Testing for the unboundedness of fifo channels in programs (1)

Thierry Jéron

INRIA

UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tél.(1) 39 63 55 11

Rapports de Recherche

N° 1159

Programme 3
Réseaux et Systèmes Répartis

TESTING FOR THE UNBOUNDEDNESS OF FIFO CHANNELS IN PROGRAMS [1]

Thierry JERON

Février 1990

RR-1159

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CÉDEX
FRANCE
Téléphone : 99 36 20 00
Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

Publication Interne n°510 - Janvier 1990 - 30 Pages

Testing for the unboundedness of fifo channels in programs [1]

Thierry JERON

Abstract :

Unsolvability of the unboundedness problem for specification models allowing fifo channels was proved a few years ago by Brand and Zafiropulo. The paper investigates a testing approach of that problem. Instead of reducing the model in order to give decidability results, we work with the largest possible framework. We find a sufficient condition for unboundedness based on a relation between reachable global states. This gives a testing procedure which can be applied as well to communicating finite state machines as to Fifo-Nets. Moreover, the test extends existing decidability results. In fact it becomes a decision procedure for a class of systems strictly including linear and monogeneous systems. A few modifications of the relation make it available for Estelle specifications.

Test du caractère non borné de canaux fifo dans les programmes

Résumé :

Brand et Zafiropulo ont prouvé que, pour des modèles de spécification autorisant des canaux fifos, savoir si le contenu de ces canaux est borné ou non est indécidable. Cet article examine une approche du problème orientée vers le test. Au lieu de réduire le modèle afin d'en déduire des résultats de décidabilité, nous travaillons sur le modèle le plus général possible. Nous trouvons une condition suffisante du caractère non borné fondée sur une relation entre états globaux accessibles. Ceci fournit une procédure de test assez générale pour être appliquée aussi bien aux automates communicants qu'aux réseaux de Petri à files. De plus, ce test généralise les résultats de décidabilité existants. En effet, il fournit une procédure de décision pour une classe de systèmes incluant strictement les systèmes linéaires et monogènes. De très légères modifications apportées à la relation entre états globaux permettent également d'appliquer le test à des spécifications écrites en Estelle.

[1] This work has been done in the ADP research team of the Irisa laboratory and is submitted to the 10^{th} Symposium IFIP WG 6.1, Ottawa, June 90

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(L. A. 227)
UNIVERSITÉ DE RENNES 1 I. N. S. A. DE RENNES

INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
(LABORATOIRE DE RENNES)

Contents

1 Introduction

The problem of verifying the specifications of distributed protocols is of major importance in their development. If these protocols communicate by messages through fifo channels, their specifications have to model the fifo mechanism implicitely or explicitely. The main specification models which explicitely describe fifo channels are communicating finite state machines (CFSMs) [Boc78], Fifo-Nets [MM81] and some specification languages like Estelle [ISO89]. If fifos are not forehand bounded, the behaviour of such protocol specifications can lead to a channel overflow. But the computation of bounds, which is a wished result of a verification, is often impossible to perform. However, this result is often requested for the use of verification tools like model-checking, which only work on finite state graphs.

So, we would like to improve those verification tools with some tests for boundedness and unboundedness. But it is a very difficult problem as the very little number of applicable results proves it.

In our paper, we choose not to focus on a particular specification model but to work with the framework of some transition systems general enough to describe the behaviour of most of the specification models allowing fifo channels.

The paper is organized as follows. Section 2 begins with a few results about combinatorics on words. We then define a general transition system we are working with and introduce the unboundedness problem. After a recall of the main decidability results for CFSMs and Fifo-Nets, we give some arguments in favor of the testing approach.

In section 3, we present a new relation between global states of the reachability tree and prove that it gives a sufficient condition for unboundedness. We then give an algorithm implementing this test.

In section 4 we show that our test generalizes the known decidability results by proving that it becomes a decision procedure for a class of systems strictly including linear and monogeneous systems which were the only non trivial classes for which decidability of unboundedness was known [GGLR87, CF87, Fin86, Fin88]. We then give a few examples of CFSMs enlightening the use of our unboundedness test. We conclude with some indications on how this test can be applied to Estelle specifications.

2 The unboundedness problem

2.1 Notations and definitions

2.1.1 Combinatorics on words

Let A be a finite *alphabet*, the elements of which are called *letters*.

A finite sequence $x = (a_1, a_2, \ldots, a_n)$ of *letters* is called a *word* and is denoted by $a_1 a_2 \ldots a_n$. Its length is $|x| = n$.

The set A^* is composed of all the words over A and $A^+ = A^* - \{\epsilon\}$ where ϵ is the empty word.

The *concatenation* of two words $x = a_1 \ldots a_n$ and $y = b_1 \ldots b_m$ is the word $x.y = a_1 \ldots a_n b_1 \ldots b_m$. The neutral element of this operation is ϵ.

If $z = x.y$ we say that x is a *left factor* of z. Then y is a *right factor* of z and can be written $y = x^{-1}.z$.

$LF(z)$ and $RF(z)$ will denote the sets of left factors and right factors of z.

The left factor usually defines a partial ordering $\leq$ on words called the *prefix ordering* :

$$x, y \in A^*, x \leq y \Leftrightarrow x \text{ is a left factor of } y$$

A subword of a word is a left factor of a right factor of this word. Let $x = a_1 \dots a_n$ be a word, if $i \leq j$ the subword $a_i \dots a_j$ will be denoted $x[i \dots j]$.

If n is a positive integer, the concatenation of n words equal to x is denoted x^n. We say that x^n is a *power* of x. The set x^* is the set of all powers of x (including $\epsilon = x^0$).

A nonempty word that cannot be written as a power of another word is called a *primitive* word. That is a word x such that

$$x \neq \epsilon \text{ and } [\forall y \in A^+, (x \in y^* \Rightarrow x = y)]$$

Our test for unboundedness explicitely works with channel contents and their transformations, so we prove some results in combinatorics on words. Most of them are consequences of the definition of a *code*. The omitted proofs can be found in [Lot83].

Proposition 1 *Let $x, y \in A^+$, $n, m > 0$.*
If $x^n = y^m$, there exists a unique primitive word z such that $x, y \in z^*$

Definition 1 *We say that two words x and y commute if and only if $x.y = y.x$*

Theorem 1 *Two non-empty words commute iff they are powers of the same word:*

$$x, y \in A^+, x.y = y.x \Leftrightarrow \exists z \in A^*, n, m > 0 \text{ such that } x = z^n \text{ and } y = z^m$$

And more precisely, the set of words commuting with a word $x \in A^+$ is a monoid generated by a single primitive word z:

$$\forall x \in A^+, \exists z \in A^+, z \text{ primitive such that } z^* = \{y \in A^* | x.y = y.x\}$$

The following corollary is a direct consequence of theorem 1

Corollary 1 *Let $x, y \in A^*$, we have:*

$$(\exists m > 0, y.x^m = x^m.y) \Longleftrightarrow (\forall n > 0, y.x^n = x^n.y)$$

Proof. The implication from right to left is straightforward. Let us see implication from left to right : If $x = \epsilon$ or $y = \epsilon$ it is trivial. Otherwise, by theorem 1, the set of words commuting with y is z^* for a primitive word z. Thus $x^m \in z^*$ and then by proposition 1, $x \in z^*$. And finally, $\forall n, y.x^n = x^n.y$.

2.1.2 The considered transition systems

We consider a restrictive form of *transition systems* [Kel72, KM82] general enough to represent the behaviour of CFSMs, Fifo-Nets and, with a few modifications, Estelle specifications. The main reasons why it is possible is that their behaviours can be described in an operational way and that they allow a fifo mechanism on channel contents.

A transition system will be a 4-uple $\mathcal{S} =< GS, T, \rightarrow, S_0 >$ where :

- GS is the set of *global states*,
- T is a finite set of *transitions*,
- the *transition function* $\rightarrow$ is a partial function from $GS \times T$ to GS,
- $S_0 \in GS$ is the initial state.

A global state $S \in GS$ is a $N + M$-uple

$$S =< E_1(S), \ldots E_N(S), C_1(S), \ldots, C_M(S) >$$

where $\forall i = 1 \ldots N, E_i$ is a function from GS to the finite set $LS_i = \{E_{0,i}, \ldots E_{n_i,i}\}$ of *local states*,
and $\forall j = 1 \ldots M, C_{f_j}$ is a function from GS to $M_{f_j}^*$ (elements of M_{f_j} are called *messages*). $C_{f_j}(S)$ is called the *channel content* of the channel f_j in the global state S. No restriction is made on the initial state $S_0 \in GS$.

A transition $t \in T$ is a $N + M$-uple

$$t =< \delta_1(t), \ldots, \delta_N(t), \chi_{f_1}(t), \ldots, \chi_{f_M}(t) >$$

where the *local transition function* $\delta_i(t)$ is a partial function from LS_i to LS_i
and the *channel interaction* $\chi_{f_j}(t)$ consists in a couple $< \psi_{f_j}(t), \varphi_{f_j}(t) >$ where ψ_{f_j} and φ_{f_j} are functions from T to $M_{f_j}^*$ which extract the input and output streams of t in the channel f_j (a transition can be composed of several inputs and outputs).

