Verifying identical communicating processes is undecidable*

Alain Finkel [†]
ENS de Cachan
Paris, France

Pierre McKenzie [‡]
Université de Montréal
Montréal, Canada

August 8, 1995

Abstract

We prove that boundedness and reachability tree finiteness are undecidable for systems of two identical automata communicating via two perfect unbounded one-way FIFO channels and constructed solely from cycles about their initial states. Using a form of mutual exclusion for such systems, we prove further that undecidability holds even when the identical automata are totally indistinguishable in the sense that their initial states are identical and both channels are initially empty.

1 Introduction

A system of communicating finite state machines (CFSMs) consists of a finite number of processes (i.e. automata) communicating with each other by sending and receiving messages via perfect unbounded one-way FIFO channels.

*Accepted for presentation at the 5th Italian Conference on Theoretical Computer Science, November 1995.

†Laboratoire d'Informatique Fondamentale et Appliquée (LIFAC), École Normale Supérieure de Cachan, 61, avenue du Président Wilson, 94235 Cachan, France. Supported by the French inter-PRC Project, Modèles et Preuves. `finkel@lifac1.ens-cachan.fr`

‡Dép. d'informatique et recherche opérationnelle, Université de Montréal, C.P. 6128, Succursale Centre-ville, Montréal (Québec), H3C 3J7 Canada. Supported by the NSERC of Canada and by the FCAR du Québec. `mckenzie@iro.umontreal.ca`

Such systems can model communication protocols or distributed algorithms written for example in Estelle [8] or in SDL [6].

Brand and Zafiropulo [4, 5] have shown in 1981 that the general CFSM model has the power of Turing machines. Other proofs for the similar FIFO Petri net model are known [11]. It is known as well that a single CFSM with a FIFO channel has the power of Turing machines [13]. Along related lines, it is known that from the formal language recognition viewpoint, a CFSM using $k+1$ FIFO channels is strictly more powerful than a CFSM using only k FIFO channels.

Recent work has shown that decidability is sometimes attainable in the case of systems of CFSMs over unreliable (i.e. lossy, insertion, duplication) channels [9, 1, 2, 7]. In a different vein, for purposes of modeling distributed algorithms in which processes are identical, and in light of the surprising difficulty of problems like leader election in anonymous networks (see for example [12]), it is interesting to consider systems of *identical* CFSMs. For example, Benslimane in [3] claims decidability results for restricted classes of systems of identical CFSMs[1].

In this paper we show that the systems considered by Benslimane, namely systems of two identical CFSMs constructed solely from elementary cycles about their initial states, can simulate Turing machines. It follows that the finite reachability tree problem and the finite reachability graph problem (also called the boundedness problem) are undecidable for such systems.

Our first Turing machine simulation "distinguishes" the two participating identical automata by the choice of one specific channel in which to store the initial Turing machine configuration. Although this simulation is straightforward and it extends that of Brand and Zafiropulo [4] in an intuitive way, we find that its formal correctness proof still requires care. Then we modify the simulation and show that the distinction between the two participating

[1]The precise decidability claims made by Benslimane in [3] are not clear. The abstract, the introduction, and the conclusion of [3] claim decidability results which we prove false in the present paper. On the other hand, restrictions are casually added to CFSMs in the body of [3]. For instance, the theorem in [3] which states decidability of the boundedness problem for identical CFSMs with initial cycles restricts each cycle to emit strictly more than it receives. If this theorem of [3] is indeed correct, we suspect that the same result holds without restricting the CFSMs to be identical. We note moreover that the undecidability results reported in the present paper extend to the case of identical automata in which each cycle emits more than it receives, but then we must drop the requirement that each automaton cycle be constructed about the automaton's local initial state.

automata can be avoided entirely, even within the restricted model in which only cycles about the initial local automata states are allowed. We do this by implementing a kind of once-only mutual exclusion, allowing one and only one of the identical automata to initialize its output channel (and preventing any future execution of the initializing cycles). This initialization problem is akin to leader election in deterministic anonymous networks. Interestingly, although leader election is provably impossible in such networks (see [12]), we succeed in "initializing a leader" by ensuring that unwanted computations are blocked (such deadlocks are generally disallowed in distributed algorithms).

Section 2 in this paper defines notation. Section 3 presents our basic simulation and undecidability result. Section 4 discusses one-time mutual exclusion and leader initialization, extending undecidability to the case of indistinguishable CFSMs. Section 5 concludes.

2 Preliminaries and notation

A finite alphabet is denoted Σ, Σ^* is the set of all finite words over Σ, λ is the empty word, and $|w|$ denotes the length of a word $w \in \Sigma^*$.

2.1 Systems of CFSMs

Definition. A *communicating finite state machine* (CFSM) is a finite automaton $A = (Q, q_0, \Sigma, \delta)$ where, defining $\pm\Sigma$ as $\{+, -\} \times \Sigma$,

- Q is a finite set of states,
- $q_0 \in Q$ is the initial state,
- Σ is a finite alphabet, and
- $\delta \subseteq Q \times (\pm\Sigma) \times Q$ is a set of possible *transitions*.

