' \in RS(TS)\ s' \in RS(TS), s' \in R^+(s), \text{ and } s < s'$$

which is equivalent to

$$\exists s, \exists s, s' \in RS(TS)\ s' \in RRT(TS), s' \in R^+(s), \text{ and } s < s'.$$

We can see here that we need to be able to decide the equality relation between states; as a matter of fact, $<$ is defined from $\leqslant$ by $s < s'$ iff $s \leqslant s'$ and $s \neq s'$. Since the last equivalence is decidable, because of the decidability of the equality for strictly structured transition systems, then the finiteness of the reachability set $RS(TS)$ is a decidable problem. ∎

Analysis of Protocol P2. The reachability tree and the reachability set are both infinite. We can conclude that at last one channel is not bounded (i.e., it can contain words having an arbitrary length).

EXAMPLE 3.14. Let $P3$ be the following protocol (Fig. 3.3.) and let $TS3$ be its associated transition system. A state s of this system is, here, a

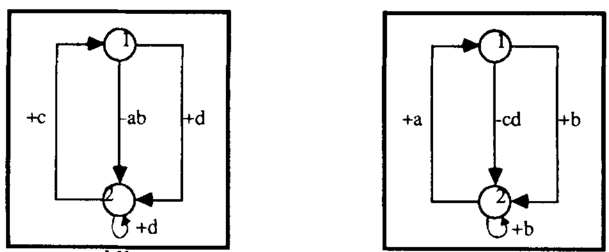


FIG. 3.3. Protocol $P3$.

4-tuple $s = (n_1, n_2, w_1, w_2)$, where n_1, n_2 are integers and w_1, w_2 are finite words. Let $\leqslant_3$ be the following quasi-ordering defined by

$$s = (n_1, n_2, w_1, w_2) \leqslant_3 s' = (n'_1, n'_2, w'_1, w'_2)$$

if and only if

$$n_1 = n'_1,\ n_2 = n'_2,\ w_1 \mid w'_1, \text{ and } w_2 \mid w'_2.$$

One can show that $(TS3, \leqslant_3)$ is structured. Let us draw its reduced reachability graph (Fig. 3.4). Let us draw the complete finite reachability graph of $TS3$ (see Fig. 3.5). Remark that the reduced reachability graph is not equal to the reachability graph of $TS3$ and that even $RG(TS3)$ is finite.

Analysis of Protocol P3. The reduced reachability graph shows us that the reachability tree and the reachability set are finite. We need the complete reachability graph to know that the two channels are bounded by 5; there is only one deadlock $(2, 2, -, -)$.

CONJECTURE 3.15. *The FRSP is undecidable for structured transition systems.*

The coverability problem of a structured transition system arises naturally. The reduced reachability tree of a structured transition system

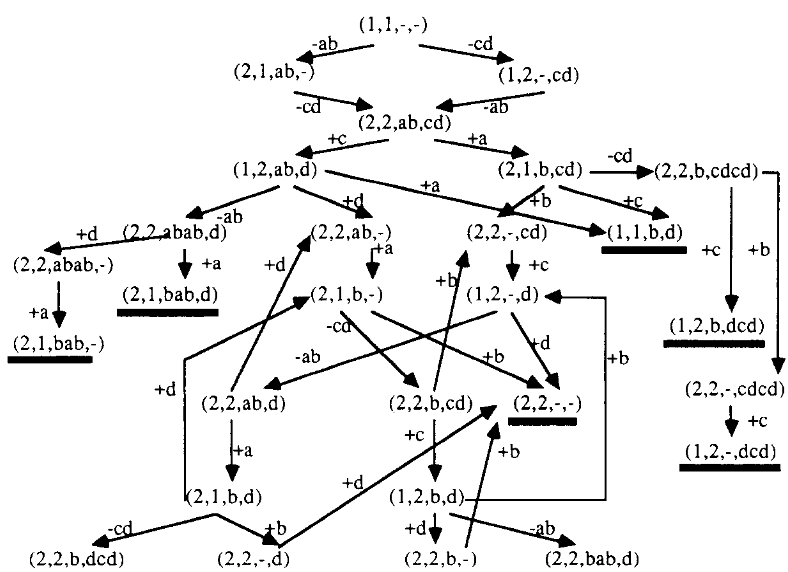


FIG. 3.4. The reduced reachability graph of $TS3$.

does not allow us to decide the *CP*. Intuitively speaking, this tree is too small and does not provide sufficient information about the system. Let us examine the reduced reachability tree in Fig. 3.2. For example, state $(1, -, cdcd)$ is reachable but it does not appear in *RRT*; hence, there is no way to know this fact using only the reduced reachability tree. In fact, to decide the *CP*, we need another finite representation of the reachability set, we call a coverability set.

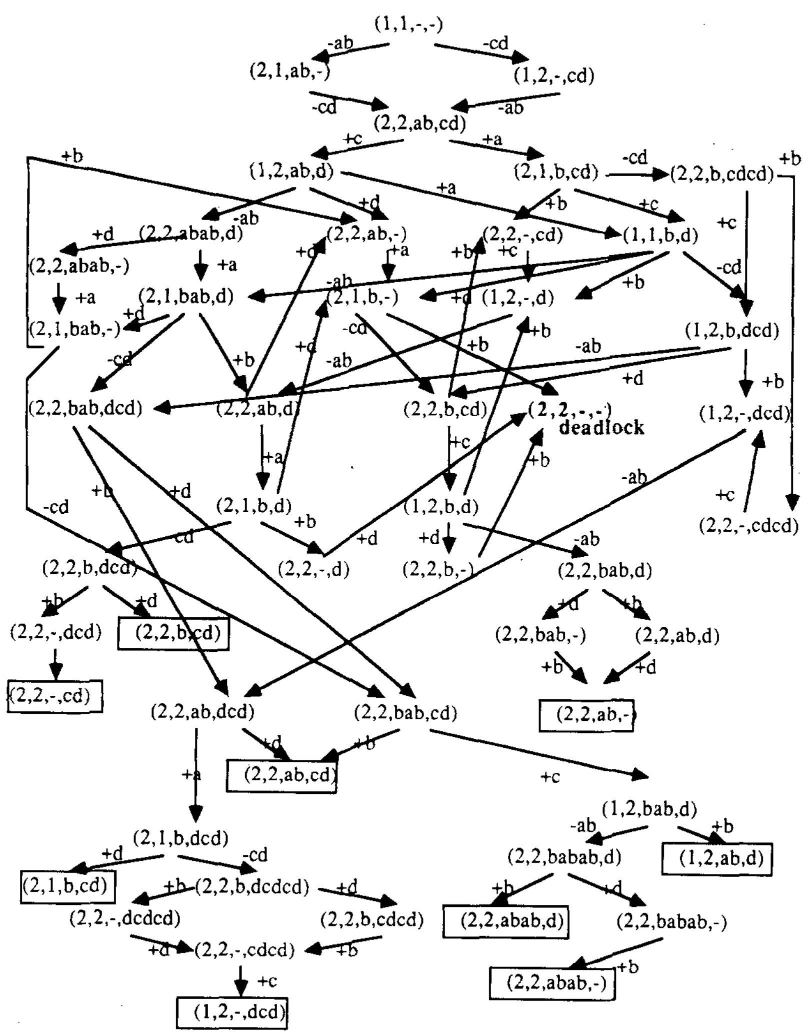


FIG. 3.5. The reachability graph of *TS*3.