We then define the transition function $\rightarrow$ which gives a fifo mechanism to channel contents. Let $S \in GS$ be a global state and $t \in T$ a transition. Then t is *fireable* in S if and only if :

- $\forall i = 1 \ldots N, \delta_i(t)(E_i(S))$ is defined,
- $\forall j = 1 \ldots M, \psi_{f_j}(t) \leq C_{f_j}(S)$

This transition then leads to a state S' such that :

- $\forall i = 1 \ldots N, E_i(S') = \delta_i(t)(E_i(S))$,
- $\forall j = 1 \ldots M, \psi_{f_j}(t).C_{f_j}(S') = C_{f_j}(S).\varphi_{f_j}(t)$.

We then write : $S \xrightarrow{t} S'$.

The transition function $\rightarrow$ is extended to transition sequences of T^* by :
$S_1 \xrightarrow{t_1 \ldots t_n}{}^* S_{n+1}$ iff there exist $S_2, \ldots S_n$ such that $\forall i = 1, \ldots n, S_i \xrightarrow{t_i} S_{i+1}$.

The two functions ψ_f and φ_f are extended into morphisms of monoids from T^* to M_f^*.

A global state $S \in GS$ is said to be *reachable* (from the initial state S_0) if there exists a transition sequence $w \in T^*$ such that $S_0 \xrightarrow{w}{}^* S$.

We define a partial ordering on global states which generalize the prefix ordering on words by :

Definition 2 *S is a generalized prefix of S' denoted by $S \leq_g S'$ iff all the local states of S are identical to those of S', and channels contents in S are prefix (on words) of those of S'.*

$$S \leq_g S' \Longleftrightarrow (\forall i = 1 \ldots N, E_i(S) = E_i(S')) \text{ and } (\forall j = 1 \ldots M, C_{f_j}(S) \leq C_{f_j}(S'))$$

We will use the following set for generalized prefix ordered global states :

Definition 3 *If S is a generalized prefix of S' we define the set $Strict(S, S')$ as the subset of strictly increasing channels, that is :*

$$\text{if } S \leq_g S', \text{ then } Strict(S, S') = \{f \text{ such that } C_f(S) < C_f(S')\}$$

The whole behaviour of a transition system can be represented as a directed possibly infinite *reachability tree* $RT(\mathcal{S})$:

- the nodes s are labelled by the reachable global states S. We will use small letters for nodes and capital letters for global states.
- the root s_0 is labelled by the initial global state S_0
- the edges are labelled by the fireable transitions of the system. $tr(s, s')$ will denote the transition sequence from node s to node s' in the reachability tree.

The *reachability graph* $RG(\mathcal{S})$ is obtained from $RT(\mathcal{S})$ by identifying all the nodes labelled by the same global state. Thus its set of nodes represents the set of reachable global states.

The *language of the system* $L(\mathcal{S})$ is the set of transition sequences from S_0.

The *input language* $L_I(f)$ of a fifo f is the set of message sequences entering channel f i.e. $L_I(f) = \varphi_f(L(\mathcal{S}))$

2.1.3 Unboundedness

Definition 4 *A fifo channel f is bounded iff there exists a constant K_f such that, for every reachable global state, the content of f is of length less than K_f.*

Kœnig's lemma asserts that every infinite tree of finite degree has an infinite branch. The reachability tree is of finite degree, thus Kœnig lemma applies. If $RG(\mathcal{S})$ is infinite then $RT(\mathcal{S})$ too. So, there exists at least an infinite sequence in $RT(\mathcal{S})$. If the nodes of each infinite sequence of $RT(\mathcal{S})$ are labelled by a finite number of different states, then $RG(\mathcal{S})$ is finite. Thus $RG(\mathcal{S})$ is infinite if and only if there exists an infinite sequence of nodes of $RT(\mathcal{S})$ labelled by an infinite number of different states.

In the considered transition system defined above, each local state can take a finite number of values and the only possibly unbounded objects are channels contents so we have the following proposition:

Proposition 2 *The reachability graph $RG(\mathcal{S})$ is finite if and only if all channels are bounded.*

2.2 Known results about CFSMs and Fifo-Nets

2.2.1 CFSMs

A CFSMs system [Boc78] is a N-uple $< P_1, \ldots, P_N >$ of CFSMs with a set of messages $M = \bigcup_{i,j=1}^{N} M_{ij}$.

A CFSM P_i is a 4-uple $< LS_i, T_i, \delta_i, q_{0_i} >$ where :

- LS_i is the finite set of local states.
- T_i is a finite set of transitions. A transition t is an output $-a, a \in M_{ij}$ or an input $+b, b \in M_{ji}$ or an internal transition e.
- δ_i is a partial function from $LS_i \times T_i$ to LS_i.
- $q_{0_i} \in LS_i$ is the initial state.

For each pair of CFSMs $(P_i, P_j), i \neq j$, there exists a fifo channel f_{ij} which allows the communication of messages M_{ij} between the two CFSMs.

The behaviour of such a system is described by a restrictive form of our transition system where a global state consists in the set of all the local states and the channel contents. The initial state consists in all the local initial states and empty channels. A global transition is an element $t \in T = \bigcup_{i=1}^{N} T_i$ so that the application of the transition function to t only modifies one CFSM and possibly one channel content. A transition is either an input or an output or an internal transition but cannot be a composition of them.

2.2.2 Fifo-Nets

A Fifo-Net [MM81, FR88] is a generalization of Petri-Nets in which places are replaced by fifo queues. Edges are labelled by words. A transition is fireable when, for each incoming edge of the transition, its label is a prefix of the starting fifo. The transition is then fired and labels of all the outcoming edges are added to the tail of the arriving fifo. It can also be seen as a generalization of CFSMs systems because every CFSMs system can easily be translated into a Fifo-Net.

2.2.3 Decidability results

In the formal framework of CFSMs, Brand and Zafiropulo proved the following theorem :

Theorem 2 *The unboundedness problem is undecidable for two communicating finite state machines.*

This result is a consequence of the capability for CFSMs to simulate Turing machines. The most frequently mentioned reference is [BZ83] but the complete proof is only available in the technical report [BZ81]. The inclusion of CFSMs systems in Fifo-Nets allows to state :

Corollary 2 *The unboundedness problem is undecidable for Fifo-Nets.*

In the same paper we can also find a test for boundedness. This test is a static one in the sense that it does not consider the reachable global states. After an analysis of local loops of each finite state machine, it gives a system of inequations on the number of messages in transit at any point of the computation. Solving this system can then give bounds to these numbers.

Some people also tried to find decidability results for limited classes of CFSMs systems. We discuss here the main results concerning this problem.

Let us give some definitions [CF87, Fin88]:

Definition 5 *A language $L \subset A^*$ is monogeneous iff it is included in a finite union of left factors of languages $u.v^*$, with $u, v \in M_f^*$.*

A language $L \subset A^*$ is linear iff it is included in a finite union of languages $a_1^*.a_2^* \ldots a_n^*$ where a_i's are different letters of A.

A CFSM is monogeneous (resp. linear) iff the input languages of all fifo channels are monogeneous (resp. linear)

We give here a summary of the main results concerning unboundedness decidability for CFSMs.

Unboundedness is undecidable for two CFSMs [BZ81, BZ83] and for 3 CFSMs even when, except for one fifo, all the others are 1-bounded over some tally language (that is $L_I(f) \subseteq a^*$, $a \in M_f$) [GGLR87].

Thus unboundedness is undecidable for general CFSMs and Fifo-Nets. However, there exists decidability results for restrictive classes.

- unboundedness is decidable for two CFSMs when :
  - one of the two fifos is bounded [BZ81, BZ83].
  - one CFSM is restricted to send only a single type of message [GR84, RY86].
  - the input language of one fifo is monogeneous [Fin86].
- unboundedness is decidable for CFSMs and Fifo-Nets when :
  - all the queues alphabets consist of a single message type (it is a subclass of Petri-Nets and thus the result is proved in [KM69])
  - all the input languages are linear [GGLR87, CF87]
  - all the input languages are monogeneous [Fin86]

2.3 Justifying tests for unboundedness

We saw in the preceding subsection that unboundedness is undecidable for CFSMs systems which are particular cases of the transition system defined in 2.1.2. Thus unboundedness is undecidable in the general framework of transition systems that we consider.

The first idea to avoid this is to work with restrictive models in which the problem is decidable. This gives the preceding results about CFSMs systems. Our approach is conceptually different for we want to find sufficient conditions of boundedness and unboundedness which can be applied to all specifications. This gives tests (or semi-decision procedure). We here justify why it seems more interesting than the first point of view.

