Testing for unboundedness of Fifo channels

Thierry JERON
IRISA, Campus de Beaulieu,
35042 RENNES Cedex, FRANCE
e-mail : jeron@irisa.fr

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 and find a sufficient condition for unboundedness based on a relation between the nodes of the reachability tree. The construction of the resulting reduced tree can then be applied as well to communicating finite state machines as to Fifo–Nets. Moreover, the test extends existing decidability results. A few modifications of the relation make it available for Estelle specifications.

1 Introduction

The main specification models explicitely describing Fifo channels are communicating finite state machines (Cfsm) [1], Fifo–Nets [18] and some specification languages like Estelle [12]. If Fifo channels are not forehand bounded, the behaviour of such protocol specifications can lead to channels overflows. But the computation of bounds, which is a wished result of a verification, is often impossible to perform. So, we would like to improve the verification tools with some tests for boundedness and unboundedness.

Section 2 begins with a few notations of combinatorics on words, generalized to p–uples of words. We define a transition system which can express the behaviours of most specification models allowing Fifo channels. and introduce the unboundedness problem. After a recall of the main decidability results for Cfsm and Fifo–Nets, we argue out of the testing approach. Section 3 introduces a new relation between global states of the reachability tree. We prove that it provides a sufficient condition for unboundedness. The induced test, based on the construction of a reduced tree, is then a decision procedure for a class of systems strictly including linear and monogeneous systems. Those classes were almost the only non trivial ones in which unboundedness was proved decidable [9, 4, 5, 7]. Section 4 describes an algorithm implementing this test and gives some applications to Cfsm systems. We conclude with some indications on how this test can be applied to Estelle specifications.

2 The unboundedness problem

2.1 Notations and definitions

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 $a_1 a_2 \ldots a_n$. A^* is the set of all the words over A and $A^+ = A^* - \{\epsilon\}$ where ϵ is the empty word. The set of p–uples of words over the alphabets $A_1, \ldots, A_p$ is the cartesian product $\times_{i=1}^{p} A_i^*$.

The *concatenation* of two words is denoted in the usual way by a dot. For p–uples of words, let $X =< x_1, \ldots, x_p >$, $Y =< y_1, \ldots, y_p >$, $Z =< z_1, \ldots, z_p > \in \times_{i=1}^{p} A_i^*$, $X.Y = Z \iff \forall i \in [1..p], x_i.y_i = z_i$

If $z = x.y$ we say that x (resp. y) is a *left (resp. right) factor* of z. $LF(z)$ and $RF(z)$ are 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* generalized to p–uples of words :

$$\text{let } x, y \in A^* \text{ and } X =< x_1, \ldots, x_p >, Y =< y_1, \ldots y_p > \text{ two } p\text{–uples of words,}$$
$$x \leq y \iff x \text{ is a left factor of } y,$$
$$X \leq Y \iff \forall i \in [1..p], x_i \leq y_i.$$

If n is a positive integer, x^n is the concatenation of n words equal to x is called a *power* of x.

Two words x and y *commute* if and only if $x.y = y.x$. Similarly, two p–uples of words $X =< x_1, \ldots, x_p >$ and $Y =< y_1, \ldots, y_p >$ commute if and only if $\forall i \in [1..p]$, x_i and y_i commute.

Remark: two non–empty words commute iff they are powers of the same word [17].

We consider a restrictive form of *transition systems* [16, 15] general enough to describe the behaviour of Cfsm systems, Fifo–Nets and, with a few modifications, Estelle specifications.

A transition system is a 4-uple $\mathcal{S} =< GS, T, \rightarrow, S_0 >$. 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 and $S_0 \in GS$ is the initial state.

A global state $S \in GS$ is an $(N + M)$-uple $S =< E(S), C(S) >$ where :

- $E(S) =< E_1(S), \ldots, E_N(S) >$ is the N–uple of *local states*. Forall i in $[1..N]$, E_i is a function from GS to a finite set LS_i. We note $LS = LS_1 \times \ldots \times LS_N$.
- $C(S) =< C_1(S), \ldots, C_M(S) >$ is the M–uple of *channel contents*. Forall j in $[1..M]$, C_j is a function from GS to M_j^*. The elements of M_j are called *messages*.