4. Decidability of the Coverability Problem

In fact, from the ordering on integers, one may deduce, for Petri nets, more things than the decidability of the *FRTP* and of the *FRSP*; one may also decide the *CP*. What allows us to decide the *CP* in Petri nets is not exactly the coverability tree (Karp and Miller, 1969) but the set of labels of the coverability tree. This fact has been used for Petri nets in (Mayr, 1984) speaking about "maximum cover pseudomarkings." In this section, we generalize the notion of coverability tree in defining the coverability sets. The definition of the closed cover in (Gouda *et al.*, 1987) in the framework of communicating finite state machines is a particular case of our coverability set definition. We give a formal definition of coverability sets after having formally defined the limit of an infinite increasing sequence of elements in a quasi-ordered set.

4.1. *Completion by Increasing Sequences*

We define, in a quasi-ordered set, the limit of an infinite increasing sequence as the equivalence class of this sequence for the following equivalence relation:

Definition 4.1. Let $(X, \leqslant)$ be a quasi-ordered set (not necessarly a set of words) and $X_{\geqslant}$ be the set of infinite increasing sequences of elements of X. A sequence $\{x_n\}$, $x_n \in X$, is **inferior** to a sequence $\{y_n\}$ if and only if for every $n \geqslant 0$ there exists $p \in \mathbf{N}$ such that $x_n \leqslant y_p$. We also denote this relation by $\leqslant$. Two sequences $\{x_n\}$ and $\{y_n\}$ are **equivalent** when $\{x_n\} \approx \{y_n\} \Leftrightarrow \{x_n\} \leqslant \{y_n\}$ and $\{y_n\} \leqslant \{x_n\}$.

Notation 4.2. For every infinite increasing sequence $\{x_n\}$, we denote by $\lim x_n$ the equivalence class of $\{x_n\}$ for the equivalence relation $\approx$. We denote by limit X the quotient $X_{\geqslant/\approx}$. There is a canonical injection from X into limit X: to an element $x \in X$, we associate the equivalence class of the stationary sequence $\{x_n\}$ defined as follows: for every $n \geqslant 0$, $x_n = x$. We denote by $\text{limit}_{\text{si}}(X)$ the set of limits of infinite strictly increasing sequences of elements of X; $\text{limit}_{\text{si}^{n+1}}(X) = \text{limit}_{\text{si}}(\text{limit}_{\text{si}^n}(X))$ and $\text{limit}_{\text{si}^1}(X) = \text{limit}_{\text{si}}(X)$.

Example 4.3. We obtain with $X = \mathbf{N}^3$:

$\text{limit}_{\text{si}}(\mathbf{N}^3) = \{(\omega, \omega, \omega),\ (\omega, \omega, n),\ (\omega, n, \omega),\ (n, \omega, \omega),\ (\omega, n, p),\ (n, \omega, p),\ (n, p, \omega);\ n, p \geqslant 0\}$,

$\text{limit}_{\text{si}}(\text{limit}_{\text{si}}(\mathbf{N}^3) = \{(\omega, \omega, \omega), (\omega, \omega, n), (\omega, n, \omega), (n, \omega, \omega); n \geqslant 0\}$,

$\text{limit}_{\text{si}}(\text{limit}_{\text{si}}(\text{limit}_{\text{si}}(\mathbf{N}^3))) = \{(\omega, \omega, \omega)\}$,

$\text{limit}_{si}(\text{limit}_{si}(\text{limit}_{si}(\text{limit}_{si}(\mathbf{N}^3)))=\varnothing$ and then, for all $k \geqslant 4$, $\lim_{si^k}(\mathbf{N}^3) = \varnothing$.

More generally, we have, for all $k \geqslant q$, $\text{limit}_{si^k}(\mathbf{N}^q)=\varnothing$.

Remark 4.4. The quasi-ordering $\leqslant$ defined on X can be extended on limit X in the following way:

(1) For every $n \geqslant 0$, $x_n \leqslant \lim x_n$.

(2) If $\{x_n\} \leqslant \{y_n\}$ then $\lim x_n \leqslant \lim y_n$.

(3) If $\{\perp\}_n$ is the stationary sequence always equal to $\perp$ ($\perp$ is supposed to be the least element of X) one has, for each element $x \in \text{limit } X$, $\perp \leqslant x$.

When the set X is countable, the following result is well known.

PROPOSITION 4.5 (Birkhoff, 1967). *The set limit X, with the quasi-ordering $\leqslant$, is a directed ordered set.*

4.2. *Coverability Sets*

We now define, in the general framework of structured transition systems, the notion of a coverability set. This notion is central and it is more general than the notion of a coverability tree (Karp and Miller, 1969).

DEFINITION 4.6. Let $(TS, \leqslant)$ be a structured transition system. A **coverability set**, $CS(TS)$ or simply CS, of $(TS, \leqslant)$ is a set, included in $RS(TS) \cup \text{limit}(RS(TS))$, such that:

(1) for every reachable state s, there exists an element s' in the coverability set such that $s \leqslant s'$: $\forall s \in RS(TS)\ \exists s' \in CS(TS),\ s \leqslant s'$.

(2) for each element s' in the coverability set, there exists an increasing sequence of reachable states which converges to s': $\forall s' \in CS(TS)\ \forall n \geqslant 0\ \exists s_n \in RS(TS),\ s_n \leqslant s_{n+1}$, and $\lim s_n = s'$.

Remark 4.7. The reachability set is always a coverability set. The problem is to obtain a finite and effectively computable coverability set. In general, the reachability set is neither finite nor effectively computable.

Now we can state the main theoretical result of this section.

THEOREM 4.8. *Let $(TS, \leqslant) = (\langle S, R, s_0 \rangle, \leqslant)$ be a structured transition system. If there exists a finite and effectively constructible coverability set CS then the CP is decidable.*

Proof. Let CS be a coverability set of a structured transition system

$(TS, \leqslant) = (\langle S, R, s_0 \rangle, \leqslant)$. We show that given a state $s \in S$, there exists a reachable state $s' \geqslant s$ if and only if there exists an element $s'' \geqslant s$ in CS.

$\rightarrow$ If there exists a reachable state $s' \geqslant s$ then there exists s'' in CS (because CS is a cover) such that $s'' \geqslant s'$. Hence $s'' \geqslant s$.

$\rightarrow$ If there exists an element s'' in CS such that $s'' \geqslant s$ then there are two possibilities:

— s'' is in $RS(TS)$ and then s' can be equal to s'',

— s'' is not in $RS(TS)$. Hence s'' is the limit of an infinite strictly increasing sequence of states: $s'' = \lim u_n$.

As $s'' \geqslant s$ there exists an integer p such that $u_p \geqslant s$, s' can be equal to u_p. ∎

A coverability set CS is not minimal if there exists an element $s \in CS$ such that $CS\text{-}\{s\}$ is still a coverability set; we then define the notion of a minimal coverability set in the following way.

DEFINITION 4.9. A coverability set CS of TS is **minimal** iff every proper subset of CS is not a coverability set of TS.