The CFSM A is *initial* if, for every accessible state $q \in Q$, there exist transitions $(q, y_1, q_1), (q_1, y_2, q_2), \ldots, (q_k, y_k, q_0) \in \delta$.

(Note that the alphabet of A in the usual finite automaton sense is $\pm\Sigma$ rather than Σ, i.e. A sees $(+, a) \in \pm\Sigma$ and $(-, a) \in \pm\Sigma$ as single symbols, which we henceforth write as $+a$ and $-a$ respectively. Intuitively, $+a$ denotes the reception of a, and $-a$ the emission of a.)

Definition. A *system S of two CFSMs* is a pair of CFSMs $S = (A_1, A_2)$ with $A_1 = (Q_1, q_{0,1}, \Sigma, \delta_1)$ and $A_2 = (Q_2, q_{0,2}, \Sigma, \delta_2)$. We say that A_1 is the *mate* of A_2 and that A_2 is the *mate* of A_1. The *global state* (*state* for short) of S is a quadruple $(q_1, q_2; w_{12}, w_{21}) \in Q_1 \times Q_2 \times \Sigma^* \times \Sigma^*$.

The operational semantics of a system of CFSMs is defined by the firing of a transition which changes the system's global state in one step.

Definition. Let $S = (A_1, A_2)$ be a system of two CFSMs. A state $s' = (q'_1, q'_2; w'_{12}, w'_{21})$ is *reachable* from another state $s = (q_1, q_2; w_{12}, w_{21})$ by the firing of a transition t, written $s \to s'$ or redundantly $s \xrightarrow{t} s'$, if one of the following two cases holds:

1. There exist $i, j \in \{1, 2\}$, $i \neq j$, such that $t = (q_i, -a, q'_i)$ with
   (a) $q'_j = q_j$,
   (b) $w'_{ij} = w_{ij}a$ and $w'_{ji} = w_{ji}$.
2. There exist $i, j \in \{1, 2\}$, $i \neq j$, such that $t = (q_i, +a, q'_i)$ with
   (a) $q'_j = q_j$,
   (b) $w'_{ij} = w_{ij}$ and $w_{ji} = aw'_{ji}$.

(Condition (1) above describes the sending of a along A_i's output channel, known in global state s to contain w_{ij}. Condition (1a) says that the local state of A_j is unaffected by the transition. Condition (1b) updates A_i's output channel and says that A_j's output channel, known in global state s to contain w_{ji}, is unaffected by the transition. Condition (2) on the other hand describes the reception of a by M_i, from A_j's output channel, which is also A_i's input channel: upon the reception of a by A_i, A_i's output channel and the local state of A_j are unaffected, while an a is removed from w_{ji}.)

Definition. Consider $S = ((Q_1, q_{0,1}, \Sigma, \delta_1), (Q_2, q_{0,2}, \Sigma, \delta_2))$ a system of two CFSMs, and $s_0 \in Q_1 \times Q_2 \times \Sigma^* \times \Sigma^*$ a state of S. The *reachability set $RS(S, s_0)$ of S in s_0* is the set of states reachable in a finite number of steps from s_0:

$$RS(S, s_0) = \{s \in Q_1 \times Q_2 \times \Sigma^* \times \Sigma^* \mid s_0 \xrightarrow{*} s\}.$$

The *reachability tree $RT(S, s_0)$ of S in s_0* is the tree with root labelled s_0, such that a node labelled s has a child labelled s' iff $s \to s'$.

By a branch of $\mathrm{RT}((A_1, A_2), s_0)$ we will often refer to the sequence σ of A_1 or A_2 transitions required to produce the sequence of reachable states $s_0 \to s_1 \to s_2 \to \cdots$ found along the tree branch. More generally, let σ and γ be two sequences of transitions (not necessarily executable in this or in any order) of a system (A_1, A_2). The length of σ is denoted $|\sigma|$. We say that γ is a *subword* of σ, written $\gamma \sqsubseteq \sigma$, if inserting transitions at appropriate places within γ can produce σ. If $\gamma \sqsubseteq \sigma$, we write $\sigma - \gamma$ for the sequence obtained from σ by deleting the leftmost occurrence of the subword γ. We write $\sigma|_i$, $i \in \{1, 2\}$, for the subword of σ formed by deleting from σ all but the A_i transitions. We further write $\sigma \sim \gamma$ if, for each $i \in \{1, 2\}$, $\sigma|_i = \gamma|_i$.

The *reachability tree finiteness* problem is the following:
GIVEN: a system $S = ((Q_1, q_{0,1}, \Sigma, \delta_1), (Q_2, q_{0,2}, \Sigma, \delta_2))$ of two CFSMs and a state $s_0 \in Q_1 \times Q_2 \times \Sigma^* \times \Sigma^*$.
DETERMINE: whether $\mathrm{RT}(S, s_0)$ is finite.

The *boundedness* problem is the following:
GIVEN: a system $S = ((Q_1, q_{0,1}, \Sigma, \delta_1), (Q_2, q_{0,2}, \Sigma, \delta_2))$ of two CFSMs and a state $s_0 \in Q_1 \times Q_2 \times \Sigma^* \times \Sigma^*$.
DETERMINE: whether $\mathrm{RS}(S, s_0)$ is finite.