A transition $t \in T$ is a triple $t =< \delta(t), in(t), out(t) >$ where :

- $\delta(t) =< \delta_1(t), \ldots, \delta_N(t) >$ and each *local transition function* $\delta_i(t)$ is a partial function from LS_i to LS_i.
- $in(t) =< in_1(t), \ldots, in_M(t) >$ and $out(t) =< out_1(t), \ldots, out_M(t) >$ respectively represent the *channels inputs* and *outputs*. Forall j in $[1..M]$, in_j and out_j are functions from T to $M_j{}^*$.

The transition $t \in T$ is *fireable* in a global state S iff :

- $\delta(t)(E(S))$ is defined, i.e. all local transitions are fireable : $\forall i \in [1..N], \delta_i(t)(E_i(S))$ is defined,
- $in(t) \leq C(S)$ i.e. all receptions are posssible : $\forall j \in [1..M], in_j(t) \leq C_j(S)$.

This transition then leads to a state S'. We write $S \xrightarrow{t} S'$ and we have :

- $E(S') = \delta(t)(E(S))$ (change of local states),
- $in(t).C(S') = C(S).out(t)$ (change of channel contents in a Fifo way).

Remark : according to the definition, a transition can contain several inputs and outputs of words on the alphabets of messages.

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 $n-1$ states $S_2, \ldots, S_n$ s.t. $\forall i \in [1..n]$, $S_i \xrightarrow{t_i} S_{i+1}$

The two functions *in* and *out* are extended into functions on sequences of T^*. Each coordinate in_j (resp. out_j) is extended into a morphism of monoids from T^* to M_j^*.

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

The above partial ordering on global states generalizes the prefix ordering on words :

Definition 1 *S is a generalized prefix of S' denoted $S \leq_g S'$ if and only if the local states of S and S' are identical, and channels contents in S are prefixes (on words) of those of S' i.e. $S \leq_g S' \Longleftrightarrow (E(S) = E(S')$ and $C(S) \leq C(S'))$*

We will note $S <_g S'$ if $S \leq_g S'$ and $S \neq S'$ (thus $\exists f_j$ such that $C_j(S) < C_j(S')$).

As usual, the whole behaviour of a transition system is represented as a directed possibly infinite *reachability tree* $RT(\mathcal{S})$:

- the nodes are labelled by the reachable global states. We will use a small letter s for a node and the corresponding capital letter S for its labelling global state.
- the root s_0 is labelled by the initial global state S_0,
- the edges are labelled by the fireable transitions of the system.

The *reachability graph* $RG(\mathcal{S})$ is obtained from $RT(\mathcal{S})$ by merging nodes labelled with identical global states. Thus its set of nodes is isomorphic to the set of reachable global states.

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

The *input language* $L_I(f_j)$ of a Fifo channel f_j is the set of message sequences filling up the channel f i.e. $L_I(f_j) = out_j(L(\mathcal{S}))$ [5]. This is often used to define particular classes of transition systems :

Definition 2 *Let $L \subset A^*$ be a language. Note $\prod$ the concatenation product.*

$$L \text{ is monogeneous [7] iff } L \subseteq \bigcup_{i=1}^{N} LF(u_i.v_i^*) \text{ with } u_i, v_i \in A^*$$

$$L \text{ is linear [4] iff } L \subseteq \bigcup_{i=1}^{N} \prod_{k=1}^{N_i} a_{i,k}^* \text{ where } a_{i,k}\text{'s are distinct letters of } A$$

$$L \text{ is word–linear iff } L \subseteq \bigcup_{i=1}^{N} LF(\prod_{k=1}^{N_i} w_{i,k}^*) \text{ where } w_{i,k}\text{'s are words of } A^*$$

Linear and monogeneous languages are particular word–linear systems. A transition system is called monogeneous (resp. linear, word–linear) iff the input language of each Fifo channel is monogeneous (resp. linear, word–linear).