First of all, the decidability results obtained for CFSMs and Fifo-Nets seem unsatisfactory. As a matter of fact, the only non trivial decidability result for an unlimited number of CFSMs and Fifo-Nets were found for linear and monogeneous systems. But these properties seem to be generally undecidable [Fin86]. Moreover, the decision procedures explicitely use the input languages. So they can only be used on systems for which monogeneous and linear languages including their input languages are known. In the case of linear systems, the generalization of the decision procedure for linear languages on words ($L_I(f) \subseteq \bigcup_{i=1}^{k} w_{i,1}^* \ldots w_{i,n_i}^*, w_{i,j} \in M_f^*$) does not seem realistic because the procedure [CF87] (where $w_{i,j}$ are letters) uniquely identify the channel content with the $w_{i,j}$'s of the linear language. If $w_{i,j}$'s are words, that seems impossible even if it is a code (because channels contents are subwords). To sum up, the above decision procedures seem to be too much restrictive because they need additional information on channels which are generally indecidable. But those classes of systems allow to decide other problems with the construction of a coverability tree.

A complementary approach would be to find sufficient conditions for boundedness and unboundedness which allows testing every possible specification with fifo channels.

We are looking for a condition on global states which can be evaluated on each reachable state during the construction of the reachability tree and only depending on the states of the current sequence. This leads to the construction of a *reduced tree* in which a sequence is stopped at the states satisfying the condition or when a loop is detected. This can be improved if we keep some of the completely visited states (which are no longer in the current sequence) and detect equality. This avoids regenerating some states without storing all the completely visited ones [JJ89].

Such a test will be practicable on every execution of the system as soon as the current transition sequence (or even a window of them) is kept during the computation. That means as well during a partial exploration of the reachability tree, as during an exhaustive one.

Another very important argument is that, in practice, testing such properties as unboundedness on possibly infinite trees gives the same service as decision procedures. In fact, decidability is merely theoretical because we must store an increasing number of objects (the states of the current sequence) with a bounded memory. Thus, both give one of the three results : the property is true, false or there is a memory overflow.

Our search for a test is in fact in the continuation of decidability results for linear and monogeneous systems [Fin87]. Both were based on the construction of a reduced reachability tree with an ordering $\ll$ on global states having three essential properties allowing decidability :

- monotonicity: if $S \xrightarrow{*} S'$ and $S \ll S', S \neq S'$, then for every transition t,

$$S \xrightarrow{t} S_1 \Rightarrow S' \xrightarrow{t} S_1', S_1 \ll S_1', S_1 \neq S_1'$$

- well-ordering: in every infinite sequence of different markings, there exists a strictly increasing subsequence,

- decidability of the ordering and equality : $S \ll S'$ and $S = S'$ are computable.

Monotonicity gives a sufficient condition and well-ordering assures that the algorithm stops. Thus for a test, decidability and monotonicity properties are sufficient.

3 The considered test

This section shows how we found a relation between global states by several steps of generalization of a simple idea and then proves that it is a sufficient condition for unboundedness. In fact, at each step of our refinements, we proved that the found relation was monotonic and we worked by feedback between the definition of the relation and the proof of monotonicity. Our concern in those successive definitions was always to detect unboundedness as soon as possible and for the largest class of unbounded systems.

3.1 Definition of a relation between global states

Studying results for monogeneous and linear systems, we observed that in both classes the decision procedure was based on an ordering $\ll$ included in the generalized prefix ordering. One would then be tempted to find an ordering included in the prefix order and having a monotonicity property. But the transitivity of the order is unnecessary and thus a relation is sufficient. It is why we have searched for a decidable relation Θ between reachable global states included in the generalized prefix ordering and having some kind of monotonicity property.

The first idea was that if a transition sequence w is repeated twice between three global states D, S and S' with a strict generalized prefix increasing between them, then we could infinitely repeat w and reach an infinite number of different global states. That gives a relation Ω defined between couples of nodes by :

$$\Omega(s,s') \text{ iff } \exists D \in GS, w \in T^* \text{ such that } S_0 \xrightarrow{*} D \xrightarrow{w} S \xrightarrow{w} S' \text{ and } D <_g S <_g S'$$

We then proved that : if $\Omega(s,s')$ and $S \neq S'$ then $RG(\mathcal{S})$ is infinite

We then noticed that the proof is still available if you replace w by two different powers of w :

$$\Omega(s,s') \text{ iff } \exists D \in GS, w \in T^*, k,l > 0 \text{ such that } S_0 \xrightarrow{*} D \xrightarrow{w^k} S \xrightarrow{w^l} S' \text{ and } D <_g S <_g S'$$

The third step was to observe that transition sequences are not the essential. The input and output parts of them are sufficient and they can be treated separately and for each channel. That gives :

$\Omega(s,s')$ iff there exists a global state $D \in GS$, and for all channel f, there exist two words $\psi, \varphi \in M_f^*$, and integers $k,l,k',l' > 0$ such that

$$\begin{array}{l} S_0 \xrightarrow{*} D \xrightarrow{v} S \xrightarrow{w} S' \\ \text{with } D <_g S <_g S' \\ \text{and } \varphi_f(v) = \varphi^k \text{ , } \varphi_f(w) = \varphi^l \\ \text{and } \psi_f(v) = \psi^{k'} \text{ , } \psi_f(w) = \psi^{l'} \end{array}$$

We then noticed that :

- no condition has to be required for non-increasing channels,
- output loops must be detected in order to stop earlier,
- the existence of an unique global state D is not necessary, we only need a particular one D_f for each channel f, and even $D_f <_g S$ is not necessary.
- input streams do not need to be powers of the same word, the suffix condition is sufficient.

That leads to the following final definition :

Definition 6 *Let s and s' be two nodes of the reachability tree corresponding to two global states S and S'.*

$$\Theta(s,s') \Longleftrightarrow \begin{cases} s \xrightarrow{w}{}^{*} s' \ (s' \text{ is reachable from } s \text{ by a transition sequence } w) \\ S \leq_g S' \ (S \text{ is a generalized prefix of } S') \\ \text{for every channel } f \in Strict(S,S'), \text{ we have } \Theta_f(S,S') \end{cases}$$

where for each such channel f, $\Theta_f(S,S')$ is defined by :

- *either $\psi_f(w) = \epsilon$ (no receipt on channel f between s and s')*
- *or $\psi_f(w) \neq \epsilon$ and there exist two words $\psi, \varphi \in M_f^*$, a global state D_f reachable from S_0 and strictly positive integers k, l (ψ, φ, D_f, k, l depend on f) such that*

$$S_0 \xrightarrow{u} D_f \xrightarrow{v} S \xrightarrow{w} S'$$
$$\varphi_f(v) = \varphi^k \text{ and } \varphi_f(w) = \varphi^l$$
$$\psi_f(w) = \psi.\psi_f(v)$$

By corollary 1 we can equivalently define $\Theta_f(S,S')$ by :

- $\psi_f(w) = \epsilon$ (no receipt on channel f) or
- $\psi_f(w) \neq \epsilon$ and there exist two words $\psi, \varphi \in M_f^*$ and a global state D_f (ψ, φ, D_f depend on f) such that

$$S_0 \xrightarrow{u} D_f \xrightarrow{v} S \xrightarrow{w} S'$$
$$\varphi_f(v).\varphi_f(w) = \varphi_f(w).\varphi_f(v)$$
$$\psi_f(w) = \psi.\psi_f(v)$$

The two above definitions of Θ_f are mathematically equivalent but the second one is more suited in the algorithm because it is easier to test commutation (by equality of two words) than to find a common generator. However we will rather use the first one in the proofs.

The generalization of the first idea presented in the beginning of the section to find the final expression of Θ made us see several intermediate relations which are particular cases of Θ. Thus if Θ gives a sufficient condition for unboundedness, the others too. They are certainly easier to compute, thus if Θ is too much expensive, any of its particular cases can be used.

We do not assert that Θ is the best relation we can find. Perhaps could we relax some conditions and keep the desired monotonicity property.

3.2 The relation is a sufficient condition for unboundedness

We now prove that Θ gives a sufficient condition for unboundedness. We need some kind of monotonicity property weakest than the one given in 2.3 that is : if two nodes s and s' satisfy $\Theta(s,s')$ and $S \neq S'$ then the transition sequence from s to s' is again fireable from s' and leads to a third node s'' such that $\Theta(s',s'')$ and $S' \neq S''$.

For that aim we need the following very important result :

Proposition 3 *If $S \leq_g S'$ and t is a fireable transition from S then t is fireable from S' and the local states of the two reached states are identical.*

i.e. if $S \xrightarrow{t} S_1$ and $S \leq_g S'$ then $S' \xrightarrow{t} S'_1$ and $\forall i, 1 \leq i \leq N$, $E_i(S_1) = E_i(S'_1)$.

Proof.