2.2 Turing machine assumptions

Our model of a Turing machine $M = (Q, \Sigma, \Gamma, q_0, B, \delta)$ is the standard deterministic one-way-infinite single tape model (see [10]), except that we omit final states. Hence Q is the state set, $q_0 \in Q$ the initial state, Σ the input alphabet, Γ the tape alphabet, $B \in \Gamma$ the blank symbol, and $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\texttt{left}, \texttt{right}\}$ the (partial) transition function. We assume with no loss of generality that

- M accepts an input w iff M halts on w,
- if M does not halt on w, then M eventually moves its tape head arbitrarily far to the right.

Definition. A *configuration* of the Turing machine M is a word $uqv\#$ with $uv \in \Gamma^*$, $q \in Q$, and $\#$ a fixed symbol not in Γ. (Word $uqv\#$ represents M in state q, with initial tape content uv and the rest blank, and with M's tape head positioned under the first symbol to the right of u; symbol $\#$ is a redundant marker used for notational convenience later.) We write $c \vdash_M c'$ when one transition of M leads from configuration c to configuration c'.

3 Identical initial CFSMs

In this section we reduce the halting problem for Turing machines to the boundedness and to the tree reachability finiteness problems for systems of two identical initial CFSMs.

In Section 3.1 we construct, from any Turing machine M and from any word w, a system $S(M)$ of two identical initial CFSMs with initial global state $s_0(M, w)$. This system simulates the computation of M on w in the sense of Theorem 3.9: M accepts w iff $\mathrm{RT}(S(M), s_0(M, w))$ is finite iff $\mathrm{RS}(S(M), s_0(M, w))$ is finite. We prove in Section 3.2 that the simulation works and we draw the undecidability consequences in Section 3.3.

Throughout Section 3, we fix $M = (Q_M, \Sigma_M, \Gamma_M, q_0, B, \delta_M)$ an arbitrary Turing machine and we fix an arbitrary input $w \in \Sigma_M^*$.

3.1 The construction

Our basic construction of a system of two identical initial CFSMs (A_1, A_2) is straightforward and borrows from Brand and Zafiropulo [4]. We specify only one initial CFSM A, with the understanding that A_1 and A_2 are identical copies of A.

The core idea of the simulation is that A reads the current configuration of M from its input channel, skipping and reemitting symbols until it reaches the vicinity of M's tape head. Then A processes this vicinity by emitting the new vicinity resulting from the appropriate transition of M. Then A returns to skipping and emitting until the next time it encounters M's tape head.

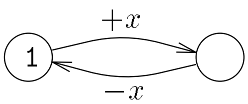
In the Brand and Zafiropulo construction, one CFSM actively performs the simulation, while its mate blindly skips and reemits. In our construction, both (identical) CFSMs actively perform the simulation. Hence, in a legal transition sequence of our system, A_1 and A_2 "advance" the simulation in alternation. Although our construction is simple, ensuring its correctness requires a careful proof that no undesired interference occurs in this process.

Our CFSM A will be defined as $A = (Q, 1, \Gamma_M \cup \{\#\} \cup Q_M, \delta)$. We will not define the set Q explicitly, but we will specify δ and let the reader deduce Q. Since A is initial, A is made up of "cycles" about state 1. There are four types of cycles and each type will be specified by the sequence of transitions encountered when starting in state 1 and traversing the cycle.

3.1.1 Type 1: The copying cycles

There is one copying cycle for each $x \in \Gamma_M \cup \{\#\}$:

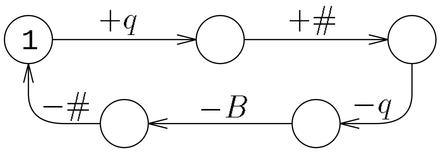
1. `Receive:` x
2. `Emit:` x



3.1.2 Type 2: The blank insertion cycles

There is one B insertion cycle for each $q \in Q_M$:

1. `Receive:` q
2. `Receive:` $\#$
3. `Emit:` q
4. `Emit:` B
5. `Emit:` $\#$

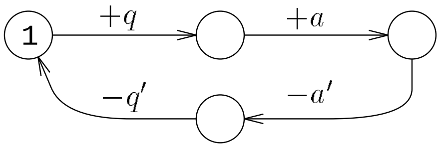


Hence the representation of the configuration of M is extended to the right by a blank, whenever the tape head of M points past the rightmost currently represented position.

3.1.3 Type 3: The right head motion cycles

There is such a cycle for each $(q, a, q', a') \in Q_M \times \Gamma_M \times Q_M \times \Gamma_M$ such that $\delta_M(q, a) = (q', a', \texttt{right})$:

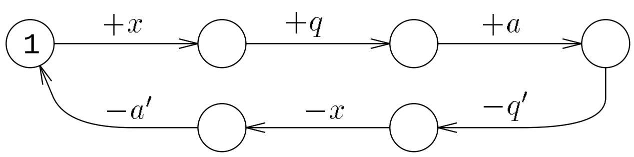
1. `Receive:` q
2. `Receive:` a
3. `Emit:` a'
4. `Emit:` q'



3.1.4 Type 4: The left head motion cycles

There is such a cycle for each $(x, q, a, q', a') \in \Gamma_M \times Q_M \times \Gamma_M \times Q_M \times \Gamma_M$ such that $\delta_M(q, a) = (q', a', \texttt{left})$:

1. `Receive:` x
2. `Receive:` q
3. `Receive:` a
4. `Emit:` q'
5. `Emit:` x
6. `Emit:` a'



3.2 The construction works

We have constructed, from a Turing machine $M = (Q_M, \Sigma_M, \Gamma_M, q_0, B, \delta_M)$ and a word w, a system $S(M)$ of two CFSMs. We define the state $s_0(M, w)$ as $(1, 1; q_0 w\#, \lambda)$. We henceforth denote $S(M)$ by S and $s_0(M, w)$ by s_0.