Definition 3 *A Fifo channel f_j is bounded iff there exists a constant K_j such that, for every reachable global state, the content of f_j is of length less than K_j.*

Kœnig's lemma asserts that every infinite tree of finite degree has an infinite branch. As a consequence, $RG(\mathcal{S})$ 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. But in the transition system defined above, each local state can take at most a finite number of values and the only possibly unbounded objects are channels contents. So, we have :

Proposition 1 *The reachability graph $RG(\mathcal{S})$ is finite iff all channels are bounded [19].*

2.2 Known results about Cfsm and Fifo–Nets

A system of Cfsm [1] is an N-uple $< P_1, \ldots, P_N >$ of finite state machines which communicate asynchronously by messages. Each 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, and $q_{0\,i} \in LS_i$ is the initial state. For each pair $(P_i, P_j), i \neq j$ of Cfsm, there exists a Fifo channel f_{ij} which allows the communication of messages in M_{ij} from P_i to P_j.

The behaviour of such a system is described by a restrictive form of our transition system. A global state is an $N + (N^2 - N)$–uple of local states and channel contents. The initial state consists in all the local initial states and empty channels. A transition of the system is an element $t \in T = \bigcup_{i=1}^{N} T_i$. A transition is either an input or an output or an internal transition. Thus, the application of the transition function to t modifies a single local state and possibly one channel content.

A Fifo–Net [18, 8] is a generalization of Petri-Nets in which places are replaced by Fifo queues. Edges are labelled by words. A transition is fireable if the label of each incoming edge 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. Its behaviours can also be described by our transition system.

Theorem 1 *The unboundedness problem is undecidable for two communicating finite state machines, Fifo–Nets and our transition system.*

The result for Cfsm systems is a consequence of the capability for Cfsm systems to simulate Turing machines (see [2, 3]). The inclusion of Cfsm systems into Fifo–Nets and into our transition system implies the remainder.

The first idea to bypass the undecidability of the unboundedness problem is to work with restrictive models. This gave some decidability results. For example, for two Cfsm, unboundedness is decidable when one of the two Fifos is bounded [2, 3], one Cfsm is restricted to send only a single type of message [10, 20] or when the input language of one Fifo is monogeneous [5]. For general Cfsm systems and Fifo–Nets unboundedness is decidable when all the queues alphabets consist of a single message type (it is a sub–class of Petri-Nets and then the result is proved in [14]), all the input languages are linear [9, 4] or all the input languages are monogeneous [5, 7].

2.3 Justifying tests for unboundedness

The decidability results presented above seem unsatisfactory. As a matter of fact, the only non trivial ones obtained for more than two Cfsm (and Fifo–Nets) were found for the classes of linear and monogeneous systems. But the membership to those classes is generally undecidable [5]. Moreover, the decision procedures explicitely use the input languages. So they can only be used on systems for which monogeneous or linear languages including their input languages are known. In the case of linear systems, the generalization of the decision procedure for word–linear systems seems impossible. To sum up, the above decision procedures seems too much restrictive because they need additional information on channels which are generally undecidable. But, in counterpart, those classes of systems allow to decide other problems in reachability analysis.

Our approach is conceptually different for we are looking for a sufficient condition for unboundedness which could be applied to all specifications. The basis of this test is the construction of a *reduced tree* (some kind of Karp and Miller tree) : during the construction of the reachability tree, a sequence is stopped at the nodes satisfying the condition or when a loop is detected.

A very important argument in favor of the testing approach in reachability analysis is that, because of the state explosion problem, decidability results, if they exist, are merely theoretical. The reachability trees of real size systems, even if not infinite, are often very large. We must then store an increasing number of objects (the nodes 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 somewhat in the continuation of decidability results for linear and monogeneous systems. Both were based on the construction of a reduced reachability tree with an ordering $\ll$ on global states having three essential properties [6] :

- 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 sub–sequence,
- computability of the ordering and of equality.

Monotonicity and computability gives a sufficient condition and well-ordering assures that the algorithm always stops.