If t is fireable in S and leads to S_1 we have :

$$\begin{array}{l} \forall i = 1 \ldots N, \delta_i(t)(E_i(S)) = E_i(S_1), \\ \forall j = 1 \ldots M, \psi_{f_j}(t) \leq C_{f_j}(S), \\ \forall j = 1 \ldots M, \psi_{f_j}(t).C_{f_j}(S_1) = C_{f_j}(S).\varphi_{f_j}(t). \end{array}$$

We have $S \leq_g S'$ thus $\forall i = 1 \ldots N, E_i(S) = E_i(S')$ and $\forall j = 1 \ldots M, C_{f_j}(S) \leq C_{f_j}(S')$.

Therefore $\forall j = 1 \ldots M, \psi_{f_j}(t) \leq C_{f_j}(S')$ and $\forall i = 1 \ldots N, \delta_i(t)(E_i(S'))$ is defined. Then t is fireable in S' and leads to a state S'_1 such that

$$\forall i, E_i(S'_1) = \delta_i(E_i(S')) = \delta_i(E_i(S)) = E_i(S_1)$$

Warning : generally, we do not have $S_1 \leq_g S'_1$.

The monotonicity property is a consequence of the following lemma :

Lemma 1 *If $\Theta(s,s')$ and $S \neq S'$ then the transition sequence $tr(s,s')$ from s to s' is fireable in s' and reach a node s'' such that $S' <_g S''$.*

Proof : The transition sequence w from S to S' is a sequence $t_1.t_2 \ldots t_n$ of transitions. Assume S_i is the global state reached by the sequence $t_1 \ldots t_i$ ($S' = S_n$). The demonstration is inductive in i:

1. prove that t_1 is fireable in S' and reach a state S'_1 such that $S_1 <_g S'_1$ and $Strict(S_1, S'_1) = Strict(S, S')$,
2. suppose that for all $j < i$, $t_1 \ldots t_j$ is fireable in S' and reach a state S'_j such that $S_j <_g S'_j$ and $Strict(S_j, S'_j) = Strict(S, S')$. Then the transition t_i is fireable in S_{i-1} and reach a state S'_i such that $S_i <_g S'_i$ and $Strict(S_i, S'_i) = Strict(S_{i-1}, S'_{i-1})$.

$$S_0 \xrightarrow{u} D_f \xrightarrow{v} S \xrightarrow{t_1 \ldots t_{i-1}} S_{i-1} \xrightarrow{t_i} S_i \xrightarrow{t_{i+1} \ldots t_n} S' \xrightarrow{t_1 \ldots t_{i-1}} S'_{i-1} \xrightarrow{t_i} S'_i$$

1. Proposition 3 implies that t_1 is fireable in S' and $\forall i, E_i(S_1) = E_i(S'_1)$.

   We must now prove :

$$\forall f, [C_f(S) = C_f(S') \Rightarrow C_f(S_1) = C_f(S'_1)] \wedge [C_f(S) < C_f(S') \Rightarrow C_f(S_1) < C_f(S'_1)]$$

- For every channel $f \notin \mathit{Strict}(S, S')$ (i.e. $C_f(S) = C_f(S')$).
  We have $\psi_f(t_1).C_f(S'_1) = C_f(S').\varphi_f(t_1)$ and $\psi_f(t_1).C_f(S_1) = C_f(S).\varphi_f(t_1)$.
  Therefore $C_f(S'_1) = C_f(S_1)$.
- For every channel $f \in \mathit{Strict}(S, S')$.
  - if $\psi_f(w) = \epsilon$ then $\psi_f(t_2 \dots t_n.t_1) = \epsilon$ because it is a circular permutation of $\psi_f(w)$. The evolution of channel contents between S_1 and S'_1 gives :

    $$C_f(S'_1) = C_f(S_1).\varphi_f(t_2 \dots t_n.t_1)$$

    We have $\varphi_f(w) \neq \epsilon$ ($f \in \mathit{Strict}(S, S')$) thus $\varphi_f(t_2 \dots t_n.t_1) \neq \epsilon$ and then

    $$C_f(S_1) < C_f(S'_1)$$

  - otherwise $\psi_f(w) \neq \epsilon$. We know that $C_f(S) < C_f(S')$. Then there exists a nonempty word $Q \in M_f^*$ s.t. :

    $$C_f(S') = C_f(S).Q \tag{1}$$

    The evolution of channel contents between S and S_1 and between S' and S'_1 gives the equations :

    $$\begin{aligned} \psi_f(t_1).C_f(S_1) &= C_f(S).\varphi_f(t_1) \\ \psi_f(t_1).C_f(S'_1) &= C_f(S').\varphi_f(t_1) \\ &= C_f(S).Q.\varphi_f(t_1) \end{aligned}$$

    Therefore

    $$C_f(S_1) < C_f(S'_1) \Longleftrightarrow \varphi_f(t_1) < Q.\varphi_f(t_1)$$

    The equations of channel contents evolution are :

    $$\begin{aligned} \psi_f(u).\psi_f(v).C_f(S) &= C_f(S_0).\varphi_f(u).\varphi_f(v) \\ \psi_f(w).C_f(S') &= C_f(S).\varphi_f(w) \end{aligned}$$

    We have $\psi_f(u) < C_f(S_0).\varphi_f(u)$ so, if we note $\delta = \psi_f(u)^{-1}.(C_f(S_0).\varphi_f(u))$, the preceding equations can be written :

    $$\psi_f(v).C_f(S) = \delta.\varphi^k \tag{2}$$
    $$\psi_f(w).C_f(S') = C_f(S).\varphi^l \tag{3}$$

    We now prove $Q.\varphi^k = \varphi^k.Q$ using the three equations (1), (2) and (3) on channel contents. The equalities are indexed by the equation used.
    We have

    $$\begin{aligned} \psi.\psi_f(v).C_f(S).Q.\varphi^k &\overset{1}{=} \psi.\psi_f(v).C_f(S').\varphi^k \\ &\overset{3}{=} C_f(S).\varphi^{l+k} \end{aligned}$$

Thus

$$\begin{aligned} Q.\varphi^k &= (C_f(S).\varphi^{l+k})[|\psi.\psi_f(v).C_f(S)| + 1 \ldots |\psi.\psi_f(v).C_f(S)| + |Q| + k|\varphi|] \\ &= (\varphi^{l+k})[|\psi| + |\psi_f(v)| + 1 \ldots |\psi| + |\psi_f(v)| + |Q| + k|\varphi|] \end{aligned}$$

We also have

$$\begin{aligned} \psi_f(v).\psi.\delta.\varphi^k.Q &\stackrel{2}{=} \psi_f(v).\psi.\psi_f(v).C_f(S).Q \\ &\stackrel{1}{=} \psi_f(v).\psi.\psi_f(v).C_f(S') \\ &\stackrel{3}{=} \psi_f(v).C_f(S).\varphi^l \\ &\stackrel{2}{=} \delta.\varphi^{k+l} \end{aligned}$$

Thus

$$\begin{aligned} \varphi^k.Q &= (\delta.\varphi^{k+l})[|\psi_f(v).\psi.\delta| + 1 \ldots |\psi_f(v).\psi.\delta| + k|\varphi| + |Q|] \\ &= (\varphi^{k+l})[|\psi_f(v)| + |\psi| + 1 \ldots |\psi_f(v)| + |\psi| + k|\varphi| + |Q|] \end{aligned}$$

Thus $Q.\varphi^k = \varphi^k.Q$ and by corollary 1 we have

$$\forall i, Q.\varphi^i = \varphi^i.Q \tag{4}$$

In particular $Q.\varphi^l = \varphi^l.Q$, that is $Q.\varphi_f(t_1 \ldots t_n) = \varphi_f(t_1 \ldots t_n).Q$
And therefore

$$C_f(S_1) < C_f(S'_1)$$

2. By proposition 3, $S_{i-1} <_g S'_{i-1}$ implies that t_i is fireable in S'_{i-1} and leads to a global state S'_i such that $\forall j, E_j(S_i) = E_j(S'i)$. We must now prove :

$$\forall f, [C_f(S) = C_f(S') \Rightarrow C_f(S_i) = C_f(S'_i)] \wedge [C_f(S) < C_f(S') \Rightarrow C_f(S_i) < C_f(S'_i)]$$