It can be easily shown that a minimal coverability set is a set of non-comparable and maximal elements. In the case where the quasi-ordering $\leqslant$ is still a well quasi-ordering on $RS(TS) \cup CS(TS)$, every minimal coverability set if finite.

PROPOSITION 4.10. *Let $CS(TS)$ be a coverability set of TS and $\leqslant$ be a well quasi-ordering on $RS(TS)$ and $CS(TS)$. Then the associated minimal coverability set is finite.*

4.3. *How to Find a Finite and an Effectively Computable Coverability Set*

To be able to generalize the procedure of Karp and Miller, we must know how to construct effectively an infinite strictly increasing sequence of states from only its two first elements s_1 and s_2, where s_2 is reachable from s_1 and $s_1 < s_2$. A way to achieve this is to study a particular class of labelled structured transition systems.

DEFINITION 4.11. A structured labelled transition system $LTS = \langle S, A, h, s_0 \rangle$ is said **strictly structured** when $(\forall s, s' \in S)$ $(\forall a \in A)$ $(h(s, a) \neq \perp$ and $s < s' \Rightarrow h(s, a) < h(s', a))$.

We have to use the **continuous extension** of the function h of a labelled transition system $LTS = \langle S, A, h, s_0 \rangle$. We still denote this extension by h.

DEFINITION 4.12. The extension h: $(\text{limit}(S)) \times A \rightarrow \text{limit}(S)$ is defined by $h(\lim s_n, a) = \lim h(s_n, a)$.

According to the classical definitions in Petri net theory, we now define a coverability tree.

DEFINITION 4.13. A **coverability tree** is a tree CT such that Labels(CT) is a coverability set.

Our aim is to obtain an algorithm which computes one finite coverability set (not necessarily minimal). The previous results allow us to generalize the procedure for constructing a coverability tree, first defined by Karp and Miller in the framework of parallel program schemata (Karp and Miller, 1969), to the class of strictly structured labelled transition systems.

DEFINITION 4.14. The **generalized Karp and Miller tree**, $GKMT(LTS)$, of a strictly structured labelled transition system $(LTS, \leqslant)$ is constructed by the following procedure.

(1) The root is labelled by the initial state s_0.

(2) A node m, labelled by s, $s \in S$, has no successor if and only if for all actions a, $h(s, a) = \perp$ or there exists a node $m_1 \neq m$ also labelled by s, on the path from r to m.

(3) **If** m is a node labelled by s, $s \in S$, which does not satisfy condition (2), **then** for every s' such that $h(s, a) = s'$, there exists a node m', a successor of m, labelled by s''. The arc (s, s'') is labelled by a.

If there exists a node m_1, on the path from r to m, labelled by s_1 such that $s_1 < s'$ and $h(s_1, x) = s'$ (let m_1 be the first node from the root satisfying this condition),
then $s'' = \lim u_n$ with $u_1 = s_1$, $u_2 = s'$, $\forall n \geqslant 1$, $u_{n+1} = h(u_n, x) = h(s_1, x^n)$, $x \in A^*$
else $s'' = s'$.
The associated **generalized Karp and Miller graph**, $GKMG(LTS)$, is obtained by identifying nodes with the same label.

When the reachability graph is finite, the $GKMG$ is equal to the reachability graph.

EXAMPLE 4.15. Figure 4.1 presents the $GKMG$ of the transition system $TS2$ naturally considered as a labelled transition system. Let us recall that a state s of this net is a 3-tuple $s = (n_1, w_1, w_2)$, where n_1 is an integer and w_1, w_2 are finite words. The quasi-ordering $\leqslant_2$ is extended on infinite words as follows:

$$s = (n_1, w_1, w_2) \leqslant_2 s' = (n'_1, w'_1, w'_2)$$

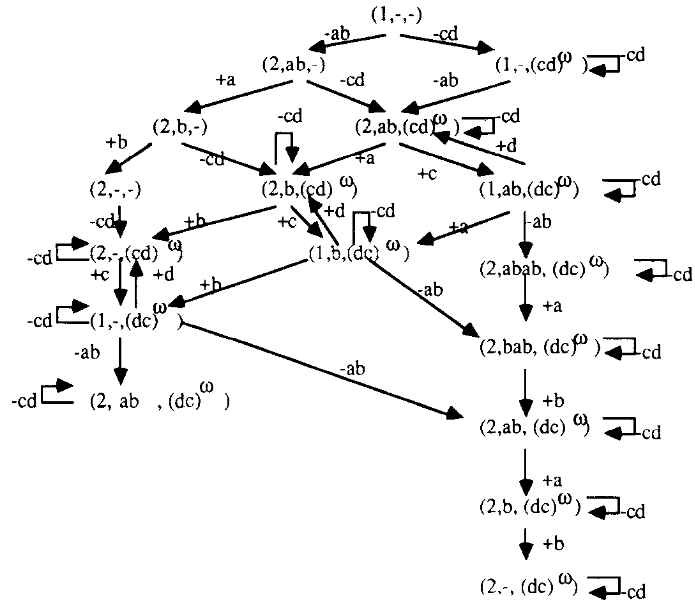


FIG. 4.1. The generalized Karp and Miller graph of $TS2$.

if and only if

$$n_1 = n_1', \qquad [w_1' \in w_1(ab)^* \text{ or } w_1' = w_1(ab)^{\omega}],$$
$$\text{and} \qquad [w_2' \in w_2(cd)^* \text{ or } w_2' = w_2(cd)^{\omega}].$$

For every $w_1, w_2 \in A^*$, we have $\lim(w_1 \cdot (ab)^n) = w_1 \cdot (ab)^{\omega}$ and $\lim(w_2 \cdot (cd)^n) = w_2 \cdot (cd)^{\omega}$.

The generalization of the procedure of Karp and Miller gives a finite coverability set (the set of labels of the coverability graph), but not a minimal coverability set. This coverability set, denoted by $CS(GKMG)$, naturally associated with the $GKMG$, contains 16 elements: $CS(GKMG) = \{(1, -, -),\ (2, ab, -),\ (1, -, (cd)^{\omega}),\ (2, b, -),\ (2, ab, (cd)^{\omega}),\ (2, -, -),\ (2, b, (cd)^{\omega}),\ (1, ab, (dc)^{\omega}),\ (2, -, (cd)^{\omega}),\ (1, b, (dc)^{\omega}),\ (2, abab, (dc)^{\omega}),\ (1, -, (dc)^{\omega}),\ (2, bab, (dc)^{\omega}),\ (2, ab, (dc)^{\omega}),\ (2, b, (dc)^{\omega}),\ (2, -, (dc)^{\omega})\}$. But $CS(GKMG)$ is not minimal; as a matter of fact, the minimal coverability set is the following set having only 7 elements: $\{(1, -, (cd)^{\omega}),\ (1, ab, (dc)^{\omega}),\ (1, b, (dc)^{\omega}),\ (2, abab, (dc)^{\omega}),\ (2, bab, (dc)^{\omega}),\ (2, b, (cd)^{\omega}),\ (2, ab, (cd)^{\omega})\}$. The generalization of the procedure of Karp and Miller always gives a coverability tree for structured labelled transition systems but in the general case, the $GKMT$ is neither finite nor effectively constructible.