3 The unboundedness test

3.1 Definition of a relation between global states

The generalized prefix ordering seems to be a good candidate for an unboundedness sufficient condition. But we can easily see that it doesn't work : we can build finite graphs in which nodes s and s' satisfy $s \xrightarrow{*} s'$ and $S <_g S'$. Therefore, we are looking for a minimum additional condition to add to this ordering so that the resulting relation $\mathcal{U}$ is a sufficient condition. $\mathcal{U}$ must satisfy the following monotonicity property :

$$s \xrightarrow{w}{}^{*} s' \text{ and } \mathcal{U}(s,s'), S \neq S' \Longrightarrow \exists w' \in T^*, s' \xrightarrow{w'}{}^{*} s'' \text{ and } \mathcal{U}(s',s''), S' \neq S''$$

But the existence of w' is difficult to prove if we don't explicitely characterize it. So we will try to satisfy the weakest property :

$$s \xrightarrow{w}{}^{*} s' \text{ and } \mathcal{U}(s,s'), S \neq S' \Longrightarrow s' \xrightarrow{w}{}^{*} s'' \text{ and } \mathcal{U}(s',s''), S' \neq S'' \qquad (1)$$

Suppose that (1) is true and the prefix ordering $\leq_g$ contains the relation $\mathcal{U}$. Then $\mathcal{U}(s,s')$ implies that $\exists Q$ s.t. $C(S') = C(S).Q$. Using the definition of $\xrightarrow{*}$ we can easily prove that $out(w) \leq Q.out(w)$. Inversely we will prove in the next section that if we add it to $\leq_g$ we have a sufficient condition for unboundedness. Thus $out(w) \leq Q.out(w)$ is the minimum condition to add to $\leq_g$ in order to have a sufficient condition for unboundedness satisfying the property (1).

Remark : $E(S) = E(S')$ and $S \leq_g S'$ and $out(w) \leq Q.out(w)$ (where Q satifies $C(S') = C(S).Q$) is equivalent with $E(S) = E(S')$ and $C(S).out(w) \leq C(S').out(w)$.

We can now define the relation $\mathcal{U}$. It contains the well–ordering on monogeneous systems found in [5] :

Definition 4 *If s and s' are two nodes of the reachability tree labelled by global states S and S' and such that $s \xrightarrow{w}{}^{*} s'$*

$$\mathcal{U}(s,s') \Longleftrightarrow E(S) = E(S') \text{ and } C(S).out(w) \leq C(S').out(w)$$

We will note $\mathcal{U}_{\neq}(s,s')$ in the event of $\mathcal{U}(s,s')$ and $S \neq S'$.

As a consequence of the definition and with the same notations, we have :

Proposition 2 *If $s \xrightarrow{w}{}^{*} s'$ and $\mathcal{U}_{\neq}(s,s')$ then $out(w)$ and Q commute.*

Proof : Let $j \in [1..M]$ be a channel index.
We have $out_j(w) \leq Q_j.out_j(w)$ thus $\exists y \in {M_j}^*$ s.t. $Q_j.out_j(w) = out_j(w).y$.
Now $C_j(S).out_j(w) = in_j(w).C_j(S') = in_j(w).C_j(S).Q$ and $out_j(t)$ is shorter than Q_j, thus $\exists z \in M_j$ s.t. $out_j(w) = z.Q_j$.
Now $Q_j.z.Q_j = Q_j.out_j(w) = out_j(w).y = z.Q_j.y$ and $Q_j.z$ has the same length than $z.Q_j$, Thus $y = Q_j$ and finally $Q_j.out_j(w) = out_j(w).Q_j$. This is true for each channel f_j, and then $Q.out(w) = out(w).Q$.

Remark : Q and $out(w)$ are not necessarily powers of the same M–uple of words.