- For every channel $f \notin Strict(S, S') = Strict(S_{i-1}, S'_{i-1})$.
  We have $\psi_f(t_i).C_f(S_i) = C_f(S_{i-1}).\varphi_f(t_i)$ and $\psi_f(t_i).C_f(S'_i) = C_f(S'_{i-1}).\varphi_f(t_i)$.
  Therefore

$$C_f(S'_i) = C_f(S_i)$$

- For every channel $f \in Strict(S, S')$.
  - if $\psi_f(w) = \epsilon$ then $\psi_f(t_{i+1} \ldots t_n.t_1 \ldots t_i) = \epsilon$ (circular permutation of $\psi_f(w)$).
    The evolutions of channel contents gives the following equation :

$$C_f(S'_i) = C_f(S_i).\varphi_f(t_{i+1} \ldots t_n.t_1 \ldots t_i)$$

    Now $C_f(S) < C_f(S')$, thus $\varphi_f(w) \neq \epsilon$ implies $\varphi_f(t_{i+1} \ldots t_n.t_1 \ldots t_i) \neq \epsilon$ and then :

$$C_f(S_i) < C_f(S'_i)$$

– otherwise $\psi_f(w) \neq \epsilon$. We want to prove that $C_f(S_i) < C_f(S'_i)$ knowing that $\forall j < i, C_f(S_j) < C_f(S'_j)$.
In particular we still have the equations (1), (2) and (3) which give the result :

$$\forall i, Q.\varphi^i = \varphi^i.Q \qquad (4)$$

Thus, the evolution of channel contents between S and S_i and between S' and S'_i gives :

$$\begin{aligned} \psi_f(t_1 \dots t_i).C_f(S_i) &= C_f(S).\varphi_f(t_1 \dots t_i) \\ \psi_f(t_1 \dots t_i).C_f(S'_i) &= C_f(S').\varphi_f(t_1 \dots t_i) \\ &\stackrel{1}{=} C_f(S).Q.\varphi_f(t_1 \dots t_i) \end{aligned}$$

Thus

$$C_f(S_i) < C_f(S'_i) \Longleftrightarrow \varphi_f(t_1 \dots t_i) < Q.\varphi_f(t_1 \dots t_i)$$

Now result (4) with $i = l$ gives :

$$Q.\varphi_f(t_1 \dots t_n) = \varphi_f(t_1 \dots t_n).Q$$

From which follows :

$$\varphi_f(t_1 \dots t_i) < Q.\varphi_f(t_1 \dots t_i)$$

And then $C_f(S_i) < C_f(S'_i)$

Theorem 3 *Let s and s' be two nodes of the reachability tree $RT(\mathcal{S})$ corresponding to two global states S and S'.*

$$\Theta(s, s'),\ and\ S \neq S'] \Longrightarrow [s' \stackrel{tr(s,s')}{\longrightarrow} s''\ and\ \Theta(s', s'')\ and\ S' \neq S'']$$

Applying this again, we can build an infinite number of different global states :

$$(\Theta(s, s')mboxandS \neq S') \Longrightarrow RG(\mathcal{S})\ is\ infinite$$

Proof. It is a consequence of the preceding lemma 1. Suppose that $\Theta(s, s')$ and $S \neq S'$. Lemma 1 asserts that the transition sequence w from s to s' is again fireable in S' and reach a node s'' such that $S' <_g S''$.

We are then in the situation :

$$S \stackrel{w}{\longrightarrow} S' \stackrel{w}{\longrightarrow} S'' \text{ and } S <_g S' <_g S''$$

which is a particular case of $\Theta(s', s'')$ and $S' \neq S''$.

Figure 1: The unboundedness test algorithm

```
type
   gl_st_tp = record
         loc_st: array[1..N] of local_state_tp; — array of local states —
         q_cont: array[1..M] of queue_content_tp; — array of queue contents —
         end;
   cur_seq_tp = ↑ elt_cur_seq_tp;
   elt_cur_seq_tp = record
         gl_st: gl_st_tp;
         fireable: set of trans_tp; — set of fireable transitions not yet fired —
         succ: cur_seq_tp;
         end;
   seq_φψ_tp =↑ elt_seq_φψ_tp
   elt_seq_φψ_tp= record
         φ, ψ: array[1..M] of word;
         succ: seq_φψ_tp;
         end;
var
   bounded, unbounded, saturated: boolean init false;

function fireable_trans ( g_s: gl_st_tp): set of trans_tp;
      — returns the set of fireable transitions from g_s —

function succ (g_s: gl_st_tp; t: trans_tp): gl_st_tp;
      — returns the global state reached from g_s after the fireing of t —

function pick_off ( var f_t: set of trans_tp): trans_tp;
      — chooses and extracts one transition from the set f_t of fireable transitions —

function static_verif_bounded: boolean;
      — performs a static test of boundedness described in [BZ81]—

procedure add_or_replace ( g_s: gl_st_tp; Mem: set of gl_st_tp);
      — if Mem is full, replaces at random a state in Mem by the state g_s —
      — else adds state g_s in Mem —

procedure test_Θ ( S': gl_st_tp, c_seq: cur_seq_tp, s_φψ: seq_φψ_tp, var lp, unbd: boolean );
      — unbd:= (∃S ∈ c_seq s.t. Θ(S, S') ∧ S ≠ S') — unboundedness detected —
      — lp:= (∃S ∈ c_seq s.t. S = S') — loop detected —
```

Figure 2: The unboundedness test algorithm (continuation)

```
procedure test_unbounded ( unbounded, saturated: boolean);
  var
    memory: set of gl_st_tp init ∅;
    — contains some already visited states —
    cur_seq: stack_tp init nil;
      — stack of ( state, not yet fired transitions) of the current sequence from S0 —
    seq_φψ: seq_φψ_tp init nil;
      — stack of input and output streams of the current transition sequence—
    cur_state: gl_st_tp; — current state —
    t: trans_tp; — just fired transition —
  begin
    cur_seq:= push(S0, fireable_trans(S0));
    while cur_seq ≠ nil and ( ¬unbounded and ¬saturated) do
      if cur_seq ↑ .fireable ≠ ∅ then begin
        t:= pick_off(cur_seq ↑ .fireable);
        cur_state:= succ(cur_seq ↑ .gl_st, t);
        seq_φψ:=push_φψ( seq_φψ, {φf(t), ψf(t)}1≤f≤M);
        test⊖ (cur_state, cur_seq, seq_φψ, loop, unbounded);
        if ¬(loop or unbounded) then
          if cur_state ∉ memory then begin
            saturated:= (|cur_seq| = max_seq_length);
            if ¬saturated then
              cur_seq:= push( cur_state, fireable_trans(cur_state)
            else pop_φψ(seq_φψ);
            end
          else pop_φψ(seq_φψ) — if (cur_state ∈ memory) —
        else pop_φψ(seq_φψ); -- if (loop or unbounded) —
        end
      else begin — cur_seq ↑ .fireable = ∅, all successors of cur_seq ↑ .gl_st are visited —
        add_or_replace(cur_seq ↑ .gl_st, memory);
        pop(cur_seq); pop_φψ(seq_φψ);
        end;
    end;

  main
    begin
      bounded:=static_verif_bounded;
      if ¬bounded then
        test_unbounded ( unbounded, saturated);
      if ¬(saturated or unbounded) then bounded:= true;
    end.
```

3.3 An algorithm for unboundedness test

An algorithm performing the unboundedness test is described in figure 1 and 2. It uses a depth first strategy for the construction of the reachability tree. The algorithm takes into account the following remarks :

- before testing for unboundedness, we can use a static boundedness test, for example the one proposed in [BZ81].
- when reaching a new state S', testing the unboundedness condition or the existence of a loop consists in finding a preceding state S of the current sequence such that $\Theta(s, s')$ or $S = S'$. Thus we only need to know the current transition sequence and the states from the root. We can then use a depth first strategy in which we only keep trace of this sequence in a stack. However, a breadthwise strategy can also be used and is theoretically better. It would avoid us traversing an infinite sequence where our test is always false whereas a neighbouring one would satisfy it. But this strategy requires to store the whole part of the graph that is already visited. Thus this strategy allows us to analyse smaller systems than with a depth first one.
- In order to avoid retraversing states we can also use a memory in which we record completely visited states (all their successors have been visited) and check for equality. This memory is of course bounded and the state graph may be infinite, so, when this memory will overflow we must replace states. The best strategy seems to be random replacement [Hol87]. So, every time we pop a completely visited state from the current sequence, if the memory is full, we take off a state at random and replace it with the new one [JJ89].
- we already saw that the definition of Θ_f can be written in two different ways and we noticed that the second definition (using commutation rather than power of the same word) is easier to compute.
- for the computation of Θ_f we do not need to know transitions but we only need the values of $\varphi_f(t)$ and $\psi_f(t)$ for each queue f and each transition t.

3.4 Complexity

We cannot really speak of a complexity for such an algorithm because it is only a test and then the time required for an answer is theoretically infinite. However we could give, for finite state graphs, the increase of time needed for the test. But it is a very difficult problem and a theoretical complexity could not lay enough stress on the link between the graph structure and the time needed. The best measurement of its efficiency is to know, for real specifications having an infinite number of states, whether the test detects unboundedness before memory overflow or not. This can only be observed if we include the test in a real verification tool.

When you reach a new node s' you theoretically need to visit all the previous nodes s until $s = s_0$ or you find one such that $\Theta(s, s')$. Thus, when the sequence grows the number of nodes to visit is also growing. The time needed to verify $\Theta(s, s')$ at each newly reached

node is not bounded. However, you can avoid this if you only test Θ on a window of nodes of the current sequence. The size of the window which is chosen must depend on each specification. The increase of time requested for the test for each node is then bounded by a constant.