Let us first show that the generalized Karp and Miller tree is a coverability tree.

THEOREM 4.16. *Let $(LTS, \leqslant)$ be a strictly structured labelled transition system. Then $GKMT(LTS)$ is a coverability tree.*

Proof. We need to show that every reachable state of a structured transition system is covered (for the quasi-ordering) by a node in its generalized Karp and Miller graph; and we also must show that for every node s in the Karp and Miller graph such that $s \in \text{limit}_{\text{si}}(RS(LTS))$, there exists an infinite strictly increasing sequence of reachable states converging to s. More precisely, we must show:

(1) $\forall s \in RS(LTS)$, $\exists s' \in \text{Labels}(GKMT(LTS))$ such that $s \leqslant s'$.

(2) $\forall s' \in \text{Labels}(GKMT(LTS))$, $s' \in \text{limit}_{\text{si}}(RS(LTS))$ implies that $\forall n \geqslant 0$, $\exists s_n \in RS(LTS)$, $s_n < s_{n+1}$, and $\lim s_n = s'$.

The first property can be shown by induction on the length of a path in the reachability tree of LTS from s_0 to s, then on the length of a word $x \in A^*$ such that $h(s_0, x) = s$. (Let us recall that h is a function from $S \times A$ into S.)

→ If $h(s_0, a) = s$, $a \in A$, then by definition of the $GKMT(LTS)$, there exists, in $GKMT(LTS)$, a path from s_0 to an element $u \in \text{limit}(RS(LTS))$ such that $s \leqslant u$.

→ Let us denote $|x| = k+1$, $x = x'a$, $|x'| = k$, and $a \in A$.

Let $h(s_0, x') = s$, $h(s, a) = s'$; then, by the induction hypothesis, there exists a path, labelled by x', in the $GKMT(LTS)$, from s_0 to u such that $s \leqslant u$. As $(LTS, \leqslant)$ is structured and h is continuous (h is extended on $\lim(RS(LTS))$), we have

$$s \leqslant u \text{ and } h(s, a) = s' \Rightarrow h(u, a) = u' \text{ and } s' \leqslant u'.$$

As there exists a path from u to u' in $GKMT(LTS)$, we deduce the first result.

We show (2) by induction on the number z of labels in $\text{limit}_{\text{si}}(RS(LTS))$ between the root r labelled by s_0 and a node m labelled by $s' \in \text{limit}_{\text{si}}(RS(LTS))$.

→ $z = 0$ means that every label between s_0 and s is in the reachability set of LTS; then, by definition of the $GKMT$, there exists two states $s_1, s_2 \in RS(LTS)$ (in the reachability tree) such that

$$s_1 < s_2, h(s_1, x) = s_2, s_2 < s', x \in A^*.$$

As LTS is strictly structured and h is continuous, there exists an infinite strictly increasing sequence of reachable states $\{s_n\}$ such that $\forall n \geqslant 1$ $h(s_n, x) = s_{n+1}$ and $\lim s_n = s'$.

$\rightarrow$ $z = k$, there exist k elements $(s_1, s_2, ..., s_k)$ of $\text{limit}_{\text{si}}(RS(LTS))$ which label k nodes from the root r (of the *GKMT*) to a node labelled by a $(k+1)$th element, s_{k+1}, of $\text{limit}_{\text{si}}(RS(LTS))$. Let us note $(h(s_k, x) = s_{k+1}$.

By the induction hypothesis, we know that there exists an infinite strictly increasing sequence $\{u_n\}$, $u_n \in S$, such that $\text{limit}\, u_n = s_k$. From $h(s_k, x) = s_{k+1}$, $x \in A^*$, we obtain

$$h(\lim u_n, x) = s_{k+1}.$$

Hence, by continuity of h,

$$\lim(h(u_n, x)) = s_{k+1}.$$

As $(LTS, \leqslant)$ is strictly structured and $\{u_n\}$ is strictly increasing, the sequence $h(u_n, x)$ is strictly increasing from an index p; this sequence admits s_{k+1} as a limit. ∎

To be able to effectively construct the *GKMT*, we must show that this tree is finite and that it is effectively constructible. It is not always the case.

DEFINITION 4.17. A labelled transition system $(LTS, \leqslant)$ is **well structured** if and only if the five following conditions are satisfied:

(1) $(LTS, \leqslant)$ is strictly structured;

(2) the well quasi-ordering $\leqslant$ on $RS(LTS)$ is still a well quasi-ordering on $\text{limit}(RS(LTS))$;

(3) there exists an integer k such that $\text{limit}_{\text{si}^k}(RS(LTS)) = \varnothing$;

(4) $\leqslant$ and $=$ are decidable on $\text{limit}(RS(LTS))$;

(5) for every $s \in \text{limit}(RS(LTS))$, $a \in A$, one can decide whether $h(s, a) = \perp$.

THEOREM 4.18. *The generalized Karp and Miller tree GKMT(LTS) of a well-structured labelled transition system is finite and it is effectively constructible.*

Proof. Let us suppose the contrary and suppose the generalized Karp and Miller tree is infinite. As the reachability tree of *LTS* has a finite degree, so does the generalized Karp and Miller tree. Then we can apply Koenig's Lemma. Let $r, m_1, m_2, ..., m_n, ...$ be the nodes of an infinite branch and let $s_0, s_1, s_2, ... s_n, ...$ be the sequence of corresponding labels. By hypothesis $\leqslant$ is a well quasi-ordering on limit S, so there exists an infinite increasing subsequence $\{s_{np}\}$ such that: for each $i \geqslant 0$, $s_{ni} \leqslant s_{ni+1}$. But this is in contradiction with the definition of the generalized Karp and Miller tree of LTS and with the fact that there exists an integer k such that $\text{limit}_{\text{si}^k}(RS(LTS)) = \varnothing$; so the generalized Karp and Miller tree is finite. As $\leqslant$ and $=$ are decidable on limit $(RS(LTS))$, the generalized Karp and Miller tree is effectively constructible. ∎

If condition (2) or (3) is not satisfied then the *GKMT* can be infinite. Let us remark that a well quasi-ordering on X is not always a well quasi-ordering on X is not always a well quasi-ordering on limit(X) (Parigot, 1986).

COUNTER-EXAMPLE 4.19. Let R the ordering defined on $\mathbf{N}^2$ by (x, y) $R(x'y')$ iff $(y = y'$ and $x \leqslant x')$ or $(y \neq y'$ and $y' \geqslant x + y)$ where $\leqslant$ is the usual ordering on $\mathbf{N}$. We verify that R is a well ordering on $\mathbf{N}^2$, but R is not a well ordering on limit $(\mathbf{N}^2) = (\mathbf{N} \cup \{\omega\})^2$ because the infinite set $\{(\omega, n);\, n \in \mathbf{N}\}$ does not contain two comparable elements.

Let us remark that the *GKMT* is not effectively constructible for structured transition systems because when we have $R(s) < R^+(s')$, $s' \in R^+(s)$, we cannot know whether there exists $s'' \in R^+(s')$ such that $s'' > s$. We can only know that there exists $s'' \in R^+(s')$ such that $s'' \geqslant s$.

If condition (4) or (5) is not satisfied then the *GKMT* cannot be effectively constructed.