3.2 The relation is a sufficient condition for unboundedness

Proposition 3 *Let s and s' be two nodes in $RT(\mathcal{S})$ s.t. $S \leq_g S'$, and $t \in T$ a single transition fireable in s. Then t is fireable from s' and the local states of the two reached nodes are identical i.e. if $s \xrightarrow{t} s_1$ and $S \leq_g S'$ then $s' \xrightarrow{t} s'_1$ and $E(S_1) = E(S'_1)$.*

Proof : From the definition of $s \xrightarrow{t} s_1$ we have $\delta(t)(E(S)) = E(S_1)$, $in(t) \leq C(S)$ and $in(t).C(S_1) = C(S).out(t)$

If $S \leq_g S'$ then $E(S) = E(S')$ and $C(S) \leq C(S')$. Therefore $in(t) \leq C(S) \leq C(S')$ and $\delta(t)(E(S'))$ is defined. Then t is fireable in S' and leads to a state S'_1 such that $E(S'_1) = \delta(E(S')) = \delta(E(S)) = E(S_1)$

Remark : the proposition 3 can't be generalized to a sequence of transitions because we generally don't have $S_1 \leq_g S'_1$ and thus we can't apply it again.

We have some kind of monotonicity property :

Proposition 4 *Let $w = t_1 \dots t_n$ and $s, s' \in RT(\mathcal{S})$ such that $s \xrightarrow{w}{}^{*} s'$ and $\mathcal{U}_{\neq}(s, s')$. Note s_1 the node s.t. $s \xrightarrow{t_1} s_1$. Then $s' \xrightarrow{t_1} s'_1$ and $\mathcal{U}_{\neq}(s_1, s'_1)$.*

Proof : $\mathcal{U}(s, s')$ implies $S \leq_g S'$. Thus, proposition 3 asserts that t_1 is fireable from s'. Let s'_1 such that $s' \xrightarrow{t_1} s'_1$. Now note $w' = t_2 \dots t_n.t_1$. We have $s_1 \xrightarrow{w'}{}^{*} s'_1$.

Proposition 3 gives $E(S_1) = E(S'_1)$. So we must prove $C(S_1).out(w') < C(S'_1).out(w')$. Note Q the M–uple of words such that $C(S') = C(S).Q$.

We have $in(t_1).C(S_1).out(w') = C(S).out(t_1).out(w') = C(S).out(w).out(t_1)$.

But $out(w) < Q.out(w)$ implies $out(t_1) < Q.out(t_1)$.

$$\begin{array}{rccccc} \text{Thus } in(t_1).C(S_1).out(w') & < & C(S).out(w).Q.out(t_1) & = & C(S).Q.out(w).out(t_1) \\ & = & C(S').out(t_1).out(w') & = & in(t_1).C(S'_1).out(w') \end{array}$$

And finally $C(S_1).out(w') < C(S'_1).out(w')$.

Theorem 2 *Let s and s' be two nodes of the reachability tree $RT(\mathcal{S})$. If $s \xrightarrow{w}{}^{*} s'$ and $\mathcal{U}_{\neq}(s, s')$ then the reachability graph $RG(\mathcal{S})$ is infinite.*

Proof : Let $w = t_1 \dots t_n$. Proposition 4 can be applied with s, s' and t_1 and again with s_1 and s'_1. If you reiterate this n times you build two sequences $s, s_1, \dots, s_n = s'$ and $s', s'_1, \dots, s'_n = s''$ such that $s_i \xrightarrow{t_{i+1} \dots t_n . t_1 \dots t_i} s'_i$ and $\mathcal{U}_{\neq}(s_i, s'_i)$. Thus $s' \xrightarrow{w} s''$ and $\mathcal{U}_{\neq}(s', s'')$ which implies $S' <_g S''$ for global states. You can now reiterate the same process and build an infinite sequence of nodes $s, s', s'', \dots, s^{(k)}, \dots$ such that the infinite sequence of labelling global states is strictly increasing for $<_g$.

Definition 5 *The (possibly infinite) reduced tree $RedT(\mathcal{S})$ is built like the reachability tree $RT(\mathcal{S})$ except for the following reduction : if $s' \in RT(\mathcal{S})$ and there exists a node s in the sequence from s_0 to s' such that $\mathcal{U}(s, s')$ then s' is marked and the subtrees starting from s' are not explored.*

3.3 Unboundedness decidability for word–linear systems

Theorem 3 *The reduced tree $RedT(\mathcal{S})$ of a word–linear system $\mathcal{S}$ is finite. Thus the unboundedness problem is decidable for word–linear systems.*

Proof : According to the Kœnig's lemma, we only need to prove that in all infinite transition sequence of $RT(\mathcal{S})$, there exist two nodes s and s' such that $s \xrightarrow{*} s'$ and $\mathcal{U}(s, s')$.