Proposition 3.1 *For any sequence $C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \cdots \vdash_M C_n$, there exists a branch in $RT(S, s_0)$ such that*

$$s_0 \xrightarrow{*} s_1 \xrightarrow{*} s_2 \xrightarrow{*} \ldots \xrightarrow{*} s_n$$

where $s_{2i} = (1, 1; C_{2i}, \lambda)$ and $s_{2j+1} = (1, 1; \lambda, C_{2j+1})$, $0 \leq 2i \leq n$, $1 \leq 2j+1 \leq n$.

Proof. By induction on n, carrying out the simulation in the obvious way. The base case $n = 0$ is clear by definition of s_0. In the inductive step, we conclude that $s_n \xrightarrow{*} s_{n+1}$, knowing that $s_0 \xrightarrow{*} s_n$ and knowing which Turing machine transition led M from C_n to C_{n+1}, by another induction on the length of C_n. $\square$

Proposition 3.1 states that to any derivation sequence of configurations of M corresponds a precise sequence of states of S. Due to the unpredictable interleavings of A_1 and A_2, the converse is harder to express properly, let alone to prove. We begin with a series of lemmas.

Lemma 3.2 *Let $i \in \{1, 2\}$. If $\alpha = \alpha_1\alpha_2\alpha_3$ is a branch of $RT(S, s_0)$ such that $\alpha_2 = \alpha_2|_i$, then $|\alpha_2| \leq 3|\alpha_1| + 6|w| + 17$.*

Proof. Let v be the content of A_i's input channel immediately after the execution of α_1. Since by construction every cycle in A_i consumes one or more symbols from v, and since no cycle has length greater than 6, we have

$$|\alpha_2| \leq 6|v| + 5.$$

Now by overestimating the number of executions of a blank extension cycle in either A_1 or A_2, we obtain

$$|v| \leq |\alpha_1|/2 + |q_0 w\#| = |\alpha_1|/2 + |w| + 2.$$

Putting together the two inequalities yields $|\alpha_2| \leq 3|\alpha_1| + 6|w| + 17$. □

Lemma 3.3 *Let $x = (q, y, q')$ be an A_i transition, $i \in \{1, 2\}$. If $\sigma = \sigma_1 x \sigma_2$ is a branch of $RT(S, s_0)$ such that $|\sigma| \geq 4096(|\sigma_1| + 2|w| + 7)$, then $\bar{x} \sqsubseteq \sigma_2$, where $\bar{x}$ is a sequence of A_i transitions, $|\bar{x}| \geq 0$, leading A_i from its state q' back to its initial state* 1.

Proof. Write

$$\sigma_2 = \gamma_0 x_1 \gamma_1 x_2 \gamma_2 \ldots x_n \gamma_n,$$

where $x_1, x_2, \ldots, x_n$, $n \geq 0$, are the only A_i transitions occurring in σ_2. A straightforward induction[2] using Lemma 3.2 proves that, for $0 \leq k \leq n$,

$$|\sigma_1 x \gamma_0 x_1 \gamma_1 \ldots x_k \gamma_k| \leq 4^{k+1}(|\sigma_1| + 2|w| + 7) - 2|w| - 7.$$

Hence, using our hypothesis,

$$\begin{array}{rcl} 4096(|\sigma_1| + 2|w| + 7) \leq |\sigma| & \leq & 4^{n+1}(|\sigma_1| + 2|w| + 7) - 2|w| - 7 \\ & < & 4^{n+1}(|\sigma_1| + 2|w| + 7), \end{array}$$

[2] For the base case,

$$\begin{array}{rcl} |\sigma_1 x \gamma_0| = |\sigma_1 x| + |\gamma_0| & \leq & |\sigma_1 x| + 3|\sigma_1 x| + 6|w| + 17 \\ & = & 4(|\sigma_1| + 2|w| + 7) - 2|w| - 7, \end{array}$$

where the inequality follows from Lemma 3.2, which we use again in the inductive step:

$$\begin{array}{rcl} |\sigma_1 x \gamma_0 \ldots x_{k+1} \gamma_{k+1}| & \leq & |\sigma_1 x \gamma_0 \ldots x_k \gamma_k x_{k+1}| + 3|\sigma_1 x \gamma_0 \ldots x_k \gamma_k x_{k+1}| + 6|w| + 17 \\ & = & 4\{4^{k+1}(|\sigma_1| + 2|w| + 7) - 2|w| - 7 + |x_{k+1}|\} + 6|w| + 17 \\ & = & 4^{k+2}(|\sigma_1| + 2|w| + 7) - 2|w| - 7. \end{array}$$

so that $n > 4$. This implies that sufficiently many A_i transitions are available within σ_2 to complete an A_i cycle, thus completing the proof. $\square$