We deduce the two following results from Theorems 4.8, 4.16, and 4.18.

COROLLARY 4.20. *For well-structured labelled transition systems, there exists a finite and effectively constructible coverability set.*

COROLLARY 4.21. *The CP is decidable for well-structured labelled transition systems.*

Let us formulate Theorems 3.12 and 3.13 in the framework of well-structured labelled transition systems by using *GKMG*(LTS).

COROLLARY 4.22. *Let* $(LTS, \leqslant)$ *be a well-structured labelled transition system.*

(1) *The reachability tree* $RT(LTS)$ *is infinite if and only if there exists a circuit in* $GKMG(LTS)$.

(2) *The reachability set* $RS(LTS)$ *is infinite if and only if there exists at least one element of* $\text{limit}_{\text{si}}(RS(LTS))$ *in* $GKMG(LTS)$.

Analysis of Protocol P2. We know that the reachability tree and the reachability set are both infinite. Now we can refine these results. Observing the *GKMG* (which contains 18 elements), we can say that the two channels are not bounded. There is no deadlock because there are circuits in *GKMG*, but machine *M*1 is blocked since the system reaches a state $(2, w_1, d \cdot w_2)$ for every word $w_1, w_2 \in A^\infty$. We can decide the *CP* or equivalently the Quasi-Liveness Problem (*QLP*) (Finkel, 1986).

PROPOSITION 4.23. *Petri nets are well structured.*

Proof. We examine the five conditions of the Definition 4.17.

(1) Petri nets are 1'-structured by Proposition 3.11, hence, they are also strictly structured.

(2) The usual ordering on vectors of integers is still a well ordering on $(\mathbf{N} \cup \{\omega\})p$.

(3) If *PN* is a Petri net with *p* places, we have $\text{limit}_{p+1}(RS(PN)) \subseteq \text{limit}_{p+1}(\mathbf{N} \cup \{\omega\})p = \varnothing$; as $RS(PN)$ is included in $(\mathbf{N} \cup \{\omega\})p$, hence $\text{limit}_{p+1}(RS(PN)) = \varnothing$.

(4) $\leqslant$ and $=$ are decidable on limit $(\mathbf{N} \cup \{\omega\})p$.

(5) For all $s \in \text{limit}(\mathbf{N} \cup \{\omega\})p$, $a \in A$, one can decide whether $h(s, a) = \perp$; as a matter of fact, it remains to verify if a state s (a vector with p components (integers)) is larger than or equal to the vector of precondition associated with action a. ∎

EXAMPLE 4.24. Let *LTS*4 be the following system (Fig. 4.2). A state s of this system is, here, a 4-tuple $s = (n_1, n_2, w_1, w_2)$, where n_1, n_2 are integers and w_1, w_2 are finite words. Let $\leqslant_4$ be the following quasi-ordering defined by

$$s = (n_1, n_2, w_1, w_2) \leqslant_4 s' = (n'_1, n'_2, w'_1, w'_2)$$

if and only if

$$n_1 = n'_1,\ n_2 = n'_2,\ w'_1 \in w_1(ab)^*, \text{ and } w'_2 \in w_2(cd)^*.$$

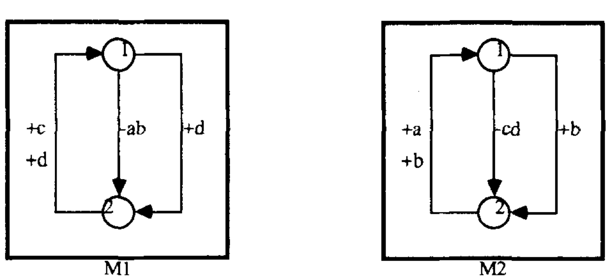


FIG. 4.2. Protocol *P*4.

One can show that $(LTS4, \leqslant_4)$ is well-structured. Let us draw a part of its finite *GKMG* (Fig. 4.3). The minimal coverability set is equal to $\{1, 2\} \times \{1, 2\} \times \{(ab)^\omega, (ba)^\omega\} \times \{(cd)^\omega, (dc)^\omega\}$ and it contains $2^4 = 16$ elements.

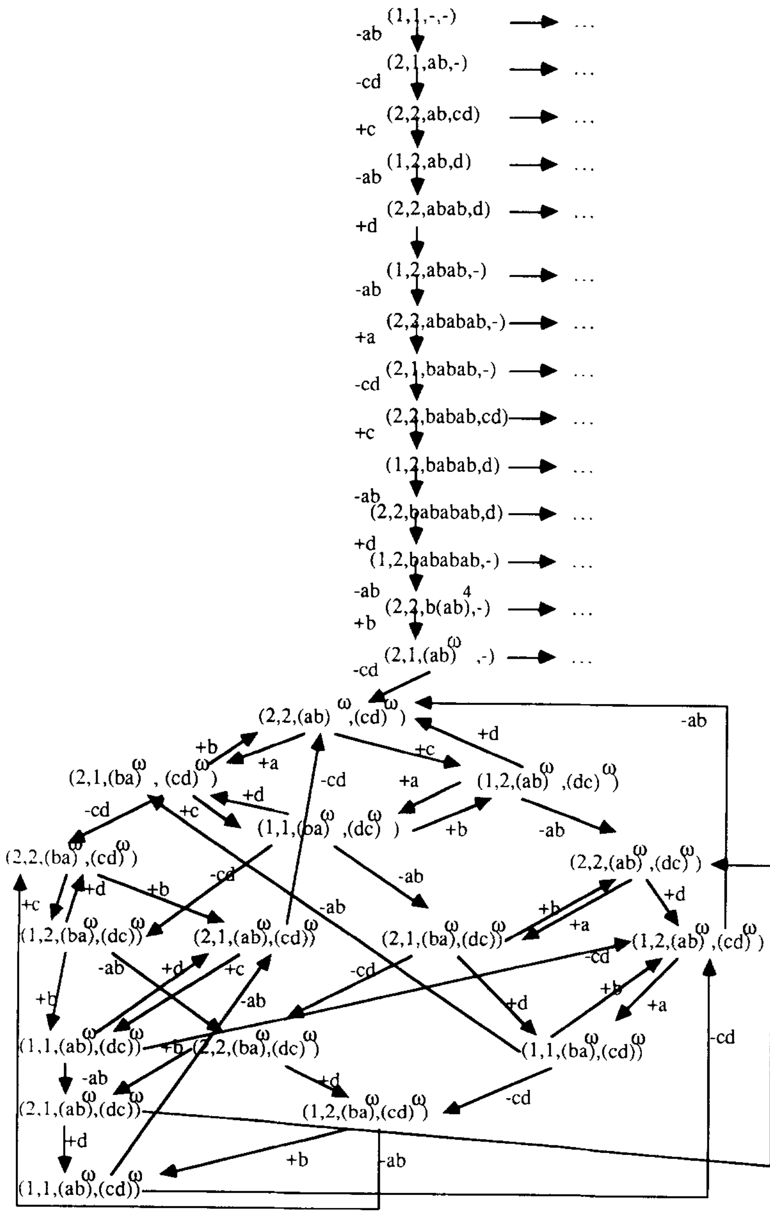


FIG. 4.3. A part of the generalized Karp and Miller graph of *TS4*.

We can deduce three corollaries.