Consider an infinite sequence of nodes in $RT(\mathcal{S})$. The set LS of products of local states is finite. So from this sequence we can extract a finite number of sub–sequences such that the local states $E(S)$ in all these nodes are identical. Among them, at least one is infinite. Let $(s_m)_m$ be one of them.

For a particular channel f_j we have $L_I(f_j) \subseteq \bigcup_{i=1}^{N} LF(\prod_{k=1}^{N_i} w_{i,k}^*)$. $L_I(f)$ is a finite union thus, considering all possible contents of f_j, there exists at least one index i such that :

$$\forall m, C_j(S_m) \in \bigcup_{k=1}^{N_i} [LF(RF(C_j(S_0).w_{i,k})) \cup \bigcup_{l=k}^{N_i} (RF(C_j(S_0).w_{i,k}).\prod_{h=k}^{l} w_{i,h}^*.LF(w_{i,l}))]$$

From $(s_m)_m$ we can extract a sub–sequence $(s_n)_n$ such that there exist two indexes k and l with $1 \leq k \leq l \leq N_i$, sequences $(r_{k,n})_n \dots (r_{l,n})_n$ of integers and two constants c and d such that $c \in RF(LF(C_j(S_0).w_{i,k}))$ and $d \in LF(w_{i,l})$ satisfying :

$$\forall n, C_j(S_n) = c.(\prod_{h=k}^{l} w_{i,h}^{r_{h,n}}).d$$

If there exist indexes n_1 and h_1 s.t. $r_{h_1,n_1} \neq 0$ then $\forall h < h_1$ the sequences $(r_{h,n})_{n>n_1}$ are stationary. Thus either from an index n_1, all sequences $(r_{h,n})_n$ are stationary and $(C_j(S_n))_{n>n_1}$ too, or there exists an index h_1 such that from $(r_{h_1,n})_n$ we can extract a strictly increasing sub–sequence $(r_{h_1,p})_p$. In the second case, all sequences $(r_{h,p})_{h<h_1}$ are stationary and $(r_{h,p})_{h>h_1}$ are null. Then $(C_j(S_p))_p$ is strictly increasing for the prefix ordering and $C_j(S_p) = C.w_{i,h_1}^{r_{h_1,p}}.D$ with C a constant in $RF(LF(C_j(S_0).w_{i,k})).\prod_{h=k}^{h_1-1} w_{i,h}^{r_h}$, and D a constant in $LF(w_{i,h_1})$ and constant integers $r_k, \dots, r_{h-1}$. Note $Q_{j,p}$ the word s.t. $C_j(S_{p+1}) = C_j(S_p).Q_{j,p}$ and v_p the transition sequence from s_p to s_{p+1}.

- if $k \neq h_1$ then $\forall p, in_j(v_p) = \epsilon$ and $out_j(v_p) = Q_{j,p}$.
  Thus $C_j(S_p).out_j(v_p) < C_j(S_{p+1}).out_j(v_p)$.

- otherwise $C \in RF(LF(C_j(S_0).w_{i,h_1}))$. If there is a finite number of p such that $in_j(v_p) \neq \epsilon$ then there is an index p_1 such that $\forall p > p_1, in(v_p) = \epsilon$ and it is like in the preceding case. Otherwise there is an index p_1 s.t. $C_j(S_0) \leq inj(v_0 \dots v_{p_1})$ (i.e. $C_j(S_0)$ is completely received). We can then take C in $RF(w_{i,h_1})$. Now note B and E the words s.t. $B.C = D.E = w_{i,h_1}$. Thus $out_j(v_p)$ has the form $(E.D)^{l_p}$. We have :

$C_j(S_p) = C.(B.C)^{r_{h_1,p}}.D = C.(D.E)^{r_{h_1,p}}.D$
and $C_j(S_{p+1}) = C.(B.C)^{r_{h_1,p+1}}.D = C.(B.C)^{r_{h_1,p}}.(D.E)^{r_{h_1,p+1}-r_{h_1,p}}.D$.
Thus $C_j(S_p).out_j(v_p) < C_j(S_{p+1}).out_j(v_p)$.