Let $\{i, j\} = \{1, 2\}$. We define a *head processing state* of A_i to be any local state of A_i found along a cycle of type 2, 3 or 4 strictly between the reception of a symbol $q \in Q_M$ and the emission of the next symbol $q' \in Q_M$. We say that A_i is *active* in a global state $(q_1, q_2; \gamma_1, \gamma_2)$ iff one of the following two conditions holds:

(i) q_i is a head processing state of A_i, q_j is not a head processing state of A_j, and $\gamma_1\gamma_2$ contains no symbol $q \in Q_M$, or

(ii) neither q_i nor q_j are head processing states, A_i's input channel γ_j contains a symbol $q \in Q_M$, and A_j's input channel γ_i does not contain a symbol $q \in Q_M$.

Lemma 3.4 *In any state s of $RT(S, s_0)$, A_1 is active iff A_2 is not.*

Proof. By induction on the length of the sequence of transitions leading from s_0 to s. $\square$

We say that a sequence of transitions σ of the system (A_1, A_2) *contains an interrupted cycle* if, for some $i \in \{1, 2\}$ and for some A_i transition $x = (q, y, q')$ in σ with $q' \neq 1$, A_i's mate performs a transition in σ after x but before A_i can return to state 1 from q'.

Lemma 3.5 *For every branch σ of $RT(S, s_0)$, there exists a branch $\sigma_1\sigma_2 \sim \sigma$ such that $|\sigma_1| > |\sigma|/4096 - 2|w| - 7$ and σ_1 contains no interrupted cycle.*

Proof. Write $\sigma = \gamma_1 x \gamma_2$, with x the leftmost occurrence of a transition belonging to an interrupted cycle, say of A_i, $i \in \{1, 2\}$. If

$$|\gamma_1| > |\sigma|/4096 - 2|w| - 7,$$

then we set $\sigma_1 = \gamma_1$, $\sigma_2 = x\gamma_2$, and we are done. Otherwise,

$$|\sigma| \geq 4096(|\gamma_1| + 2|w| + 7),$$

so that Lemma 3.3 applies to $\sigma = \gamma_1 x \gamma_2$. It follows that $\bar{x} \sqsubseteq \gamma_2$, for some minimal $\bar{x}$ eventually completing the A_i cycle interrupted at x. We claim that we can permute $x\gamma_2$ into an executable sequence $c\gamma_2' \sim x\gamma_2$ where c is an uninterrupted cycle. We will thus have found an executable sequence $\sigma' = \gamma_1 c \gamma_2' \sim \sigma$ having a prefix $\gamma_1 c$ with no interrupted cycle. We can then iterate the argument, replacing σ by σ'. This process will eventually terminate because the uninterrupted prefix increases in length at each iteration, while $|\sigma|$ remains unchanged. We now prove our claim, distinguishing two cases.

Case 1: $\gamma_1 x\bar{x}(\gamma_2 - \bar{x})$ is a branch in $\mathrm{RT}(S, s_0)$. Then our claim is proved.

Case 2: $\gamma_1 x\bar{x}(\gamma_2 - \bar{x})$ is not a branch in $\mathrm{RT}(S, s_0)$. This case arises because A_i gets blocked within $x\bar{x}$ (since executing an A_i transition earlier than expected cannot hinder the progress of A_i's mate). Hence x is the first A_i transition in a cycle of type 2, 3, or 4, and A_i gets blocked within $x\bar{x}$ upon emptying its input channel (because the blocking of A_i on a nonempty input channel would contradict $\bar{x} \sqsubseteq \gamma_2$). Now A_i's mate is in its initial state after x, by the choice of x. Let y be the first transition of A_i's mate in γ_2, and write $\bar{y}$ for the rest of A_i's mate's cycle beginning with y. Note that $\bar{y} \sqsubseteq \gamma_2$ because γ_2 is sufficiently long and A_i's input channel is empty after x. We further distinguish two subcases.

Subcase 2.1: A_i gets blocked within $x\bar{x}$ immediately after executing $+q$, $q \in Q_M$, in a cycle of type 2, 3, or 4. Then, by Lemma 3.4, A_i's mate cannot become active until A_i is unblocked. Hence y is the first transition of a cycle of type 1. But then $\gamma_1 y\bar{y}x(\gamma_2 - (y\bar{y}))$ is executable because x is a reception and $\bar{y}$ an emission.

Subcase 2.2: A_i gets blocked immediately after x in a cycle of type 4. Then A_i will not emit until it receives some $q \in Q_M$ followed by some $a \in \Gamma_M$. Since A_i's mate is in its initial state, y must be the beginning of a cycle of type 2 or 4. In such a cycle, A_i's mate emits only when all its receptions are complete. Hence $\gamma_1 xy\bar{y}(\gamma_2 - (y\bar{y}))$ is executable, and so is $\gamma_1 y\bar{y}x(\gamma_2 - (y\bar{y}))$ because x is a reception.