4 Application

4.1 Coverability of Θ as a necessary and sufficient condition

In this section, we try to define the largest class of transition systems in which the relation Θ gives a decision procedure for unboundedness (remember that it is a theoretical decidability depending on the size of the system).

We must distinguish two cases according as we use a depth first or a breadthwise strategy :

1. if a depth first strategy is used we must have the property : *in all infinite sequence of nodes of the reachability tree there must exist two nodes s and s' such that $\Theta(s, s')$.*

2. if a breadthwise strategy is used the weakest following property is necessary : *if the reachability set is infinite, there must exist two nodes s and s' of the reachability tree such that $\Theta(s, s')$ and $S \neq S'$.*

Thus if the depth first strategy provides decidability then the breadthwise one too. Thus the largest class of transition systems for which Θ provides a decision procedure is the largest class for which the breadthwise strategy does.

However, we will see that a depth first strategy is sufficient for a class of transition systems including linear and monogeneous systems. This class is that of transition systems where all input languages are included in finite unions of left factors of linear languages on words. That is :

$$\text{for all channel } f, L_I(f) \subseteq \bigcup_{i=1}^{K} LF(w_{i,1}^*.w_{i,2}^*.\ldots.w_{i,N_i}^*) \text{ where } w_{i,j} \text{ are words of } M_f^*$$

We call it the class of transition systems linear on words. The inclusion of linear and monogeneous systems in that class is straightforward. For that class we prove that a depth first strategy decides unboundedness :

Theorem 4 *In all infinite transition sequence of the reachability tree of transition system linear on words, there exist two nodes s and s' such that $\Theta(s, s')$.*

Proof.

From every infinite sequence of reachable global states we can extract a finite number of infinite subsequences such that the local states in all these global states are identical (because the number of local states is finite). Let $(S_n)_n$ be one of them.

Let f be a channel with an input language defined as above. For the given sequence $\{S_n\}_n$, there exists a particular index i (not necessarily unique) such that the input stream during this sequence is included in $LF(w_{i,1}^*.w_{i,2}^*.\ldots.w_{i,N_i}^*)$

The behaviour of fifo channels implies that the states of that sequence have channel contents of the form :

$$C_f(S_n) \in \bigcup_{j=1}^{N_i}(\bigcup_{k=j}^{N_i}[(RF(C_f(S_0).w_{i,j}).w_{i,j}^*.\ldots.w_{i,k}^*.LF(w_{i,k})] \cup LF(RF(C_f(S_0).w_{i,j})))$$

Then, from sequence $\{S_n\}_n$ we can extract a subsequence $\{S_m\}_m$ such that there exist indexes $j,k, 1 \leq j \leq k \leq N_i$, sequences $\{r_{j,m}\}_m, \ldots \{r_{k,m}\}_m$ of positive integers, two constant words $c \in RF(LF(C_f(S_0).w_{i,j}))$ and $d \in LF(w_{i,k})$ such that

$$C_f(S_m) = c.w_{i,j}^{r_{j,m}}.\ldots.w_{i,k}^{r_{k,m}}.d$$

The form of the input language implies that if we have begun to send some $w_{i,l}$ in f then you can no more send the preceding ones. That is, if there exists an index m_1 and $l_1, j \leq l_1 \leq k$ such that $r_{l_1,m_1} \neq 0$, then from m_1, all sequences $\{r_{l,m}\}_m$ with $l < l_1$ are stationnary.

Thus there are two possible cases :

- from an index m_1 all sequences $\{r_{l,m}\}_m$ are stationnary and then $\forall m \geq m_1, C_f(S_m) = C_f(S_{m+1})$.
- otherwise, there exists a sequence $\{r_{l_1,m}\}_m$ from which we can extract a strictly increasing infinite subsequence $\{r_{l_1,p}\}_p$. According to the above remark, we have :

$$\begin{array}{lll} \forall l > l_1, \forall p & , & r_{l,p} = 0 \text{ (otherwise, } r_{l_1,p} \text{ would be stationnary)} \\ \forall l < l_1, \forall p & , & \{r_{l,p}\}_p \text{ is stationnary} \end{array}$$

The corresponding $\{C_f(S_p)\}_p$ are then strictly increasing for the prefix ordering and can be written :

$$C_f(S_p) = c.w_{i,j}^{c_j} \ldots w_{i,l_1-1}^{c_{l_1-1}}.w_{i,l_1}^{r_{l_1,p}}.d$$

where $c_j, \ldots, c_{l_1-1}$ are constants and $\{r_{l_1,p}\}_p$ is strictly increasing.

- if $j \neq l_1$ (and $c_j \neq 0$) there cannot exist any receipt on channel f in the transition sequence. Thus $\forall p, \Theta_f(S_p, S_{p+1})$.
- otherwise $j = l_1$ and we can write :

$$C_f(S_p) = c.w_{i,j}^{r_{j,p}}.d$$

with $c \in RF(LF(C_f(S_0).w_{i,j}))$, $d \in LF(w_{i,j})$ and $\{r_{j,p}\}_p$ strictly increasing. Let us note $\varphi_p = \varphi_f(tr(S_p, S_{p+1}))$ and $\psi_p = \psi_f(tr(S_p, S_{p+1}))$ ($tr(S_p, S_{p+1})$ is the transition sequence from S_p to S_{p+1}).

* if the number of p such that $\psi_p \neq \epsilon$ is finite then, starting from an index p_1, all ψ_p are equal to ϵ and then

$$\forall p \geq p_1, \Theta_f(S_p, S_{p+1})$$

* otherwise, there is an infinite number of p such that $\psi_p \neq \epsilon$.
  Thus there is a moment where all $C_f(S_0)$ has been received, so we can suppose that $c \in RF(w_{i,j})$ (let b be the word such that $b.c = w_{i,j}$). We also have $d \in LF(w_{i,j})$ thus there exists a word e such that $d.e = w_{i,j}$. The form of $C_f(S_p)$ implies that $\varphi_p = (e.d)^{l_p}$ and $\psi_p = (c.b)^{k_p}$ with $l_p > 0$, $k_p \geq 0$ and $l_p - k_p = r_{p+1} - r_p > 0$.
  - if $k_p = 0$ then $\Theta_f(S_p, S_{p+1})$
  - otherwise $\exists q > p$ such that $k_p + \ldots + k_q \geq k_{p-1}$ (there is an infinite number of non-null k_q), and then $\psi_p \ldots \psi_q = (c.b)^{k_p+\ldots+k_q-k_{p-1}}.\psi_{p-1}$. We also have $\varphi_{p-1} = (e.d)^{l_{p-1}}$ and $\varphi_p \ldots \varphi_q = (e.d)^{l_p+\ldots+l_q}$ which proves $\Theta_f(S_p, S_q)$.

Then, for a given channel f we can find a subsequence $\{S_q\}_q$ of $\{S_n\}_n$ such that $\{C_f(S_q)\}_q$ is stationnary or $\Theta_f(S_q, S_{q+1})$.

We can then begin again with that subsequence and an other channel f', and so on. The number of channels is finite thus we finally find a subsequence $\{S_r\}_r$ such that two consecutive elements satisfy $\Theta(S_r, S_{r+1})$. And therefore the theorem is satisfied.

4.2 Examples

This section gives some examples of CFSMs systems producing infinite reachability trees. They were all tested with a prototype of the algorithm presented above.