Then for the channel f_j we have found a sub–sequence $(s_p)_p$ such that the sequence $(C_j(S_p))_p)$ is either stationary, either strictly increasing for $\leq$ and satisfying $\forall p, C_j(S_p).out_j(v_p) < C_j(S_{p+1}).out_j(v_p)$. We then apply the same construction with the obtained sub–sequence consecutively with the other channels. Finally, this diagonalization builds an infinite sub–sequence in which two consecutive nodes satisfy $\mathcal{U}$.

4 Application

4.1 Implementation of the unboundedness test

A prototype of the program building the reduced trees of extended Cfsm systems has been written in Pascal.

While testing unboundedness we only needs to know the current transition sequence and states from the root. We can then theoretically use a depth first strategy in which we only keep trace of this sequence in a stack. In order to avoid retraversing nodes with same labelling global states we also use a memory in which we record completely visited nodes and check for equality. The reachability graph may be infinite, so, when the memory overflows we randomly replace nodes [11, 13].

The number of tests performed for each new node s' is growing with its depth in $RT(\mathcal{S})$. In order to avoid this and in the continuity of the testing approach, we can bound the number of tested nodes s. The increase of time needed for the tests for each node s' is then bounded by a function independent on its depth.

Although it heavily depends on the graph structure, we can give the theoretical time needed for all the computations of $\mathcal{U}(s, s')$ for a fixed node s' in function of its depth $d(s')$ in $RT(\mathcal{S})$. The number of nodes s in question is less than $d(s')$. Their depth range over $[0..d(s') - 1]$. For each node s, checking the equality $E(S) = E(S')$ requires less than N comparisons. Let $s \xrightarrow{*} \xrightarrow{w} s'$ and note $|C(S)| = Max_{j=1}^{M}(C_j(S))$, $|out(w)| = Max_{j=1}^{M}(out_j(w))$. Checking whether $C(S).out(w) \leq C(S').out(w)$ needs less than $M.(|C(S')| + |out(w)|)$ message comparisons. The total number of comparisons $F(s')$ for s' fixed is less than $\Sigma_{d(s)=0}^{d(s')-1} M.(|C(S')| + |out(w)|)$. We can reasonably suppose that $|C(S')| \leq K_1.d(s')$ and $|out(w)| \leq K_2.(d(s') - d(s))$.

Thus $F(s') \leq M.K_1.d(s')^2 + M.K_2.\frac{d(s')^2+d(s')}{2}$. The complexity of checking whether there exists an s such that $\mathcal{U}(s, s')$ for a given s' is then in $\mathcal{O}(d(s')^2)$.

4.2 Examples

This section gives some examples of unbounded Cfsm systems. They were all tested with a prototype of the algorithm presented above, running on a SUN4 workstation.

In the system $\mathcal{S}_1$ of figure 1, $L_I(f_1)$ is word–linear ($\subseteq LF((ab)^*.a.(ba)^*)$) $L_I(f_2)$ is monogeneous ($\subseteq LF(cd)^*$). Unboundedness is then decidable (Theorem 3). The program

Figure 1: A word–linear system

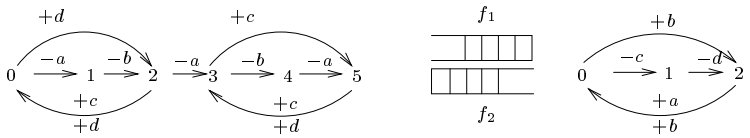


Figure 2: A simplified connect-disconnect protocol

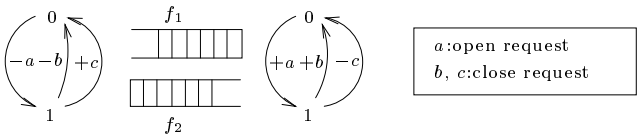
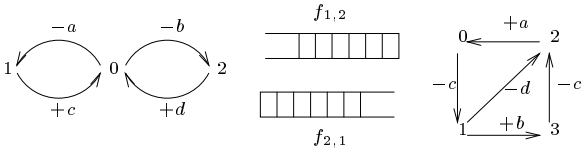


Figure 3: An unbounded system not detected as unbounded



generates the 1221 nodes of the reduced tree in less than 7s. It detects 247 marked states and spent 27% of the time in the computation of $\mathcal{U}$.