All other subcases in fact fall into Case 1. This therefore proves our claim and concludes the proof of the lemma. □

Lemma 3.6 *Let $\{i, j\} = \{1, 2\}$ and suppose that A_i is active in global state $s = (1, 1; \gamma_1, \gamma_2)$. If $k > 4|\gamma_i| + 6|\gamma_j| + 3$ and $s \to s_1 \to s_2 \to \cdots \to s_k$, then A_j enters a head processing state in some s_j, $1 \le j \le k$.*

Proof. The following strategy (or any interleaving thereof) will postpone A_j entering a head processing state for the longest time:

1. A_j consumes γ_i using copying cycles,
2. A_i consumes $\gamma_j\gamma_i$ using copying cycles and one cycle of type 2, 3 or 4,
3. A_j executes at most $|\gamma_j|$ copying cycles and stops short of consuming $q \in Q_M$,
4. A_i empties its input channel using at most $|\gamma_j|$ copying cycles,
5. A_j enters its head processing state. □

Corollary 3.7 *An infinite sequence σ in $RT(S, s_0)$ with no interrupted cycle contains an infinite number of cycles of type 3 or 4.*

Proof. An induction using Lemma 3.6 proves that σ contains an infinite number of cycles of type 2, 3, or 4. Let $\gamma \sqsubseteq \sigma$ be the subsequence composed of all such cycles. Then by Lemma 3.4 and by the nature of type 2 cycles, no two type 2 cycles can appear consecutively in γ. Hence γ must contain infinitely many cycles of type 3 or 4. □

Consider a state $s = (1, 1; \gamma_1, \gamma_2)$ in $\mathrm{RS}(S, s_0)$. We wish to extract from s a configuration of M. Adapting the corresponding notion from [4], we thus define *contour*$(s) = \sigma_1\gamma_j\sigma_2\#$, where $\{i, j\} = \{1, 2\}$ and γ_i is the unique channel content expressible as $\sigma_2\#\sigma_1$.

Proposition 3.8 *Let $s_0 \to s_1 \to \cdots \to s_n$ be a sequence with no interrupted cycle, in which there are k occurrences of a cycle of type 3 or 4. Then there exist $i_1 < i_2 < \ldots < i_k$ such that $C_j = contour(s_{i_j})$ for $1 \le j \le k$, and*

$$C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \cdots \vdash_M C_k.$$

Proof. By induction on k. In the inductive step, we use the fact that contour is not affected by copying cycles, and that the execution of one cycle of type 2 maintains contour as a faithful representation of the configuration of M attained inductively. Hence the next cycle of type 3 or 4 encountered prescribes the next legal transition of M. □

3.3 Undecidability consequences

Theorem 3.9 *The boundedness problem and the finite reachability tree problem for a system of two identical initial CFSMs are undecidable.*

Proof. In the terminology of Sections 3.1 and 3.2, we prove that the following are equivalent:

1. M accepts w,
2. $\mathrm{RT}(S(M), s_0(M, w))$ is finite,
3. $\mathrm{RS}(S(M), s_0(M, w))$ is finite.

(1) $\Rightarrow$ (2): Let M accept w. Suppose to the contrary that $\mathrm{RT}(S, s_0)$ is infinite. Then some branch in $\mathrm{RT}(S, s_0)$ is infinite. Therefore, by Lemma 3.5 and Corollary 3.7, $\mathrm{RT}(S, s_0)$ contains an infinite branch with no interrupted cycle and with an infinite number of transitions of type 3 or 4. But then Proposition 3.8 implies the existence of an infinite computation of M from configuration $q_0w\#$. This is a contradiction since M accepts w.
(2) $\Rightarrow$ (3): Immediate.
(3) $\Rightarrow$ (1): Let $\mathrm{RS}(S, s_0)$ be finite. Suppose to the contrary that M rejects w. Then $C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \cdots$ extends indefinitely. By our Turing machine assumptions, all the C_i's are distinct. Hence, by Proposition 3.1, there exists a branch in $\mathrm{RT}(S, s_0)$ having infinitely many distinct states. This is a contradiction.

The halting problem for Turing machines therefore reduces to the reachability tree finiteness problem, and it reduces to the boundedness problem (both via a many-one reduction). Hence the latter two problems are undecidable. $\Box$

4 Indistinguishable initial CFSMs

In the notation of Section 3, here we show how to implement one-time mutual exclusion and thus construct an initial global state in which initial local states are identical and initial channels are empty.

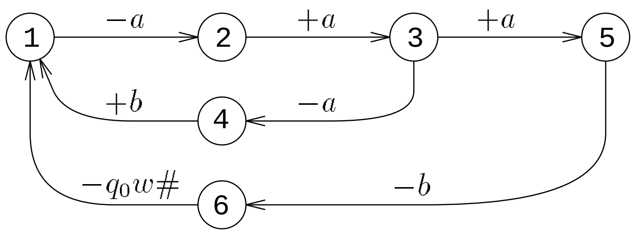
The idea is to add initializing cycles to the CFSMs constructed in Section 3. The difficulty is to prevent these new cycles from creating havoc in the rest of the simulation, keeping in mind that the two CFSMs constructed must remain identical. There are two initializing cycles, each of which is added to each of the two CFSMs. In our diagram we partly overlap the two cycles in order to make the correctness proof more manageable, but the two cycles can be thought of as meeting only at local state 1. Symbols a and b are new symbols never encountered before.

First initializing cycle:

1. `Emit:` a
2. `Receive:` a
3. `Emit:` a
4. `Receive:` b

Second initializing cycle:

1. `Emit:` a
2. `Receive:` a
3. `Receive:` a
4. `Emit:` b
5. `Emit:` $q_0w\#$



We still write $S = (A_1, A_2)$ for the resulting system, and we define $s_\lambda = (1, 1; \lambda, \lambda)$. Intuitively, both CFSMs cannot engage from s_λ into the first initialization cycle because the system would block on $+b$. On the other hand, when a CFSM A_i engages into the second cycle, its mate must have engaged far enough into the first cycle to produce two consecutive a. At this point, A_i emits a b to release its mate, and A_i sets up its mate's input channel for its mate to begin the simulation proper. From then on, except in harmless transient situations, the two communicating channels are never simultaneously empty, so that any attempt to reexecute an initializing cycle quickly blocks the system. The next lemma makes this formal.

Lemma 4.1 *For any sufficiently long branch σ in RT(S, s_λ), there exists a branch $\sigma_1\sigma_2 \sim \sigma$ in RT(S, s_λ) such that*

1. $s_\lambda \xrightarrow{\sigma_1} (1, 1; q_0w\#, \lambda)$ *or* $s_\lambda \xrightarrow{\sigma_1} (1, 1; \lambda, q_0w\#)$*, and*
2. $x \not\sqsubseteq \sigma_2$ *for any transition x belonging to an initialization cycle.*

Proof. Consider the top part of RT(S, s_λ), depicted on Figure 1. Any nonblocking branch σ out of s_λ either has a prefix σ_1 that satisfies the first

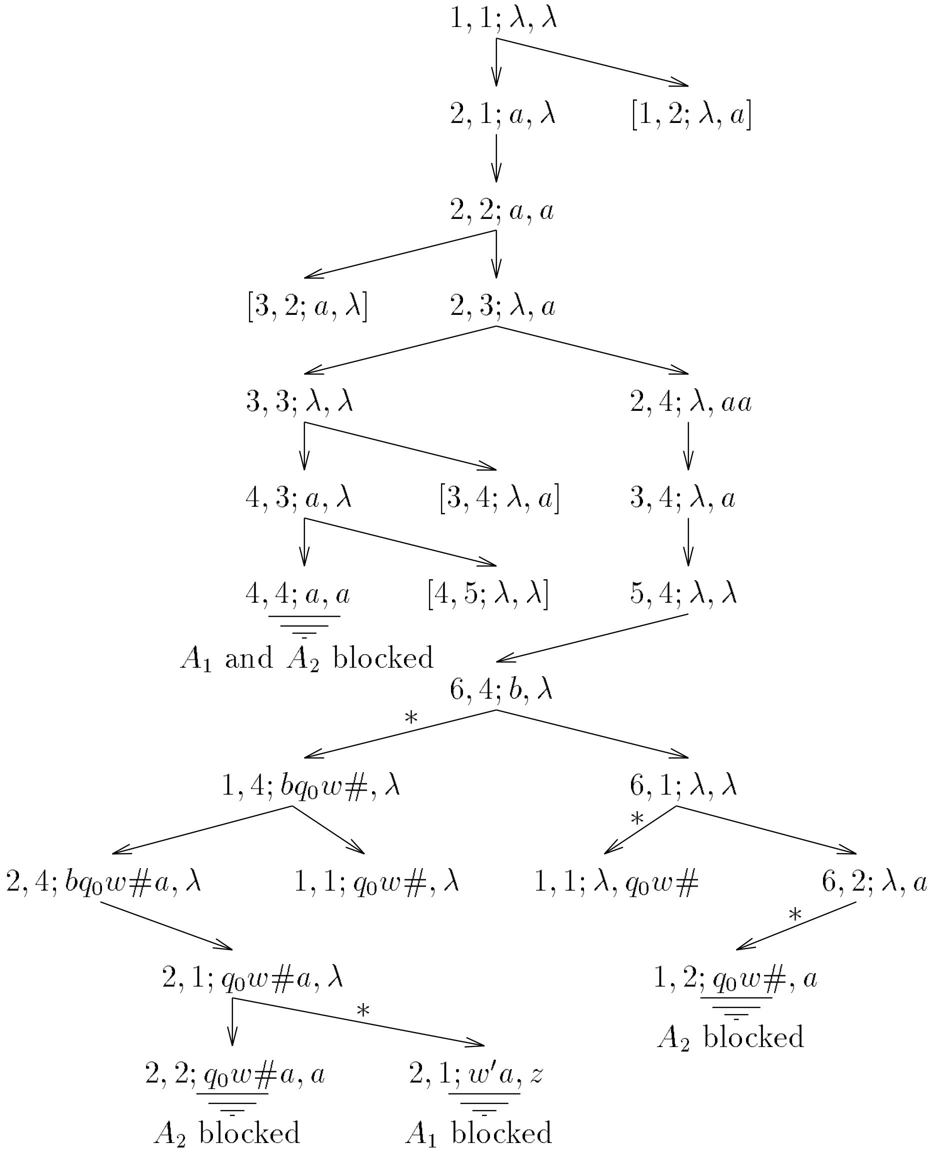


Figure 1: Initialization portion of $\mathrm{RT}(S,s_\lambda)$. Local states are those in the initialization cycles. $[s]$ indicates that the state s is the dual of a state explored elsewhere in the tree, z is a symbol different from a, and "blocked" means "blocked forever".

condition, or it has a prefix which differs from such a σ_1 in that one CFSM has begun consuming $q_0w\#$ which its mate has not finished emitting. In the latter situation the prefix can be rearranged into a σ_1 satisfying the first condition.

Now consider an attempt to reinitialize within the suffix $\sigma_2 = \sigma - \sigma_1$. Let $\sigma_2 = \gamma_1 x \gamma_2$ where x is the first occurrence of the first transition (emission of a) in one of the two initialization cycles. Then, by Lemma 3.4, A_i for some $i \in \{1, 2\}$ is active immediately after γ_1; in particular, A_i's input channel is nonempty or A_i is in a head processing state.

If x is an A_i transition, then A_i's input channel is nonempty and A_i is blocked forever on $+a$, so σ cannot be sufficiently long. On the other hand, if x is an A_j transition, $j \neq i$, then A_j's input channel must have been empty before x (otherwise A_j itself blocks forever on $+a$). Hence the first A_i emission in γ_2 must be a. This forces A_i in γ_2 to engage into an initialization cycle on its nonempty input channel. Hence A_i blocks forever on $+a$, once again contradicting the length of σ. □

Theorem 4.2 *The boundedness problem and the finite reachability tree problem for a system of two CFSMs are undecidable, even when the CFSMs are taken to be initial, identical, with identical starting states and with empty initial channel contents.*

Proof. The proof of Theorem 3.9 applies as well when the initializing cycles described in this section are added and when the initial global state is s_λ rather than s_0. Indeed it is easy to see that an analog of Proposition 3.1 holds here. For the converse, Lemma 4.1 guarantees that any infinite branch in $\mathrm{RT}(S, s_\lambda)$ can be thought of as containing the global state s_0 (or its dual $(1, 1; \lambda, q_0w\#)$), and that the initializing cycles cannot interfere with an infinite branch out of s_0 or its dual, so that an analog of Proposition 3.8 holds as well. □

5 Conclusion

We have generalized the first result of Brand and Zafiropulo [4] by showing that, even under new natural constraints arising from the modelisation and the verification of distributed algorithms, the CFSM model remains intrinsically undecidable. More precisely, even if the communicating automata are identical, indistinguishable and initial, the finite reachability tree problem and the boundedness problem remain undecidable.

Our results strengthen Brand and Zafiropulo's first result and confirm that in general, systems of CFSMs are extremely difficult to verify. It seems that the verification of such systems will require new formal test methods which, despite their partial coverage of all possible input situations, would nonetheless often allow full verification. An example of such a test would be to verify the Petri net naturally associated with a system of identical CFSMs and to extract from this necessary or sufficient conditions for its correctness.

Acknowledgements. We thank Hervé Caussinus for comments on a preliminary version of this paper.

References

[1] P. Abdullah and B. Jonsson, Verifying programs with unreliable channels, *Proc. of Logic in Computer Science,* (1993).

[2] P. Abdullah and B. Jonsson, Undecidability of verifying programs with unreliable channels, *Proc. of ICALP,* (1994).

[3] A. Benslimane, Deciding boundedness for systems of two communicating finite state machines, *Proc. of the Third IEEE Intern. Symp. on High Performance Distributed Computing* (1994), pp. 262-269.

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

[5] D. Brand and P. Zafiropulo, On communicating finite state machines, *J. of the Association for Computing Machinery* **30** (2), pp. 1983.323-342

[6] CCITT Recommendation Z.100: *Specification and description language SDL,* Blue Book Vol X.1-X.5 (1988), ITU General Secretariat, Geneva.

[7] G. Cécé, A. Finkel, and S. Purushothaman Iyer, Duplication, insertion and lossiness errors in unreliable communication channels, *Proc. of the Second ACM SIGSOFT* (1994), pp. 35-43.

[8] M. Diaz, J.P. Ansart, P. Azena, and V. Chari, *The formal description technique Estelle,* North Holland, 1989.

[9] A. Finkel, Decidability of the termination problem for completely specified protocols, *Distributed Computing* **7** (1994), pp. 129-135.

[10] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation,* Addison-Wesley (1979).

[11] G. Memmi and A. Finkel, An introduction to Fifo nets - Monogeneous nets: a subclass of Fifo nets, *Theoretical Computer Science* **35** (1985), pp. 191-214.

[12] G. Tel, *Introduction to Distributed Algorithms* (1994), Cambridge University Press.

[13] B. Vauquelin and P. Franchi-Zannettacci, Automates à files, *Theoretical Computer Science* **80** (1980), pp. 221-225.