COROLLARY 4.25. *The Quasi-Liveness Problem is decidable.*

COROLLARY 4.26. *The language $L(LTS)$ is included in the regular language defined by the finite automaton naturally associated with the GKMG such that every node of GKMG is a terminal state, the root is the initial state, the transition function is given by the paths of the GKMG, and the language is on the labels of the GKMG.*

COROLLARY 4.27. *Let LTS be a well-structured labelled transition system and let $x \in A^+$ such that x labels a circuit in $GKMG(LTS)$; then for every integer $n \geqslant 0$, there exists a word $y_n \in A^*$ such that $y_n x n \in L(LTS)$.*

Proof. Let $x \in A^+$ such that x labels a circuit in $GKMG(LTS)$. Let s be a node of the circuit in $GKMG(LTS)$ such that $h(s, x^n) = s$. With Theorem 4.16, one can deduce that there exists an infinite increasing sequence $\{s_p\}$ such that $\lim s_p = s$; then we obtain the first following equivalence.

$$\begin{aligned}
\forall n \geqslant 0, h(s, x^n) = s \Leftrightarrow\ & \forall n \geqslant 0, h(\lim s_p, x^n) = s \qquad \text{(by Theorem 4.16)} \\
\Leftrightarrow\ & \forall n \geqslant 0, \lim h(s_p, x^n) = s \text{ (by continuity of } h) \\
\Rightarrow\ & \forall n \geqslant 0, \exists q_n \in \mathbf{N}, \exists y_{qn} \in A^* \text{ such that } h(s_0, y_{qn}) = s_{qn} \\
& \text{and } h(s_{qn}, x^n) \neq \perp \\
\Rightarrow\ & \forall n \geqslant 0, \exists q_n \in \mathbf{N}, \exists y_{qn} \in A^* \\
& \text{such that } h(s_0, y_{qn} \cdot x^n) \neq \perp. \quad \blacksquare
\end{aligned}$$

5. TRANSITION SYSTEMS HAVING AN INFINITE SET OF TERMINAL STATES

Sometimes we need to refine the behaviour of a transition system by considering a set of terminal states. If we look at a general set of terminal states, we increase the power of almost all models to the power of Turing machines. Finite sets of terminal states have been studied for Petri nets (Hack, 1976; Peterson, 1981); when problems are not undecidable they are often equivalent to the reachability problem (Hack, 1976).

We first need to define transition systems having a set $\mathscr{T}$ of terminal states.

5.1. *Definitions*

We first give some definitions and the new list of the six problems we are interested in solving.

DEFINITION 5.1. Let TS be a transition system having a set $\mathscr{T}$ of terminal states ($\mathscr{T}$ is not necessary included in S). The **reachability tree** for $(TS, \mathscr{T})$, denoted by $RT(TS, \mathscr{T})$, is simply the reachability tree for TS pruned by truncating a path whenever it leaves $\mathscr{T}$. (Let us recall that R is the binary relation included in $S \times S$.) We define the sets $R^n((TS, \mathscr{T}), s)$, or simply $R^n(\mathscr{T}, s)$, where s is a terminal state and n is an integer, by the following: $R^0(\mathscr{T}, s) = \{s\}$, $R^1(\mathscr{T}, s) = R(\mathscr{T}, s) = \{s' \in S \cap \mathscr{T};\ (s, s') \in R\}$, $R^{n+1}(\mathscr{T}, s) = R((R^n(\mathscr{T}, s))$, with $R(\mathscr{T}, s, u) = R(\mathscr{T}, s) \cup R(\mathscr{T}, u)$. The transitive closure R^* of R is defined by $R^*(\mathscr{T}, s) = R^0(\mathscr{T}, s) \cup R^1(\mathscr{T}, s) \cup \cdots \cup R^n(\mathscr{T}, s) \cup \cdots$. Hence, the **reachability set** $RS(TS, \mathscr{T})$ is the set of states that are reachable terminal states and it is equal to $RS(TS, \mathscr{T}) = R^*(\mathscr{T}, s_0) = RS(TS) \cap \mathscr{T}$. The set $R^+(\mathscr{T}, s)$ is defined by $R^+(\mathscr{T}, s) = R^*(\mathscr{T}, s)\text{-}\{s\}$. A state s is a **deadlock state** (or simply a **deadlock**) when $R^+(\mathscr{T}, s) = \varnothing$.

We now define the problems we are interested in solving in the framework of transition systems having a set of terminal states:

1. The **Finite Reachability Tree Problem (FRTP)**: Is $RT(TS, \mathscr{T})$ finite?

2. The **Finite Reachability Set Problem (FRSP)**: Is $RS(TS, \mathscr{T})$ finite?

3. The **Coverability Set Problem (CSP)**: Given a state $s \in S$, is there a reachable state $s' \in RS(TS, \mathscr{T})$ such that $s' \geqslant s$?

4. The **Reachability Problem (RP)**: Given a state $s \in S$, does $s \in RS(TS, \mathscr{T})$?

5. The **Reachability Sets Inclusion Problem (RSIP)**: Given two states $s, s' \in S$, is the reachability set from s included in the reachability set from s': $R^*(s) \subseteq R^*(s')$?

6. The **Deadlock Problem (DP)**: Is there a deadlock state?

Remark 5.2. The *FRTP* is now reducible to the following problem: given a reachable state s, does there exist a state s', reachable from s, such that s belongs to $\mathscr{T}$? That is,

$$\text{is } R^+(s) \cap \mathscr{T} = \varnothing \text{ or equivalently } R^+((TS, \mathscr{T}), s) = \varnothing?$$

This last problem is in fact also the Empty Reachability Set Problem. Without any more information on $\mathscr{T}$, it seems that this problem and the other five problems are all undecidable.

Here we consider infinite set of terminal states which have a structure allowing us to decide some of our six problems. Structured sets of terminal markings of a Petri net have been introduced for the first time in (Choquet and Finkel, 1987). We generalize this notion to general structured transition systems having a structured set of terminal states.

DEFINITION 5.3. Let $(TS, \leqslant) = (\langle S, R, s_0 \rangle, \leqslant)$ be a structured transition system and $\mathscr{T}$ be a set of terminal states. The set $\mathscr{T}$ is a **structured set of terminal states** iff the four following conditions are satisfied:

(1) Membership in $\mathscr{T}$ is decidable.

(2) The initial state s_0 belongs to $\mathscr{T}$.

(3) Each state reached on a path into $\mathscr{T}$ must be in $\mathscr{T}$. More formally, if $h(s_0, x) = s$, $s \in \mathscr{T}$, then for every $n \geqslant 1$, $s_n \in \mathscr{T}$, where s_n is the state reached after the firing of the sequence $x[n]$ $(h(s_0, x[n]) = s_n)$.

(4) Let $h(s, x) = s_1$, $s \leqslant s_1$, $s, s_1 \in \mathscr{T}$, and $\{s_n\}$ be an infinite increasing sequence such that for every $n \geqslant 1$, s_n is the state reached after the firing of the sequence z_n $(h(s_n, z_n) = s_{n+1})$, where $z_1 = x$ ($\{s_n\}$ can be constructed because of the monotonicity property); then for every $n \geqslant 1$, $s_n \in \mathscr{T}$.