The Cfsm system $\mathcal{S}_2$ of figure 2 is a simplified connect–disconnect protocol which is detected unbounded. It is also word–linear : $L_I(f_1) \subseteq LF((ab)^*.a.(ab)^*)$ and $L_I(f_2) \subseteq c^*$. Then its reduced tree is finite. The reduced tree has only 13 nodes, 4 of which marked.

The Cfsm system $\mathcal{S}_3$ of figure 3, which produces an infinite reachability tree, has been specially designed to fail our test. There exists, among others, an infinite transition sequence of the form $w.v.u^2.v^2. \ldots u^{2^n}.v^{2^n} \ldots$ traversing an infinite number of increasing states (for the generalized prefix ordering) but the relation $\mathcal{U}$ is always false. This kind of "exponential growing" cannot be detected by our procedure. With channels bounded to 8 messages, the 1473 nodes of the finite graph are generated in 11 s. The time spent computing $\mathcal{U}$ is 50% of the total time.

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 that of $\mathcal{S}_3$. Thus there are infinite sequences in which the test is always false. But there are some other sequences in which the test is satisfied. We are in the case where unboundedness should be detected although the reduced tree is infinite.

4.3 Unboundedness of Estelle specifications

The Estelle language is a Formal Description Technique (FDT) standardized by ISO and well suited for the specification of distributed systems [12]. If we restrict to a static part of the Estelle language (no dynamical change of topology, all variables are bounded), a specification is a set of automaton extended with Pascal constructs which communicate by messages through Fifo queues. Each local transition is atomic. The semantics of the complete Estelle specification is defined in an operational way in terms of transformations of a *global state* (union of all local states and channel contents). Several parallelism can be described : 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 possible interleavings of transitions must be considered as possible computations.

In order to apply our unboundedness test we must insure that the transition system of an Estelle specification satisfies the essential proposition 3. The *delay* and *priority* clauses used in transitions force us to redefine the notions of global state and prefix ordering. The definition of $\mathcal{U}$ and the proof of theorem 4 are then almost identical.

5 Conclusion and prospects

Dealing with the difficult problem of unboundedness of Fifo channels, we have defined a general transition system explicitely describing Fifo channels and including Cfsm systems and Fifo–Nets.

We have exhibited a new relation between global states included in the prefix order and proved that it is a sufficient condition for unboundedness. This induces a test for unboundedness by the construction of a reduced tree. The testing approach allows the detection of unboundedness even when the reduced tree is infinite. The test could also be improved if it was preceded by a static analysis of the system [2] which could give bounds to some channels.

We proved that this test is, in a certain sens, strictly more powerful than the decision procedures for monogeneous and linear systems : the reduced tree of a word–linear system is always finite. Thus it provides a decision procedure for the class of word–linear systems. But quasi–liveness is still an open problem in this class. Our relation doesn't allow to build a coverability tree.

We think that the unboundedness test can be very useful for protocols designers in the verification phase of protocol specifications. It can easily be mixed with an on–line model–checking algorithm [13] and then test unboundedness while, for example, verifying temporal logic formulas. This is achievable on Estelle specifications.

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

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

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

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

[4] 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.

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

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

[7] 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.

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

[9] 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.

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

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

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

[13] C. Jard and T. Jéron. On-line model-checking for finite linear temporal logic specifications. In *Proceedings of the International Workshop on Automatic Verification Methods for Finite State Systems*, Grenoble, France, June 1989. Springer–Verlag, LNCS #407, pages 275–285.

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

[15] T. Kasai and R. Miller. Homomorphisms between models of parallel computation. *J.C.S.S*, 25:285–331, 1982.

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

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

[18] 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.

[19] J. Pachl. Reachability problems for cfsms. Research Report CS–82–12, Univ. of Waterloo, Dept. of Comp. Sci., May 1982.

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