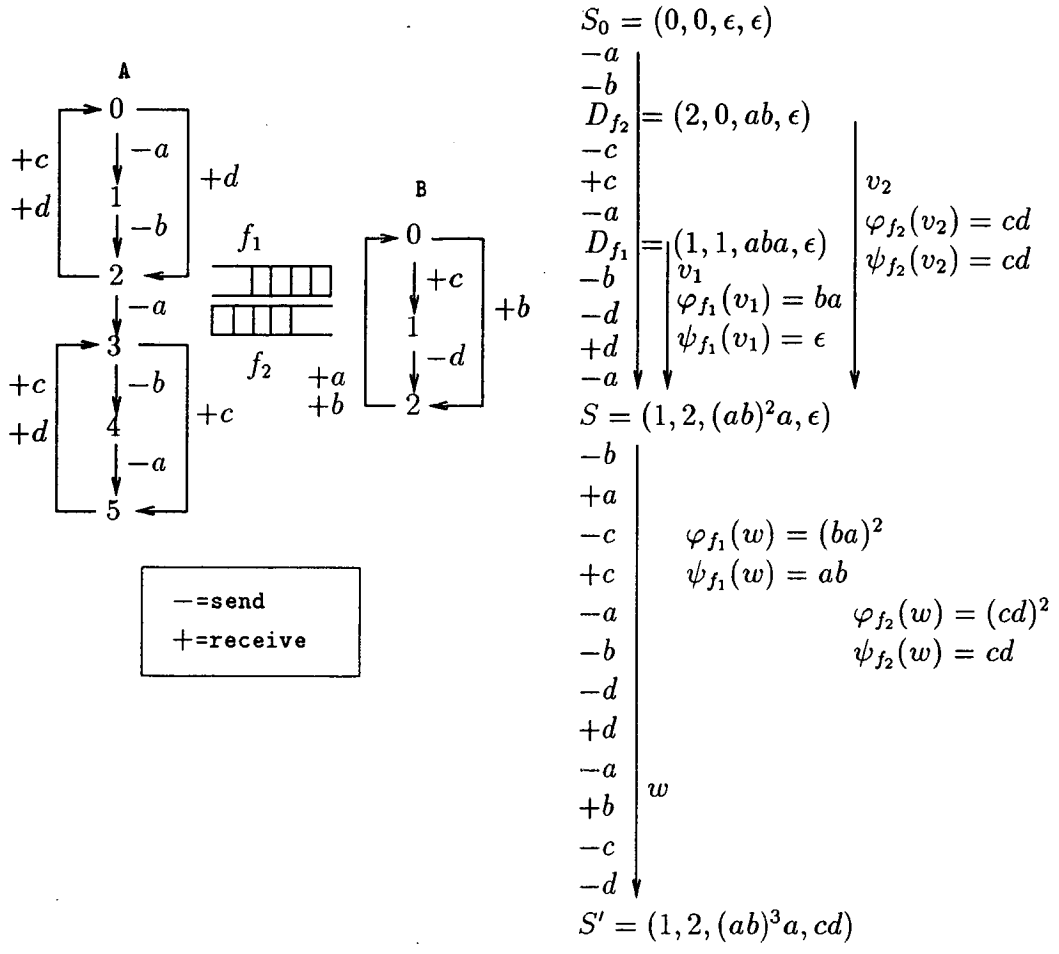
The first example (see figure 3) is a system $\mathcal{S}_1$ of two CFSMs such that one of the input languages is linear on words ($L_I(f_1) \subseteq LF(ab)^*.a.(ba)^*$) and the other one is monogeneous ($L_I(f_2) \subseteq LF(cd)^*$). The test can then decide unboundedness or not. The figure shows an example of transition sequence on which unboundedness was detected. When reaching $S' = (1, 1, (ab)^3a, cd)$ we find a state $S = (1, 2, (ab)^2a, \epsilon)$ which is a strict prefix of S' and satisfying $\Theta_{f_1}(S, S')$ and $\Theta_{f_2}(S, S')$. We can observe that the states D_{f_1} and D_{f_2} of the definition are not the same for the two channels.

The CFSMs system $\mathcal{S}_2$ of figure 4, which infinite state graph is partially represented in figure 5, is detected unbounded by the above procedure. The input language of f_1 is linear on words ($L_I(f_1) \subseteq LF(ab)^*.a.(ab)^*$) and $L_I(f_2) \subseteq c^*$. It is why its reduced tree is finite (in all infinite sequence there exist two states satisfying the test).

The CFSMs system $\mathcal{S}_3$ of figure 6, which produces an infinite reachability tree, has been specially designed to fail our procedure. There exists an infinite transition sequence that can be written $w.v.u^2.v^2.\ldots.u^{2^n}.v^{2^n}\ldots$ traversing an infinite number of increasing states (for the generalized prefix ordering) but the relation Θ is always false. We think that this kind of "exponential growing" cannot be detected by our procedure. However, it is possible to build a specialized procedure which could detect this behaviour. But it is very tricky and its coverability as a decision procedure would be very restricted.

Consider the system $\mathcal{S}_4$ obtained from $\mathcal{S}_3$ by identifying the two states 1 and 2 of the first CFSM. Its reachability tree contains the one of $\mathcal{S}_3$. Thus there are infinite sequences in which the test is always false. But there are some sequences in which the test is satisfied. For example the sequence $-a - c + c - a - d + d - a + a$ reach the state $S' = (1, 0, aa, \epsilon)$ and $S = (1, 0, a, \epsilon)$ is a strict prefix of S' and we have $\Theta(s, s')$.

Figure 3: An example of the use of the test



4.3 Application to Estelle specifications

4.3.1 The Estelle model

The Estelle language is a Formal Description Technique (FDT), now standardized by ISO and well suited for the specification of distributed systems. We present here the restrictions of static Estelle and their consequences on its semantics. Our aim is not to describe the Estelle language, therefore refer to [ISO89] for further informations.

A standard Estelle specification is a hierarchical structure of instances of *modules* communicating by messages through bidirectionnal links betweeen their *interaction points*. A module instance is an automaton extended with Pascal. To each interaction point is associated a fifo queue (which may be shared by several interaction points of the same module). In standard Estelle, the hierarchy of modules and links can change over time.

However, most of verification tools cannot accept dynamical change of topology. Thus, we can restrict to a static part of the Estelle language. The restrictions are the following:

Figure 4: A simplified connect-disconnect protocol

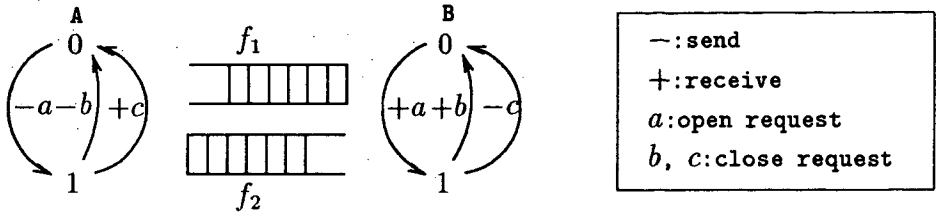


Figure 5: Its infinite state graph

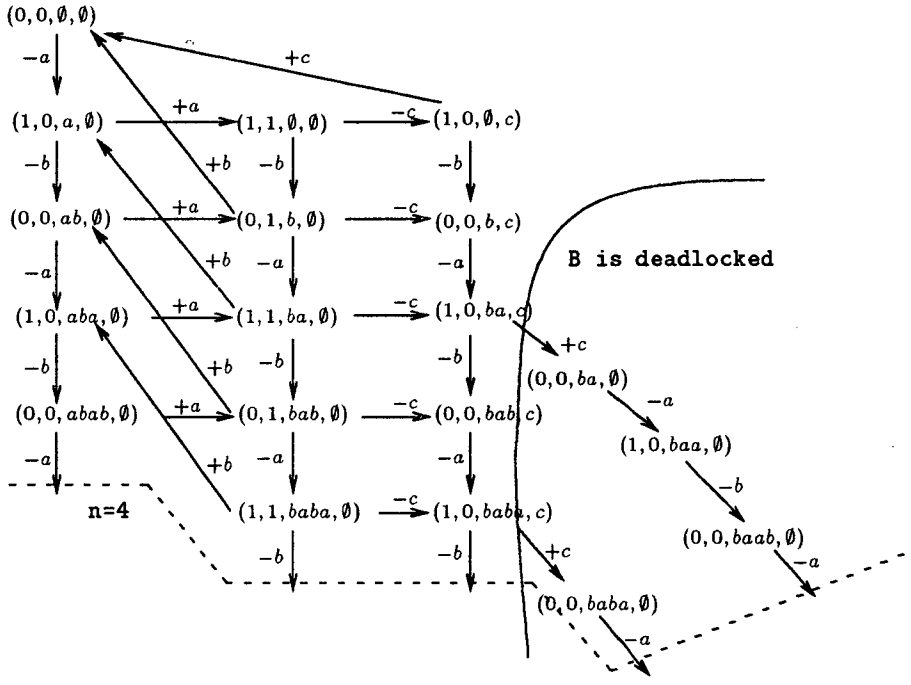
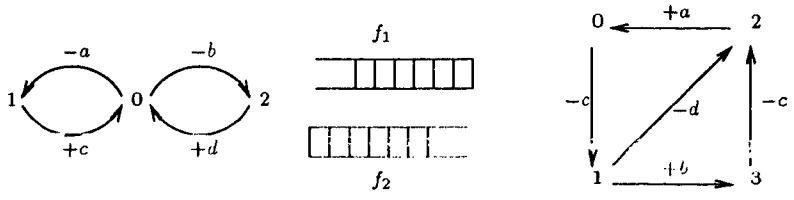


Figure 6: An unbounded system not detected as unbounded



- only one initialization part in each module.
- active modules (those who have transitions) cannot have submodules. This implies that there cannot have any release or creation of modules and links in transitions.
- all Pascal variables are bounded (in particular the use of dynamical structures is forbidden).

The restriction to static Estelle implies that the hierarchical structure is a tree of module instances in which only leaves are *active* (can have transition-parts) and thereby can be reduced to this set of active modules.

Each module instance is then a finite state automaton which states are composed of a control-part (associated with the keyword *state*), and an internal-data-part represented by values of Pascal variables. Each local transition is supposed to be atomic and is composed of a pre-condition and an action which modifies variables and output messages. The pre-condition may consist in :

- a *when* clause which is true iff the first message in a given queue corresponds to the argument and which opens visibility to message parameters,
- a *provided* clause which has a boolean value depending on the message parameters or Pascal variables,
- a *from* clause, true iff the actual control-state is the one given by the keyword *from*

The transition is *enabled* if and only if the three preceding clauses are all evaluated to true. If *priority* and *delay* clauses are used, the *fireable* transitions are chosen among the *enabled* ones with respect to priority and delay values. If we restrict the use of these clauses, *enabled* and *fireable* are identical.