Remark 5.4. From the properties of the structured set of terminal states, it follows that we can build a reachability tree: along a branch, we stop as soon as we meet a non-terminal state, which makes sense because after a non-terminal state, we are sure that we will not meet any other terminal state. The reachability tree $RT(TS, \mathscr{T})$ for $(TS, \mathscr{T})$ is simply the reachability tree for TS pruned by truncating a path whenever it leaves $\mathscr{T}$. Hence the reachability set $RS(TS, \mathscr{T})$ is equal to $RS(TS) \cap \mathscr{T}$. Note that in order to construct this tree, we must be able to decide whether or not a state is terminal.

5.2. *Decidability of the FRTP and the FRSP*

As $\mathscr{T}$ is structured, we may also consider the **reduced reachability tree** $RRT(TS, \mathscr{T})$: we stop along a branch either after we encounter a node labelled by a state that is greater than (or equal to) the labelling of one of its ancestors, or before a node that would be labelled by a non-terminal state. According to the well quasi-ordering, the reduced reachability tree is, of course, finite.

PROPOSITION 5.5. *Let $(TS, \leqslant)$ be a structured transition system having two structured sets of terminal states $\mathscr{T}_1$ and $\mathscr{T}_2$; then $\mathscr{T}_1 \cup \mathscr{T}_2$ and $\mathscr{T}_1 \cap \mathscr{T}_2$ are also two structured sets of terminal states for $(TS, \leqslant)$.*

THEOREM 5.6. *Let $(TS, \leqslant)$ be a structured transition system and let $\mathscr{T}$ be a structured set of terminal states. Then the FRTP is decidable.*

Proof. Let us show that $RT(TS, \mathscr{T})$ is infinite if and only if there exist two states $s, s_1 \in RT(TS, \mathscr{T})$ such that $s_1 \in R^+(s)$ and $s \leqslant s_1$.

First, we assume that $RT(TS, \mathscr{T})$ is infinite. Then from Koenig's Lemma, we deduce that there is an infinite branch in the reachability tree $RT(TS, \mathscr{T})$. By the well quasi-ordering, we obtain an infinite increasing subsequence of states in $RT(TS, \mathscr{T})$. Thus, there exist two states s, $s_1 \in RT(TS, \mathscr{T})$ such that $h(s, x) = s_1$ and $s \leqslant s_1$.

Let us suppose the converse. By hypothesis, there exist two states s, $s_1 \in RT(TS, \mathscr{T})$ such that $h(s, x) = s_1$ and $s \leqslant s_1$. Since $\mathscr{T}$ is structured, there exists $s_n \in \mathscr{T}$, for every $n \geqslant 1$, $h(s, z_n) = s_n$, $s_n \leqslant s_{n+1}$. Hence $RT(TS, \mathscr{T})$ is infinite.

The *FRTP* is decidable with the help of the reduced reachability tree: one need only search the reduced reachability tree for a state that is greater than or equal to one of its ancestors. ∎

THEOREM 5.7. *Let $(TS, \leqslant)$ be a strictly structured transition system and let $\mathscr{T}$ be a structured set of terminal states. Then the FRSP is decidable.*

Proof. Let us first show that $RS(TS, \mathscr{T})$ is infinite if and only if there exist two states s, $s_1 \in R(TS, \mathscr{T})$ such that $h(s, x) = s_1$ and $s < s_1$. Suppose that $RT(TS, \mathscr{T})$ is infinite. We use the same reasoning as in Theorem 5.6 and we obtain an infinite strictly increasing subsequence of states in $RT(TS, \mathscr{T})$. Note that now the subsequence can be and must be chosen as strictly increasing instead of as only increasing as in Theorem 5.6. Thus, there exist two states s, $s_1 \in RT(TS, \mathscr{T})$ such that $h(s, x) = s_1$ and $s < s_1$.

Let us now show the converse. We suppose there exist two states s, $s_1 \in RT(TS, \mathscr{T})$ such that $h(s, x) = s_1$ and $s < s_1$. As $\mathscr{T}$ is structured, for every $n \geqslant 1$, $h(s, z_n) = s_n$, $s_n \leqslant s_{n+1}$, with $s_n \in \mathscr{T}$. As $\{s_n\}$ is an infinite strictly increasing sequence of states of $RS(TS, \mathscr{T})$, we deduce that $RS(TS, \mathscr{T})$ is infinite. The *FRSP* is decidable with the help of the reduced reachability tree. One need only search the reduced reachability tree for a state that is greater than one of its ancestors. ∎

THEOREM 5.8. *The reachability problem for structured transition systems having a structured set of terminal states is equivalent to the reachability problem for structured transition systems.*

Proof. One first verifies whether or not a given state is a terminal state (if not, it is not a reachable state with respect to the terminal states), which is decidable because $\mathscr{T}$ is structured, and then, again because of the structure of $\mathscr{T}$, one need only check if this state is reachable in S without considering $\mathscr{T}$ at all. ∎

As the *RP* is decidable for Petri nets (Mayr, 1984), we deduce the following result.

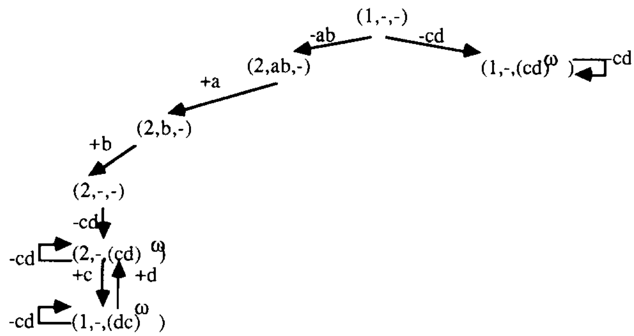


FIG. 5.1. The generalized Karp and Miller graph of *TS*2 with terminal states.

	FRTP	FRSP	CP	RP	DP	RSIP
structured	D	?U	?U	?U	?U	U
s-structured	D	D	?U	?U	?U	U
well-structured	D	D	D	?U	?U	U
Petri nets	D	D	D	D	D	U
structured-ssts	D	?U	?U	?U	?U	U
s-structured-ssts	D	D	?U	?U	?U	U
w-structured	D	D	?	?U	?U	U
Petri-ssts	D	D	D	D	?D	U

FIG. 6.1. The notation *s*-structured means strictly structured, *w*-structured means well structured, and *ssts* means with structured set of terminal states. *U* denotes undecidable, *D* means decidable, ? means an open problem without any conjecture, and ?*U* (?*D*, respectively) means that it is conjectured undecidable (decidable, respectively).

COROLLARY 5.9. *The RP is decidable for Petri nets having a structured set of terminal markings.*

5.3. *The Difficulty in Constructing a Coverability Tree*

For well-structured transition systems having a structured set of terminal states, one may construct a Karp and Miller graph. The rules are the same as in the usual case, but all the nodes must be labelled by states belonging to $\mathscr{T}$ or to $\text{limit}_{\text{si}}(\mathscr{T})$. States that do not belong to one of these sets will terminate the branch on which they should have appeared in the "usual Karp and Miller graph." We can then use this Karp and Miller graph instead of the reduced reachability tree for deciding the *FRTP* or the *FRSP*. But this Karp and Miller graph is no richer than the usual one. Problems arise because a limit state can cover terminal state as well as non-terminal state.

PROPOSITION 5.10. *The generalized Karp and Miller graph of a well-structured labelled transition system having a structured set of terminal states is not a coverability graph.*

Proof. We show that the set of labels of $GKMG(LTS)$ is not a coverability set of the reachability set. Let $\mathscr{T}_2$ be the maximal structured set of terminal states satisfying $\mathscr{T}_2 \subseteq \{s = (n_1, w_1, w_2) : |w_2| \geqslant 2$ implies $|w_1| = 0\}$. Fig. 5.1 presents the Karp and Miller graph of $(TS2, \mathscr{T}_2)$.

The set of labels of this generalized Karp and Miller graph is not a coverability set of the reachability set because, for example, the state $s = (2, ab, d)$ is reachable but there is no element s' such that $s' \geqslant_2 s$, which appears in the graph. Therefore, this generalized Karp and Miller graph does not seem to be a sufficient tool with which a coverability set can be constructed. ∎

6. CONCLUSION

We summarize the known results and the open problems in Fig. 6.1.

ACKNOWLEDGMENTS

I thank Marc Comeau and the two anonymous referee for their careful reading of the first and second versions of this paper.

References

BIRKHOFF, G. (1967), "Lattice Theory," 3rd ed., Amer. Math. Soc., Providence, RI.

BOCHMANN, G. (1978), Finite state description of communication protocols, *in* "Proceedings, Comp. Network Protocols Symp.," pp. 361–371, Liège, Belgium.

BOCHMANN, G., AND FINKEL, A. (1988), "Impact of Queued Interaction on Protocol Specification and Verification," Internal Report of the University of Montréal, No. 650; *in* "2nd International Symposium on Interoperable Information Systems (ISIIS'88), Tokyo, Japan, 10–11 November 1988."

BRAMS, G. W. (1983), "Réseaux de Petri: Théorie et pratique," Vol. 1, Masson, Paris.

BRAND, D., AND ZAFIROPULO, P. (1983), On communicating finite state machines, *J. Assoc. Comput. Mach.* **2**, 323–342.

BRAUER, W., REISIG, W., AND ROSENBERG, G. (Eds.) (1986), Petri nets: Central models and their properties, *in* "Advances in Petri Nets 1986," Part 1, "Proceedings, Advanced Course at Bad Honnef," Lecture Notes in Computer Science, Vol. 255, Springer-Verlag, Berlin.

CHOQUET, A. (1987), "Analyse et propriétés des processus communiquant par files fifo: Réseaux à files à choix libre topologique et réseaux à files linéaires," Thèse de 3-ième cycle, Université Paris 11.

CHOQUET, A., AND FINKEL, A. (1987), "Simulation of Linear Fifo Nets by Petri Nets Having a Structured Set of Terminal Markings," Rapport LRI No. 324, Univ. Paris 11, Orsay; *in* "8th European Workshop on Appl. and Theo. of Petri Nets, Zaragoza, Spain."

DICKSON, L. E. (1913), Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors, *Amer. J. Math.* **35**, 413–422.

FINKEL, A. (1986), "Structuration des systèmes de transitions: Applications au contrôle du parallélisme par files Fifo," Thèse d'Etat Université d'Orsay.

FINKEL, A. (1987), A generalization of the procedure of Karp and Miller to well structured transition systems *in* "14th International Colloquim Automata, Languages, Programming, Karlsruhe, R.F.A," pp. 409–508, Lectures Notes in Computer Science, Vol. 267, Springer-Verlag, Berlin.

FINKEL, A. (1988), "A New Class of Analysable CFSMs with Unbounded Fifo Channels: Application to Communication Protocol and Distributed Solution of the Mutual Exclusion Problem," Internal Report of the University of Montréal, No. 639; (revised version) A new class of analyzable CFSMs with unbounded Fifo channels, *in* "8th International Symposium on Protocol Specification, Testing, and Verification, Atlantic City, New Jersey, USA," IFIP WG 6.1 88.

GOUDA, M., GURARI, E., LAI, T., AND ROSIER, L. (1987), On deadlock detection in systems of communicating finite state machines, *Comput. Artificial Intelligence* **6** (3), 209–228.

HACK, M. (1975), "Petri Net Language," Comp. Struct. Group Memo 124, Project MAC, Massachusetts Institute of Technology, Cambridge, MA.

HACK, M. (1976), "Decidability Questions for Petri Nets," MIT, LCS, TR 161, Cambridge, MA.

HIGMAN, G. (1952), Ordering by divisibility in abstract algebras *in* "Proceedings, London Math. Soc. 2."

HOPCROFT, J., AND ULLMAN, J. (1979), "Introduction to Automata Theory, Languages, and Computation," Addison-Wesley, Reading, MA.

JENSEN, K. (1981), Coloured Petri nets and the invariants method, *Theoret. Comput. Sci.* **14** (3), 317–336.

KARP, R. M., AND MILLER, R. E. (1969), Parallel program schemata, *J. Comput. System Sci.* **4**, 147–195.

KASAI, T., AND MILLER, R. E. (1982), Homomorphisms between models of parallel computation, *J. Comput. System Sci.* **25**, 285–331.

KELLER, R. M. (1972), "Vector Replacement Systems: A Formalism for Modeling Asynchronous Systems," Tech. Rep. No. 117, Princeton Univ.

KELLER, R. M. (1976), Formal verification of parallel program, *Comm. ACM* **19** (7), 371–384.

KOENIG, D. (1936), "Theorie der Endlichen und Unendlichen Graphen," Akademische Verlagsgesellschaft, Leipzig.

KOSARAJU, S. R. (1982), Decidability of reachability in vector addition systems, *in* "Proceedings, 14th Ann. ACM Symp. on Theor. of Comp."

KRUSKAL, J. B. (1972), The theory of well-quasi-ordering: A frequently rediscovered concept, *J. Combin. Theory Ser. A* **13**.

MAYR, E. W. (1984), An algorithm for the general Petri net reachability problem, *in* "SIAM J. Computing," Vol. 13, No. 3.

MEMMI, G., AND FINKEL, A. (1985), An introduction to Fifo nets—monogeneous nets: A subclass of Fifo nets, *Theoret. Comput. Sci.* **35**.

PACHL, J. (1986), Protocol description and analysis based on a state transition model with channel expressions, *in* "6th Inter. Workshop on Protocol Specification, Testing, and Verification, Montréal, Québec," IFIP 86, North-Holland, Amsterdam.

PARIGOT, M. (1986), Private communication.

PETERSON, J. L. (1981), "Petri Net Theory and the Modelling of Systems," Prentice-Hall, Princeton, NJ.

ROSIER, L. E., AND GOUDA, M. (1984), On deciding progress for a class of communicating protocols, *in* "Proceedings, 8th Ann. Conf. on Inf. Sci. and Syst., Princeton University," pp. 663–667.

ROSIER, L. E., AND YEN, H. C. (1986), Boundedness, empty channel detection, and synchronisation for communicating finite automata, *Theoret. Comput. Sci.* **44** (1), 69–106.

VALK, R., AND JANTZEN, M. (1985), The residue of vector sets with applications to decidability problems in Petri nets, *Acta Inform.* **21**.