The semantics of the complete Estelle specification is defined in an operational way in terms of transformations of a *global state*. A global state is the union of all the local states of the module instances with the channel contents. Several levels of parallelism can be described. Their semantics can be sequential non-deterministic, synchronous or fully asynchronous. No assumption can be made on the relative speed of modules so, in order to model the behaviour of the specification, all the possible interleavings of local transitions must be considered as possible computations. A transition of the complete system is then a set of synchronized local transitions.

If the Estelle specification involves the initialization of N module instances $P_1 \dots P_N$ and M queues $f_1 \dots f_M$, each global state can be described as a t-uple

$$S = << E_i(S) >_{1 \leq i \leq N}, < C_{f_j}(S) >_{1 \leq j \leq M} >$$

where $E_i(S)$ represents the local state of P_i in the state S and $C_{f_j}(S)$ is the content of the queue f_j in the state S.

For each queue f, we define the two morphisms ψ_f and φ_f on transitions which isolate respectively the input and output streams through f. And for each module P_i, we define a function δ_i on transitions which isolate the next-state relation of the transition in module P_i.

4.3.2 How testing unboundedness for Estelle specifications

All what is previously defined seems to give to an Estelle specification a transition system structure. However the presence of *delay* and *priority* transitions arises a new problem because the fact that a transition is fireable not only depends on the local states and the channel contents. For delays it depends on a time process and for priorities it depends on the set of enabled transitions. If delays are used, equality of global states does not insure the necessary property that the sets of fireable transitions of two equal global states must be identical. And if priority are used the very important property 3 is not satisfied. For example suppose $S \leq_g S'$ with $C_f(S) = \epsilon$ and $C_f(S') = a$. If $fireable(S) = \{t_1 \dots t_n\}$. Now imagine t is a transition such that $\forall i = 1 \dots n, priority(t) > priority(t_i)$ and begins with an input of a. Thus $t \notin fireable(S)$ and $\forall i = 1 \dots n, t_i \notin fireable(S')$.

In fact this is because the notion of global state in Estelle is not the good one for the transition system and must be redefined. The generalized prefix order must also be redefined because of priorities.

Thus a new global state of an Estelle specification will be the classical one to which we add the set $D(S)$ of enabled delayed transitions with their delay values and the set F(S) of fireable ones.

The new generalized prefix ordering is then defined by :

$$S \leq_g S' \Longleftrightarrow \begin{cases} (\forall i, 1 \leq i \leq N, E_i(S) = E_i(S')) \\ (\forall f, 1 \leq f \leq M, C_f(S) \leq C_f(S')) \\ D(S) = D(S') \text{ and } F(S) \subseteq F(S') \end{cases}$$

The property 3 is then straightforward and the proof of monotonicity is identical. In the algorithm, this new notion of global state can be simplified. For example if you test for equality, $F(S)$ is unnecessary, thus it can be removed from the completely visited states.

In fact, most restrictions of static Estelle can certainly be removed in order to test for unboundedness on standard Estelle specifications. You must take care of the definition of global states to give sense to equality (they must contain the hierarchical structure of modules) and redefine the relation to preserve property 3. But most existing verification tools using Estelle restrict to its static part.

5 Conclusion and prospects

Dealing with the difficult problem of unboundedness of fifo channels, we have defined a general transition system in which fifo channels are expressed. We have exhibited a new relation between global states included in the prefix order. With a very careful study of fifo channels behaviour, we have proved that this relation induces a test for unboundedness.

We have proved that this test is strictly more powerful than the decision procedures for monogeneous and linear systems. In fact, it is a decision procedure for the class of systems in which the input languages of fifo channels are included in finite unions of left factors of languages of the form $w_1^* \dots w_N^*$ where w_i are words (instead of letters in linear languages).

In order to apply this test to Estelle specifications, we were obliged to add information in global states and give a new definition of the relation. These modifications were required

in order to preserve the monotonicity of the relation and an acceptable definition of equality of reachable states.

The unboundedness test presented here can be improved if it is preceded by a static analysis of the system and if we mix it with other tests. The analysis can for example give bounds to some channels (or even to all of them) and can determine linear or monogeneous languages in which its inputs languages are included. The definition of the relation Θ must then take into account these additional informations in order to advance the detection of unboundedness.

We think that this kind of test can be very useful for protocols designers in the verification phase of protocol specifications. The algorithms given in the paper and all the remarks can give a good idea of a real implementation.

Acknowledgements I am very grateful to Claude Jard who initiated this work and for all his advices, his constant support and the careful reading of the paper.

I also wish to thank Alain Finkel for all his remarks.

References

[Boc78] G.V. Bochmann. Finite state description of communication protocols. *Computer Networks*, 2, October 1978.

[BZ81] D. Brand and P Zafiropulo. *On communicating finite-state machines.* Tech Rep. RZ 1053, IBM Zurich Research Lab., Ruschlikon, Switzerland, Jan. 1981.

[BZ83] D. Brand and P Zafiropulo. On communicating finite-state machines. *J.A.C.M*, 2:323–342, April 1983.

[CF87] A. Choquet and A. Finkel. Simulation of linear fifo nets by petri nets having a structured set of terminal markings. In *Proceedings of the* 8^{th} *European Workshop on Applications and Theory of Petri Nets, Zaragoza, Spain*, June 1987.

[Fin86] A. Finkel. Structuration des systèmes de transitions. Applications au contrôle de parallélisme par files fifo. Thèse d'état, Juin 1986.

[Fin87] A. Finkel. A generalization of the procedure of Karp and Miller to well structured transition systems. In 14^{th} *ICALP , Karlsruhe, RFA*, July 1987.

[Fin88] A. Finkel. A new class of analysable cfsms with unbounded fifo channels: application to communication protocol and distributed solution of the mutual exclusion problem. In *VIII IFIP Symposium, WG 61, Atlantic City*, June 1988.

[FR88] A. Finkel and L. Rosier. *A survey on decidability questions for classes of fifo nets.* Rapport de recherche 456, L.R.I., Nov. 1988.

[GGLR87] M. Gouda, E. Gurari, T. Lai, and L. Rosier. On deadlock detection in systems of communicating finite state machines. *Computers and Artificial Intelligence*, 6(3):209–228, 1987.

[GR84] M. Gouda and L. Rosier. On deciding progress for a class of communicating protocols. In *Proceedimgs of the Eighteenth Annual Conference on Information Sciences and Systems*, pages 663–667, 1984.

[Hol87] G.J. Holzmann. Automated protocol validation in ARGOS, assertion proving and scatter searching. *IEEE trans. on Software Engineering, Vol 13, No 6*, June 1987.

[ISO89] ISO 9074. *Estelle: a Formal Description Technique based on an Extented State Transition Model.* ISO TC97/SC21/WG6.1, 1989.

[JJ89] C. Jard and T. Jéron. On-line model-checking for finite linear temporal logic specifications. In *Proceedings of the Workshop on Automatic Verification Methods for Finite State Systems,* C^3*, Grenoble, France*, June 1989.

[Kel72] R.M. Keller. *Vector replacement systems: a formalism for modeling asynchronous systems.* Tech. Rep 117, Princeton Univ., 1972.

[KM69] R.M Karp and R.E Miller. Parallel program schemata. *Journal of Comput. System Sci.*, 3(2):147–195, 1969.

[KM82] T. Kasai and R.E. Miller. Homomorphisms between models of parallel computation. [illegible] 285–331, 1982.

[Lot83] M. Lothaire. *Combinatorics on Words.* Volume 17, Gian-Carlo Rota, Encyclopedia of Mathematics and its Applications, 1983.

[MM81] R. Martin and G. Memmi. *Spécification et validation de systèmes temps réel à l'aide de réseaux de Petri à files.* Technical Report 3, Revue Tech. Thomson–CSF, Sept. 1981.

[RY86] L. Rosier and H. Yen. Boundedness, empty channel detection, and synchronization for communicating finite automata. In *T.C.S. 44*, pages 69–105, 1986.

Liste des publications internes 1990

PI 508 **RAY TRACING ON DISTRIBUTED MEMORY PARALLEL COMPUTERS: STRATEGIES FOR DISTRIBUTING COMPUTATIONS AND DATA.**
Didier BADOUEL, Kadi BOUATOUCH, Thierry PRIOL
Janvier 1990, 16 Pages.

PI 509 **STABILITY ANALYSIS AND IMPROVEMENT OF THE BLOCK GRAM-SCHMIDT ALGORITHM.**
William JALBY, Bernard PHILIPPE
Janvier 1990, 24 Pages.

PI 510 **TESTING FOR THE UNBOUNDEDNESS OF FIFO CHANNELS IN PROGRAMS.**
Thierry JERON
Janvier 1990, 30 Pages.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique