arXiv:cs/0306121v2 [cs.LO] 20 Mar 2012

REACHABILITY PROBLEMS FOR COMMUNICATING FINITE STATE MACHINES*

Jan K. Pachl
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

Research Report CS-82-12

May 1982

* This research was supported by the Natural Sciences and Engineering Research Council of Canada under grant No. A7403.

Table of contents

1. Introduction
2. Introductory examples
3. Communicating finite state machines
4. Reachability properties
5. Reachability analysis and abstract flow control
6. Affine SR-machines
7. Undecidable problems
8. Rational channels for cyclic protocols
9. Recognizable channels for general protocols
10. Abstract flow control in general graphs
11. Recapitulation and conclusions

Appendix: Post's tag systems

This is a newly formatted version of the report. Page numbers differ from those in the original version. Several typographical errors have been corrected.

1 Introduction

This paper is about a state transition model for communication protocols.

The protocols governing data communication in computer systems are becoming ever more complex, and therefore more difficult to design, understand and analyze. This leads a number of researchers to advocate the use of formal methods for description and analysis of protocols [Bo1, Bo2].

State transition models are often used to describe formally (certain aspects of) communication protocols. This paper is concerned with a state transition model in which stations (modelled by finite state machines) communicate by exchanging messages, which are subjected to *unpredictable and unbounded delays.* (Thus transitions in the finite state machines are loosely coupled, in contrast to the directly coupled transitions of [Bo2].) The communication channels function as *potentially unbounded* FIFO queues.

An attractive feature of state transition models is that various general properties (called "syntactic properties" in [Zaf]) can be automatically verified if the queues (channels) are bounded. On the other hand, Brand and Zafiropulo [Bra] show that the verification of the same properties *cannot* be automated for general collections of communicating finite state machines connected by unbounded queues.

This paper investigates the question of decidability (algorithm existence) in some detail, and concentrates on a class of communicating finite state machines in which certain general properties are algorithmically decidable, although the queues are not necessarily bounded. (Thus our goal is similar to that of [Bra], but our methods and results are different.) The technique proposed in this paper is the third stage in the following hierarchy of formalisms for protocol description. (All three stages will be exemplified in the next section.)

- The list of all interactions.
- Communicating finite state machines (CFSM).
- CFSM augmented with channel expressions.

The paper is organized as follows: Section 2, which is a continuation of this introduction, contains several examples. In section 3, where the formalism begins, communicating finite state machines (CFSM) are defined. Section 4 lists various properties that can be defined in the CFSM model. Section 5 introduces two basic techniques for analyzing CFSM protocols, the exhaustive reachability analysis and abstract flow control. Section 6 shows that certain properties of SR-machines are decidable, although they are seemingly similar to the properties proved undecidable in section 7. In section 7 we shall see that most of the interesting properties in the CFSM model are undecidable (cf. [Bra]). For example, there is no algorithm to decide whether a protocol is deadlock-free.

It is then natural to ask: When can we *prove* that a protocol is deadlock-free? A simple proof formalism is offered and investigated in sections 8, 9 and 10. Its virtue is its simplicity, which allows straightforward automatic proof checking. Not every deadlock-free protocol can be proved to be deadlock-free in the formalisms (nor in any other formalism, in view of the undecidability result), but the method applies to the protocols that "use their channels in a simple manner". Section 8 presents a simple version of the formalism, applicable to the protocols consisting of finite state machines arranged in a circle. A more general theory is presented in section 9. Section 10 generalizes the results of section 5 about abstract flow control, and concludes with several decidability results.

2 Introductory examples

This section presents three examples to illustrate the three methods of protocol description listed in the introduction.

2.1 Description by listing all interactions

A simple access authorization protocol (adapted from [Zaf], p. 652), allowing only two interactions (communication histories), is depicted in Fig. 2.1(a) and Fig. 2.1(b).

The description method is straightforward and easy to understand, and a simple matching algorithm will discover deadlocks, unspecified receptions etc. However, the protocols that allow infinitely many (or a very large number of) communication histories cannot be completely described.

2.2 Description by communicating finite state machines

Stations (processes) are represented by finite state machines whose transitions correspond to transmissions and receptions of messages. E.g. the protocol of Fig. 2.1 can be described as shown in Fig. 2.2 (cf. Fig. 1 in [Zaf]).

Since the finite state machines can contain cycles, some protocols that allow infinitely many message sequences can be described this way. Deadlock-freedom and other general properties are algorithmically verifiable, provided there is an upper bound on the number of messages that can be simultaneously in transit. This finiteness condition, which is far weaker than the one in 2.1, is further substantially relaxed in 2.3 below, at the cost of making the description more elaborate.

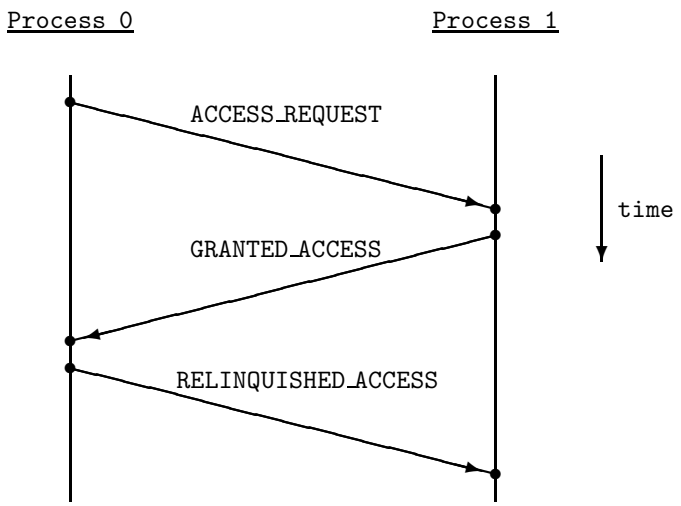


Fig. 2.1(a).

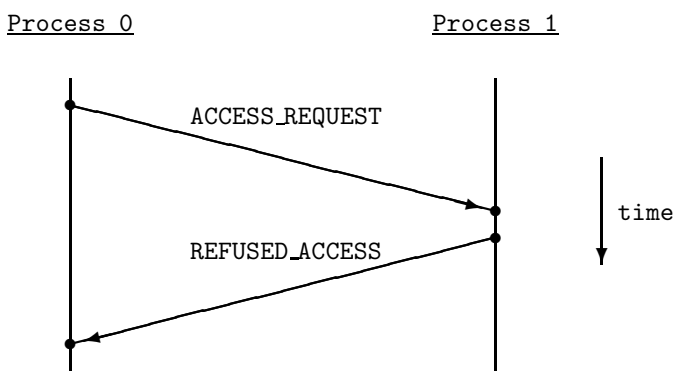


Fig. 2.1(b).

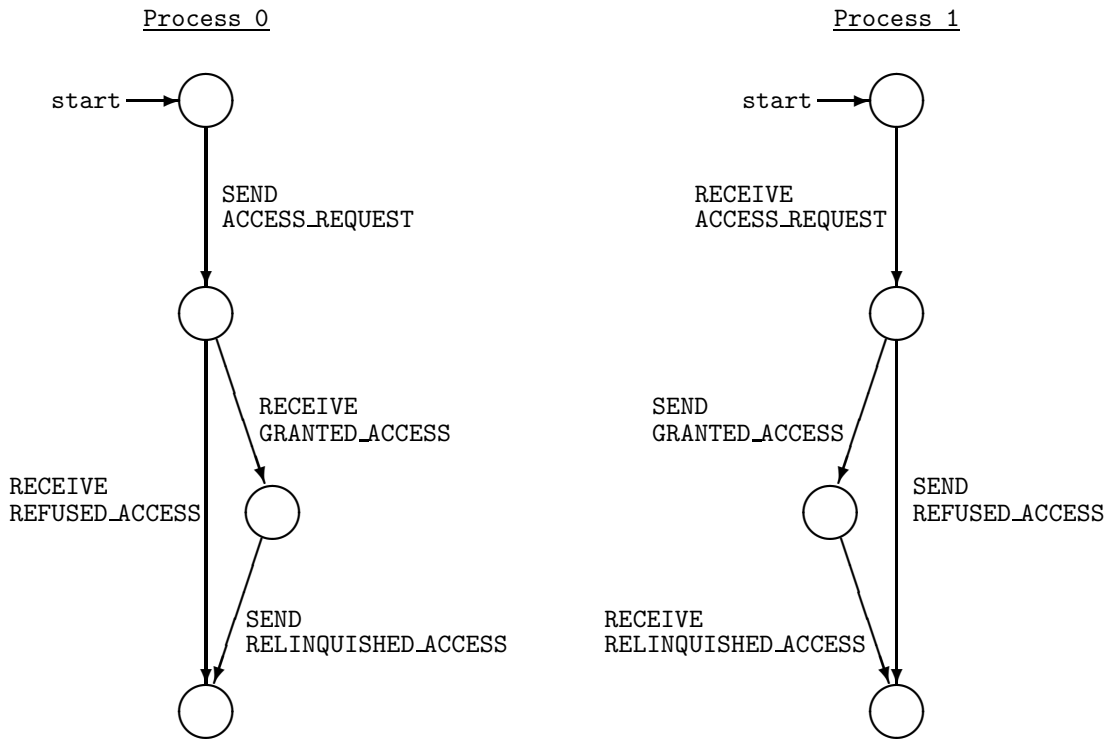

Process 0
start
SEND
ACCESS_REQUEST
RECEIVE
GRANTED_ACCESS
RECEIVE
REFUSED_ACCESS
SEND
RELINQUISHED_ACCESS
Process 1
start
RECEIVE
ACCESS_REQUEST
SEND
GRANTED_ACCESS
SEND
REFUSED_ACCESS
RECEIVE
RELINQUISHED_ACCESS

Fig. 2.2.

2.3 Communicating finite state machines augmented by channel expressions

This is an extension of the model in 2.2. The protocol designer is required to provide not only the finite state machines representing the processes, but also a complete description of channel content for each combination of states. In this paper we consider such a model, in which the channel content is described by rational expressions.

Example. A simple alternating-bit protocol for transmission over unreliable channels can be described as in Fig. 2.3. There are six message types used in the protocol:

`EV`	even data packet
`OD`	odd data packet
`ED`	end of data
`EVA`	acknowledgement of `EV`
`ODA`	acknowledgement of `OD`
`EDA`	acknowledgement of `ED`

Receptions are denoted by $+$ and transmissions by $-$. Following the suggestion in [Zaf], we describe the unreliable channels by two additional finite state machines, depicted in Fig. 2.4. We think of all errors on the channel as being concentrated in one place, under the control of a demon. The rest of the channel then functions as a perfect FIFO queue.

Fig. 2.4 makes precise what we mean by an unreliable channel: The demon retransmits some of the messages it receives, and ignores (deletes) others.

The complete model now consists of four finite state machines connected by four channels, as in Fig. 2.5.

Since Process 0 can repeatedly send the message `EV`, `OD` or `ED`, there is no upper bound on the number of messages that can be simultaneously in transit. Thus the description developed so far, although completely specifying all interactions, does not easily submit to analysis. We will aid the analysis by describing all the channel contents that can occur for each combination of states. Since the model has four state machines with four states each, the additional information will be in the form of a table with $4 \times 4 \times 4 \times 4 = 256$ entries (one for each state combination), each entry being the set of all channel contents that can coexist with the state combination. As the model has four channels, a set of channel contents is a 4-ary *relation*. All 256 relations in this example are rational, i.e. they can be described by rational expressions. Fig. 2.6 lists four of the 256 relations in question, namely those for the state combinations 00/10/20/30, 01/10/20/30, 02/10/20/30 and 03/10/20/30. In fact, it is sufficient to specify these four entries; the remaining 252 can be automatically computed.

In Fig. 2.6, $\texttt{ED}_\alpha$ is the symbol `ED` in the channel α, $\texttt{ED}_\beta$ is the symbol `ED` in the channel β, etc. By using the subscripts we make the channel alphabets disjoint, and avoid ambiguity in the channel expressions.

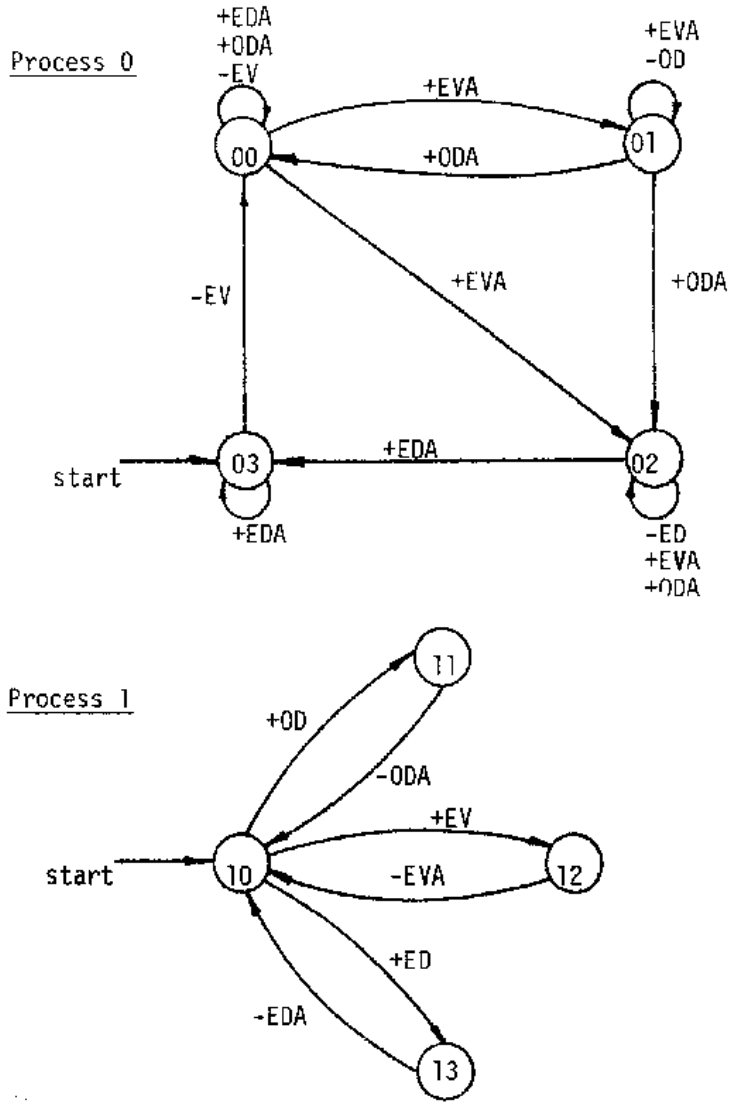


Fig. 2.3. A simple alternating-bit protocol.

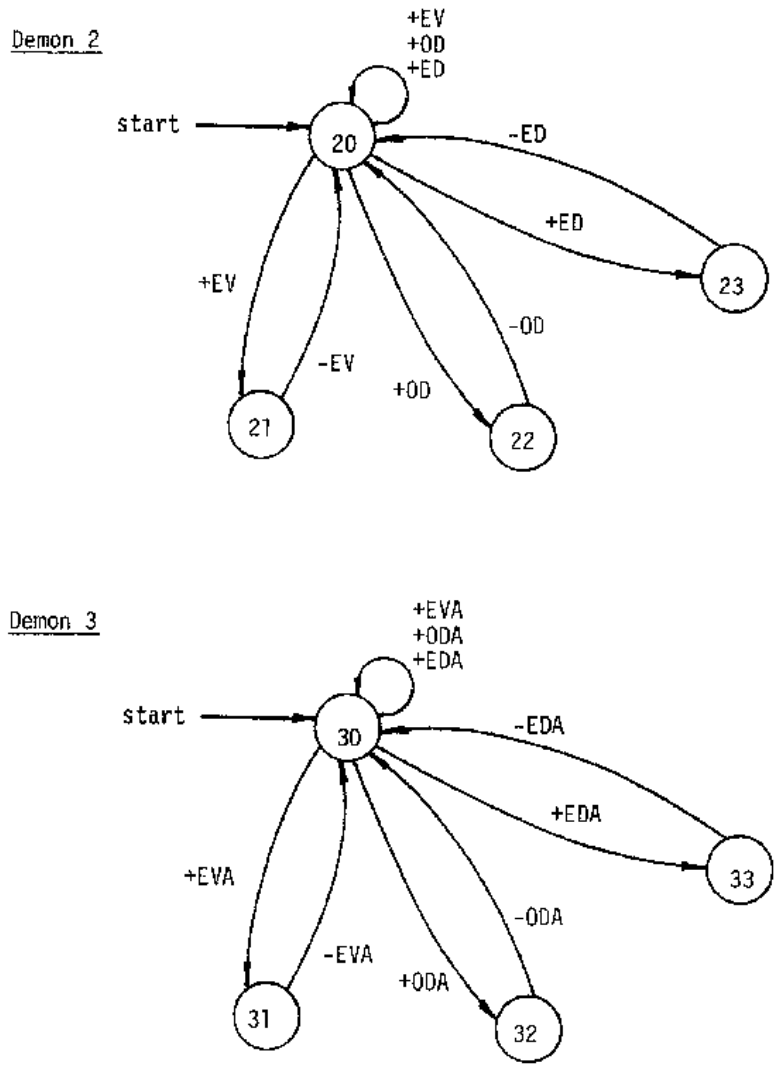


Fig. 2.4. Unreliable channels modelled by demons.

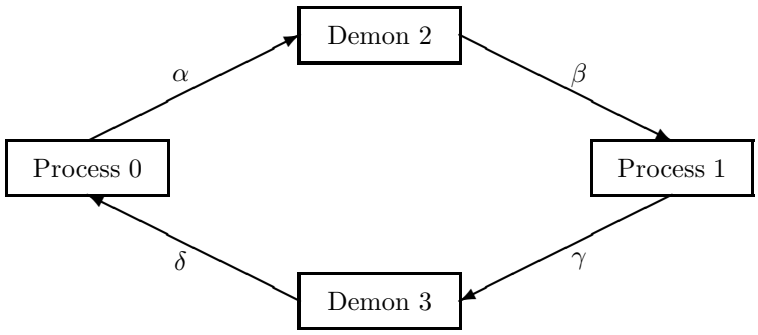


Fig. 2.5. The communication graph.

Composite state	Channel contents
00/10/20/30	$(\ \mathtt{ED}^*_\alpha\ \mathtt{EV}^*_\alpha\ \mathtt{ED}^*_\beta \cup \mathtt{EV}^*_\alpha\ \mathtt{ED}^*_\beta\ \mathtt{EV}^*_\beta\ )\ \mathtt{EDA}^*_\gamma\ \mathtt{EDA}^*_\delta \cup$ $\mathtt{EV}^*_\alpha\ \mathtt{EV}^*_\beta\ (\ \mathtt{EDA}^*_\gamma\ \mathtt{EVA}^*_\gamma\ \mathtt{EDA}^*_\delta \cup \mathtt{EVA}^*_\gamma\ \mathtt{EDA}^*_\delta\ \mathtt{EVA}^*_\delta\ ) \cup$ $(\ \mathtt{OD}^*_\alpha\ \mathtt{EV}^*_\alpha\ \mathtt{OD}^*_\beta \cup \mathtt{EV}^*_\alpha\ \mathtt{OD}^*_\beta\ \mathtt{EV}^*_\beta\ )\ \mathtt{ODA}^*_\gamma\ \mathtt{ODA}^*_\delta \cup$ $\mathtt{EV}^*_\alpha\ \mathtt{EV}^*_\beta\ (\ \mathtt{ODA}^*_\gamma\ \mathtt{EVA}^*_\gamma\ \mathtt{ODA}^*_\delta \cup \mathtt{EVA}^*_\gamma\ \mathtt{ODA}^*_\delta\ \mathtt{EVA}^*_\delta\ )$
01/10/20/30	$(\ \mathtt{EV}^*_\alpha\ \mathtt{OD}^*_\alpha\ \mathtt{EV}^*_\beta \cup \mathtt{OD}^*_\alpha\ \mathtt{EV}^*_\beta\ \mathtt{OD}^*_\beta\ )\ \mathtt{EVA}^*_\gamma\ \mathtt{EVA}^*_\delta \cup$ $\mathtt{OD}^*_\alpha\ \mathtt{OD}^*_\beta\ (\ \mathtt{EVA}^*_\gamma\ \mathtt{ODA}^*_\gamma\ \mathtt{EVA}^*_\delta \cup \mathtt{ODA}^*_\gamma\ \mathtt{EVA}^*_\delta\ \mathtt{ODA}^*_\delta\ )$
02/10/20/30	$(\ \mathtt{EV}^*_\alpha\ \mathtt{ED}^*_\alpha\ \mathtt{EV}^*_\beta \cup \mathtt{ED}^*_\alpha\ \mathtt{EV}^*_\beta\ \mathtt{ED}^*_\beta\ )\ \mathtt{EVA}^*_\gamma\ \mathtt{EVA}^*_\delta \cup$ $\mathtt{ED}^*_\alpha\ \mathtt{ED}^*_\beta\ (\ \mathtt{EVA}^*_\gamma\ \mathtt{EDA}^*_\gamma\ \mathtt{EVA}^*_\delta \cup \mathtt{EDA}^*_\gamma\ \mathtt{EVA}^*_\delta\ \mathtt{EDA}^*_\delta\ ) \cup$ $(\ \mathtt{OD}^*_\alpha\ \mathtt{ED}^*_\alpha\ \mathtt{OD}^*_\beta \cup \mathtt{ED}^*_\alpha\ \mathtt{OD}^*_\beta\ \mathtt{ED}^*_\beta\ )\ \mathtt{ODA}^*_\gamma\ \mathtt{ODA}^*_\delta \cup$ $\mathtt{ED}^*_\alpha\ \mathtt{ED}^*_\beta\ (\ \mathtt{ODA}^*_\gamma\ \mathtt{EDA}^*_\gamma\ \mathtt{ODA}^*_\delta \cup \mathtt{EDA}^*_\gamma\ \mathtt{ODA}^*_\delta\ \mathtt{EDA}^*_\delta\ )$
03/10/20/30	$\mathtt{ED}^*_\alpha\ \mathtt{ED}^*_\beta\ \mathtt{EDA}^*_\gamma\ \mathtt{EDA}^*_\delta$

Fig. 2.6. Rational expressions for channel contents.

3 Communicating finite state machines

The present paper treats communicating finite state machines as mathematical objects. They are formally defined in this section. The formalism is fairly close to that in [Bra].

A *directed graph* is a pair $G = (N, E)$ where N and E are two sets (the set of *nodes* and the set of *edges*), together with two maps, denoted $\xi \mapsto -\xi$ and $\xi \mapsto +\xi$, from E to N. We say that $-\xi$ is the *tail* and $+\xi$ the *head* of the edge ξ; when $i = -\xi$ and $j = +\xi$, we sometimes write $i \xrightarrow{\xi} j$. We say that G is finite if both N and E are finite.

A *protocol* (or, more explicitly, a *CFSM protocol*) $\mathbf{P}$ consists of a finite directed graph $G = (N, E)$ (the *communication graph* of $\mathbf{P}$), a collection of pairwise disjoint finite sets M_ξ indexed by $\xi \in E$, and a collection of finite state machines F_j indexed by $j \in N$. Each F_j operates over the alphabet

$$\Sigma_j = \{ +b \mid b \in M_\xi,\ j = +\xi \} \cup \{ -b \mid b \in M_\xi,\ j = -\xi \} .$$

Specifically, $F_j = (K_j, \Sigma_j, T_j, h_j)$ where K_j is the finite *set of states*, $h_j \in K_j$ is the *initial* (or *home*) *state*, and $T_j \subseteq K_j \times \Sigma_j \times K_j$ is the *set of transitions*.

We write $p \xrightarrow{e} q$ in F_j (or simply $p \xrightarrow{e} q$, if no misunderstanding is possible) when $(p, e, q) \in T_j$. (Here $e = -b$ or $e = +b$, for some ξ and $b \in M_\xi$.) The *transition diagram of* F_j is the labelled directed graph with nodes K_j and labelled edges $p \xrightarrow{e} q$ for $(p, e, q) \in T_j$.

Write $p \xrightarrow{w} q$, for $w \in \Sigma_j^*$, if there is a directed path from p to q, in the transition diagram of F_j, such that the labels on the edges of the path form the string w (in the order from p to q). Sometimes we write $p \xrightarrow{e}$ instead of "$p \xrightarrow{e} q$ for some q", and similarly $p \xrightarrow{w}$ for $w \in \Sigma_j^*$.

The model corresponds to reality in this way: The graph G describes the protocol configuration (the edges are unidirectional communication channels); we say that the machines F_j in $\mathbf{P}$ *communicate according to* G. The set M_ξ is the set of messages that can be sent along the channel ξ (in practice these sets need not be disjoint, but the assumption that they are causes no loss generality and is technically useful). The machine F_j represents a process located at $j \in N$ and capable of sending messages to the channels ξ such that $j = -\xi$ and of receiving messages from the channels ξ such that $j = +\xi$. Message transmissions and receptions match transitions in the state machines: $p \xrightarrow{+b} q$ in F_j means $b \in M_\xi$ received, and $p \xrightarrow{-b} q$ in F_j means $b \in M_\xi$ sent (at $j \in N$ along $\xi \in E$).

In the sequel we shall have an opportunity to deal with CFSM protocols of a special form, the SR-machines of Gouda [Gou]:

A state $p \in K_j$ is a *send state* if $p \xrightarrow{+b}$ for no b; similarly p is a *receive state* if $p \xrightarrow{-b}$ for no b. Say that F_j is an *SR-machine* if

(a) K_j has only send and receive states,

(b) the transition diagram of F_j is strongly connected, and

(c) if $p \xrightarrow{e} q_1$ and $p \xrightarrow{e} q_2$ in F_j then $q_1 = q_2$.

A *pair of communicating SR-machines* is a protocol with two SR-machines F_0 and F_1 communicating according to the graph $0 \underset{\beta}{\overset{\alpha}{\rightleftarrows}} 1$ (i.e. $N = \{0,1\}$, $E = \{\alpha, \beta\}$, $-\alpha = +\beta = 0$ and $+\alpha = -\beta = 1$).

Other variations of communicating finite state machines have been employed to describe and analyze communication protocols, but the differences between them are not essential in the present context. The popularity of the model stems from the fact that, while being simple and abstract, it is rich enough to embrace some general communication properties (sometimes called syntactic properties). Several such properties are enumerated in the next section. They are all defined in terms of the global state space, which we now proceed to describe.

In our basic model, we assume that the channels function as perfect FIFO queues. That is, they are error-free (imperfect channels are modelled indirectly, by demons), and in each channel messages are received in the same order as sent. We place no a priori bound on the queue lengths; the intention is to model unpredictable and unbounded communication delays.

Let $\mathbf{P}$ be a CFSM protocol, with the notation as above. A *composite state* of $\mathbf{P}$ is a vector $S = (p_j : j \in N)$ of states $p_j \in K_j$. A *channel content* (or "composite channel state") is a vector $C = (x_\xi : \xi \in E)$ of strings $x_\xi \in M_\xi^*$ (each x_ξ is a string over the alphabet M_ξ). A *global state* is a pair (S, C) where S is a composite state and C is a channel content. The *initial global state* is (S^0, C^0) where $S^0 = (h_j : j \in N)$ and $C^0 = (x_\xi : \xi \in E)$ with each x_ξ being the empty string λ.

Our aim is to define a labelled directed graph whose nodes will be global states of $\mathbf{P}$ and which will have two kinds of labelled edges (write $S = (p_j : j \in N)$, $S' = (q_j : j \in N)$, $C = (x_\xi : \xi \in E)$, $C' = (y_\xi : \xi \in E)$):

(1) (Receive from channel β)

$$(S, C) \overset{+b}{\vdash} (S', C')$$

if there are i and β with $i = +\beta$, such that $p_j = q_j$ for $j \neq i$, $p_i \overset{+b}{\to} q_i$ in F_i, $x_\xi = y_\xi$ for $\xi \neq \beta$, and $x_\beta = by_\beta$.

(2) (Send to channel β)

$$(S, C) \overset{-b}{\vdash} (S', C')$$

if there are i and β with $i = -\beta$, such that $p_j = q_j$ for $j \neq i$, $p_i \overset{-b}{\to} q_i$ in F_i, $x_\xi = y_\xi$ for $\xi \neq \beta$, and $y_\beta = x_\beta b$.

Write $(S, C) \vdash (S', C')$ if $(S, C) \overset{+b}{\vdash} (S', C')$ or $(S, C) \overset{-b}{\vdash} (S', C')$ for some b. Let $\vdash^*$ be the reflexive and transitive closure of $\vdash$. Say that a global state (S', C') is *reachable* from a global state (S, C) if $(S, C) \vdash^* (S', C')$.

Say that a global state is *reachable* if it is reachable from (S^0, C^0). The *global state space* of the protocol $\mathbf{P}$ is the labelled directed graph whose nodes are all the reachable global states of $\mathbf{P}$, with labelled edges $\overset{+b}{\vdash}$ and $\overset{-b}{\vdash}$ defined above.

4 Reachability properties

The general reachability problem, in its simplest form, is "Given a (possibly infinite) directed graph and two of its nodes, can one node be reached from the other along a path in the graph?" One may wish to construct an algorithm to answer the question; this leads to a decidability problem: Is there an algorithm to decide, for any given graph and two nodes, whether one can be reached from the other? In other words, is the reachability problem (algorithmically) decidable?

Algorithms to solve two problems of this kind have been found recently, after a prolonged research effort: Kannan and Lipton [Kan] constructed an algorithm to solve Harrison's orbit problem, and Mayr [May] constructed an algorithm for the Petri net reachability problem. The CFSM model brings up another reachability problem, which is, unlike the previous two, undecidable (see section 7). However, it is worthwhile to investigate restrictions on the problem that make it decidable; this is the chief subject of the present paper.

In fact, there is not one but a number of reachability problems in the CFSM model. The (possibly infinite) directed graph where they all reside is the global state space defined in the previous section.

A *simple reachability problem* (or a reachability problem of the first order) has the form "Is a given global state reachable (from (S^0, C^0))?" For example, the problem of finding stable composite states can be treated as a simple reachability problem: A composite state S is called *stable* if (S, C^0) is reachable; cf. [Zaf], [Bra]. Since there are only finitely many composite states, the problem of listing all stable ones is solved by answering finitely many simple reachability problems.

A global state (S, C) is said to be *deadlocked* if every state in S is a receive state and $C = C^0$. The protocol $\mathbf{P}$ is *deadlock-free* if no deadlocked global state is reachable. The question whether $\mathbf{P}$ is deadlock-free is a special case of the stable composite state problem in the previous paragraph.

However, there are other pertinent reachability problems that are not simple in this sense (or at least it is not immediately obvious if they are). Say that $b \in M_\beta$ *can arrive at the state* $p \in K_i$ if $i = +\beta$ and there is a reachable global state $((p_j : j \in N), (x_\xi : \xi \in E))$ such that $p = p_i$ and $x_\beta = by_\beta$ for some $y_\beta \in M_\beta^*$. The problem of finding all pairs (p, b) such that b can arrive at p ("executable receptions" in the terminology of [Bra]) is of the form "Is at least one element of a given *set* of global states reachable?" Let us call this a *second order reachability problem.* Of course, the set of global states in question can be described

in various ways; that can make the problem more or less difficult (or even decidable or undecidable).

Say that a global state (S, C) is *globally blocked* if $(S, C) \vdash (S', C')$ for no global state (S', C'). (Every deadlocked global state is globally blocked but not vice versa.)

A global state $(S, C) = (S, (x_\xi : \xi \in E))$ is *blocked on channel* $\beta \in E$ if $x_\beta = by$, $b \in M_\beta$, $y \in M_\beta^*$, and there are no global states (S', C') and (S'', C'') satisfying

$$(S, C) \vdash^* (S', C') \overset{+b}{\vdash} (S'', C'') \ .$$

The property that no reachable global state is blocked on any channel (that is, every transmitted message can be *eventually* received) should be compared with the following stronger property, defined in [Bra]. The protocol is *well-formed* if for any $p \in K_j$ and $b \in M_\beta$ we have: b can arrive at p if and only if $p \overset{+b}{\to}$ in F_j. This means that the protocol is able to receive every message *immediately* upon arrival and, moreover, the transition diagram of F_j has no useless edges.

Another example of a second order reachability property: A protocol with the communication graph $0 \underset{\beta}{\overset{\alpha}{\rightleftarrows}} 1$ is said to have the *half-duplex property* if every reachable global state $((p_0, p_1), (x_\alpha, x_\beta))$ satisfies $x_\alpha = \lambda$ or $x_\beta = \lambda$.

Finally, certain useful reachability properties are neither first nor second order. The protocol **P** has the *bounded channel property* if there is an upper bound on the total length of all strings in C, over all reachable global states (S, C). Obviously **P** has this property if and only if the global state space is finite.

We can see that, although the CFSM model is very simple and general, it allows us to formulate a number of meaningful protocol properties. Moreover, the properties are all described in a uniform manner, as reachability properties in a certain (potentially infinite) graph. The next question is whether the properties can be algorithmically decided. In this paper we concentrate on the deadlock problem ("Is the protocol deadlock-free?"), and the stable composite state problem ("Is a given composite state stable?"), two representatives of simple (first order) reachability problems. Occasionally we also note how the results apply to other reachability problems.

5 Reachability analysis and abstract flow control

When the global state space is finite, all reachability problems can be, at least in principle, algorithmically solved. Indeed, one can explicitly construct the global state space (as a finite directed graph) and search it to decide any reachability problem. We refer to this method as the *exhaustive reachability analysis.*

The method presents a number of implementation and complexity problems, because the global state space tends to be very large and exhaustive search is expensive. Nevertheless,

the question of algorithm *existence* is, in the case when the global state space is finite, uninteresting: All problems are decidable for trivial reasons. The chief aim of this paper is to investigate what can be done when the global state space is not (or is not known to be) finite.

The global state space has a highly redundant structure. Concurrent execution is modelled by a set of shuffles of sequential executions in the participating nodes. Thus if one global state is reachable from another then there are usually many paths between them. We can reduce the redundancy by restricting the order in which concurrent transmissions and receptions occur. This is the idea of the *abstract flow control.* Its special case was studied (under a different name) by Rubin and West [Rub].

Every path in the global state space defines "local paths" in the transition diagrams of the individual state machines. These will be called the *images* of the global path. In the notation of section 3, the image can be defined formally. Let $\Gamma = (S_0, C_0) \overset{e_1}{\vdash} \ldots \overset{e_k}{\vdash} (S_k, C_k)$ be a path in the global state space, and let $i \in N$. If $k = 0$ (i.e. the length of Γ is 0) and $S_0 = (p_j : j \in N)$ then $\mathrm{Im}_i(\Gamma)$, the image of Γ in F_i, is the path of length 0 from p_i to p_i. If $k > 0$, $S_{k-1} = (p_j : j \in N)$ and $S_k = (q_j : j \in N)$ then $\mathrm{Im}_i(\Gamma)$ is defined in terms of $\Gamma' = (S_0, C_0) \overset{e_1}{\vdash} \ldots \overset{e_{k-1}}{\vdash} (S_{k-1}, C_{k-1})$ as follows: If $e_k \notin \Sigma_i$ then $\mathrm{Im}_i(\Gamma) = \mathrm{Im}_i(\Gamma')$; if $e_k \in \Sigma_i$ then $\mathrm{Im}_i(\Gamma)$ is the concatenation of $\mathrm{Im}_i(\Gamma')$ with the path $q_i \overset{e_k}{\to} p_i$ (of length 1).

Say that two paths Γ and Γ' in the global state space are *locally equal* if $\mathrm{Im}_i(\Gamma) = \mathrm{Im}_i(\Gamma')$ for each $i \in N$. The following self-evident lemma is a basis of most that follows.

Lemma 5.1 *If two paths in the global state space are locally equal and start in the same global state, then they also terminate in the same global state.*

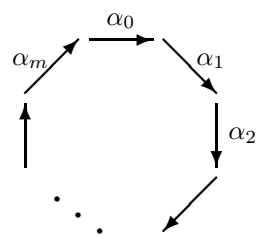
The aim of the abstract flow control, in the sense used in this paper, is to reduce the number of the locally equal paths that the reachability analysis must examine. Rubin and West [Rub] have shown how to select exactly one path in every set of locally equal paths, in the special case of two-party protocols and paths between global states of the form (S, C^0). The problem can be viewed as a scheduling problem: For a given path Γ in the global state space, the local images of Γ are concurrent sequential processes which must share a single processor. In this terminology, the Rubin and West method uses the round-robin scheduling. The methods explored in this paper are based on priority scheduling. They yield particularly simple results when the finite state machines are arranged in a circle; to have a short name for such CFSM protocols, we say that a protocol is *cyclic* if its communication graph is a directed cycle.

Theorem 5.2 *Let* $\mathbf{P}$ *be a cyclic CFSM protocol. Let* Γ *be a path in the global state space from a global state* (S, C^0) *to a global state* (S', C^0)*. If* β *is any edge in* E *then there exists a path* Γ' *such that*
(a) Γ *and* Γ' *are locally equal; and*
(b) every global state $(S, (x_\xi : \xi \in E))$ *on the path* Γ' *satisfies*

$$\sum_{\substack{\xi \in E \\ \xi \neq \beta}} |x_\xi| \ \leq \ 1 \ .$$

A more general result will be proved in section 10.

Proof. Label the edges of the communication graph G as $E = \{\alpha_0, \alpha_1, \ldots, \alpha_m\}$ and assume that $-\alpha_0 = +\alpha_m$, $-\alpha_1 = +\alpha_0$, $-\alpha_2 = +\alpha_1$, ..., $\beta = \alpha_0$:



Rearrange the execution described by Γ as follows: Assign the highest priority to the process running at the node $+\alpha_m$, the next highest to the process at $+\alpha_{m-1}$, etc., with the lowest priority at $+\alpha_0$. Thus a process can execute only if all processes with higher priorities are blocked (which means that their local images of Γ call for receptions and their input channels are empty). Let Γ' be the path corresponding to the priority execution. It follows that at most one among the channels α_1, α_2, ..., α_m is non-empty at any point along Γ', and that none can grow longer than one symbol. □

Theorem 5.2 (as well the more general results to come) simplifies the reachability algorithm. When looking for a deadlock, the algorithm can ignore the global states in which $\sum_{\xi \neq \beta} |x_\xi| > 1$. The following immediate corollary of Theorem 5.2 generalizes a result of Brand and Zafiropulo [Bra].

Corollary 5.3 *The stable composite state problem is decidable in the class of all cyclic CFSM protocols with this property: There is an edge* β *and a constant* c *such that every reachable global state* $(S, (x_\xi : \xi \in E))$ *satisfies* $|x_\beta| \leq c$*.*

It follows that deadlock-freedom is also decidable in this class.

6 Affine SR-machines

In this section we are going to see that certain properties of a pair of communicating SR-machines are algorithmically decidable, although they are superficially similar to the undecidable properties that we shall encounter later on.

Let $\mathbf{P}$ be a CFSM protocol consisting of two SR-machines $F_0 = (K_0, \Sigma_0, T_0, h_0)$ and $F_1 = (K_1, \Sigma_1, T_1, h_1)$ communicating according to the graph $0 \underset{\beta}{\overset{\alpha}{\rightleftarrows}} 1$. Recall that

$$\Sigma_0 \;=\; \{\, -b \mid b \in M_\alpha \,\} \;\cup\; \{\, +b \mid b \in M_\beta \,\}$$

and

$$\Sigma_1 \;=\; \{\, -b \mid b \in M_\beta \,\} \;\cup\; \{\, +b \mid b \in M_\alpha \,\}$$

If w is a string in Σ_0^* or Σ_1^*, denote by $\pi_\alpha(w)$ the string of all M_α symbols in w, in the same order; thus π_α erases all the symbols in w that belong to M_β, and also all $+$ and $-$ (π_α is the "projection" from $\Sigma_0^* \cup \Sigma_1^*$ onto M_α^*). The projection π_β onto M_β^* is defined similarly. For example, if $d_1, d_2 \in M_\alpha$ and $b_1, b_2 \in M_\beta$ then $\pi_\alpha(+d_1 + d_2 - b_1 + d_1 - b_2 - b_2) = d_1 d_2 d_1$ and $\pi_\beta(+d_1 + d_2 - b_1 + d_1 - b_2 - b_2) = b_1 b_2 b_2$.

The machine F_0 defines a subset $\mathbf{Z}_0$ of $M_\alpha^* \times M_\beta^*$:

$$\mathbf{Z}_0 \;=\; \{\, (\pi_\alpha(w), \pi_\beta(w)) \mid h_0 \xrightarrow{w} h_0 \;\text{ in }\; F_0 \,\} \,.$$

Similarly,

$$\mathbf{Z}_1 \;=\; \{\, (\pi_\alpha(w), \pi_\beta(w)) \mid h_1 \xrightarrow{w} h_1 \;\text{ in }\; F_1 \,\} \,.$$

Say that F_0 and F_1 are *affine* (or that the protocol $\mathbf{P}$ is affine) if $\mathbf{Z}_0 = \mathbf{Z}_1$.

Thus two SR-machines are affine if and only if for every sequence of sends and receives (beginning and ending in the "home state") in one machines there is a matching sequence in the other. However, the matching is a weak one because, intuitively, it allows a symbol to be received before it has been sent.

There are interesting connections between affinity and certain desirable protocol properties. At the same time, unlike the other properties, affinity is decidable; a minor modification of Bird's algorithm [Bir] establishes the following result.

Theorem 6.1 *There is an algorithm to decide whether an arbitrary pair of SR-machines is affine.*

Now we consider the bounded channel property for affine SR-machines. No protocol in which at least one machine can go through a cycle consisting of send transitions has the bounded channel property; the machine can repeat the sending cycle any number of times before the other machine begins receiving. The forthcoming theorem shows that for affine SR-machines the channel can grow large *only* if there is such a cycle.

Say that a state machine F_j has a *send cycle* if the transition diagram of F_j contains a directed cycle whose all labels are negative (i.e. of the form $-b$, $b \in M_\xi$, $j = -\xi$); a *receive cycle* is defined analogously.

Lemma 6.2 *Let F_0 and F_1 be two affine SR-machines. For $j = 0, 1$, let k_j be the number of states in F_j (=the cardinality of K_j). If there is a reachable global state $((p_0, p_1), (x_\alpha, \lambda))$ such that $|x_\alpha| \geq k_0(k_1 - 1) + 1$ then F_1 has a receive cycle and F_0 has a send cycle.*

This yields a new automatically verifiable *sufficient* condition for bounded channels, namely affinity and absence of send cycles; cf. [Bra] and [Gou] for other conditions of this kind. The condition is also *necessary* if the protocol is affine and deadlock-free:

Theorem 6.3 *Let F_0 and F_1 be two affine SR-machines. If the protocol is deadlock-free then it has the bounded channel property if and only if neither F_0 nor F_1 has a send cycle.*

Theorem 6.4 *There is an algorithm to decide, for an arbitrary given pair of affine SR-machines, whether the protocol is deadlock-free and has the bounded channel property.*

Another corollary of 6.2, to be proved later in this section, is the following:

Theorem 6.5 *There is an algorithm to decide, for an arbitrary given pair of SR-machines with no send cycles, whether the protocol is affine and deadlock-free.*

Now we prove the results in this section. Recall that we deal with a protocol **P** with the communication graph $0 \underset{\beta}{\overset{\alpha}{\rightleftarrows}} 1$ and two SR-machines $F_j = (K_j, \Sigma_j, T_j, h_j), j = 0, 1$. The channel alphabets are M_α and M_β.

Proof of 6.2. Recording how the global state $((p_0, p_1), (x_\alpha, \lambda))$ has been reached, we find two strings $w_0 \in \Sigma_0^*$ and $w_1 \in \Sigma_1^*$ such that $h_0 \overset{w_0}{\to} p_0$, $h_1 \overset{w_1}{\to} p_1$, $\pi_\alpha(w_0) = \pi_\alpha(w_1)x_\alpha$ and $\pi_\beta(w_0) = \pi_\beta(w_1)$. Since the transition graph of F_0 is strongly connected and has k_0 nodes, $p_0 \overset{u_0}{\to} h_0$ for some $u_0 \in \Sigma_0^*$ such that $|u_0| \leq k_0 - 1$.

By affinity, $p_1 \overset{u_1}{\to} h_1$ for some $u_1 \in \Sigma_1^*$ such that $\pi_\alpha(w_0u_0) = \pi_\alpha(w_1u_1)$ and $\pi_\beta(w_0u_0) = \pi_\beta(w_1u_1)$. This yields $\pi_\alpha(u_1) = x_\alpha\pi_\alpha(u_0)$ and $\pi_\beta(u_1) = \pi_\beta(u_0)$. Therefore u_1 contains at most $|u_0| \leq k_0 - 1$ symbols of the form $-b$, $b \in M_\beta$. At the same time, the length of $\pi_\alpha(u_1)$ is

$$|\pi_\alpha(u_1)| \geq |x_\alpha| \geq k_0(k_1 - 1) + 1 \ ,$$

and hence u_1 contains a (contiguous) subsequence v_1 of length $|v_1| \geq k_1$ that has no symbols $-b$, $b \in M_\beta$. Since F_1 has k_1 states, the path corresponding to v_1 contains a cycle. Hence F_1 has a receive cycle.

The second assertion in 6.2 now follows from the following lemma.

Lemma 6.6 *Let F_0 and F_1 be two affine SR-machines. If F_1 has a receive cycle then F_0 has a send cycle.*

Proof of 6.6. Again let k_j be the cardinality of K_j, for $j = 0, 1$. Since the transition diagram of F_1 is strongly connected and has a receive cycle, $h_1 \stackrel{w_1}{\to} h_1$ for some $w_1 \in \Sigma_1^*$ such that $|\pi_\beta(w_1)| \leq 2(k_1 - 1)$ and $|\pi_\alpha(w_1)| \geq (2k_1 - 1)(k_0 - 1) + 1$. By affinity, $h_0 \stackrel{w_0}{\to} h_0$ for some $w_0 \in \Sigma_0^*$ such that $\pi_\alpha(w_0) = \pi_\alpha(w_1)$ and $\pi_\beta(w_0) = \pi_\beta(w_1)$. It follows that w_0 contains a substring v_0 of length $|v_0| \geq k_0$ that has no symbol $+b$. Since F_0 has k_0 states, the path corresponding to v_0 contains a cycle; hence F_0 has a send cycle. □

The proof of 6.3 uses the following two lemmas.

Lemma 6.7 *If there is a reachable global state $((p_0, p_1), (x_\alpha, x_\beta))$ with $|x_\alpha| \geq k$ then there is a reachable global state $((p_0, q_1), (y_\alpha, \lambda))$ with $|y_\alpha| \geq k$.*

Lemma 6.8 *If the pair of affine machines is deadlock-free then for any w_0 such that $h_0 \stackrel{w_0}{\to} p_0$ there exists a path in the global state space, starting in (S^0, C^0), whose image in F_0 is labelled w_0.*

Proof of 6.7. There are $h_0 \stackrel{w_0}{\to} p_0$ and $h_1 \stackrel{w_1}{\to} p_1$ such that $\pi_\alpha(w_0) = \pi_\alpha(w_1)x_\alpha$ and $\pi_\beta(w_0)x_\beta = \pi_\beta(w_1)$. Find a prefix v_1 of w_1 such that $\pi_\beta(w_1) = \pi_\beta(v_1)x_\beta$. We have $h_1 \stackrel{v_1}{\to} q_1$ for some $q_1 \in K_1$. Since $\pi_\alpha(v_1)$ is a prefix of $\pi_\alpha(w_1)$, we can write $\pi_\alpha(w_1) = \pi_\alpha(v_1)y'_\alpha$ for some $y'_\alpha \in M_\alpha^*$. Set $y_\alpha = y'_\alpha x_\alpha$; then $\pi_\beta(w_0) = \pi_\beta(v_1)$ and $\pi_\alpha(w_0) = \pi_\alpha(v_1)y_\alpha$. Therefore $((p_0, q_1), (y_\alpha, \lambda))$ is reachable. □

Proof of 6.8. First observe that we can assume, without loss of generality, that $p_0 = h_0$ (because the path can be extended to h_0). Now, by affinity, there is w_1 such that $h_1 \stackrel{w_1}{\to} h_1$, $\pi_\alpha(w_1) = \pi_\alpha(w_0)$ and $\pi_\beta(w_1) = \pi_\beta(w_0)$. In the global state space, find the longest path that starts in (S^0, C^0) and whose image in F_j is labelled by a prefix v_j of w_j, for $j = 0, 1$; denote by $(S, C) = ((q_0, q_1), (x_\alpha, x_\beta))$ the end node of the path. We want to show that $v_0 = w_0$.

Assume $w_0 \neq v_0$, i.e. $w_0 = v_0 e u_0$ for some $e \in \Sigma_0$ and $u_0 \in \Sigma_0^*$. Distinguish several cases:
I. q_0 is a send state; then $e = -b$ for some $b \in M_\alpha$, and $q_0 \stackrel{-b}{\to} q'_0$ in F_0. Thus $(S, C) \stackrel{-b}{\vdash} ((q'_0, q_1), (x_\alpha b, x_\beta))$, which contradicts the maximality of the path.
II. q_0 is a receive state and $x_\beta \neq \lambda$. Then $e = +b$, $b \in M_\beta$, and b is the first symbol in x_β. Thus $q_0 \stackrel{+b}{\to} q'_0$ in F_0, and again the path in the global state space is not maximal.
III. q_1 is a receive state and $x_\alpha \neq \lambda$. This leads to a contradiction as in case II.
IV. Both q_0 and q_1 are receive states and $x_\alpha = \lambda = x_\beta$; this contradicts the assumption that the protocol is deadlock-free.

V. q_0 is a receive state, $x_\beta = \lambda$ and q_1 is a send state. Then $e = +b$, $b \in M_\beta$ and by affinity $q_1 \stackrel{-b}{\to} q_1'$. Again, the path is not maximal.

Thus in each case the assumption $w_0 \neq v_0$ leads to a contradiction. We conclude that $w_0 = v_0$. □

Proof of 6.3. By 6.2 and 6.7, if the protocol has no send cycles then it has the bounded channel property.

Conversely, assume that, for example, F_0 has a send cycle. Thus there are $p_0 \in K_0$ and $u_0 \in \Sigma_0^*$ such that $p_0 \stackrel{u_0}{\to} p_0$, $u_0 \neq \lambda$ and $\pi_\beta(u_0) = \lambda$. Denote $y_\alpha = \pi_\alpha(u_0)$. By Lemma 6.8, there are p_1, x_α and x_β such that the global state $((p_0, p_1), (x_\alpha, x_\beta))$ is reachable. It follows that, for every integer $i \geq 0$, the global state $((p_0, p_1), (x_\alpha y_\alpha^i, x_\beta))$ is reachable, and therefore the protocol has not the bounded channel property. □

Proof of 6.4. This algorithm solves the problem:
1. Check whether there are any send cycles.
2. If there are no send cycles, then (by 6.2) the protocol has the bounded channel property. Apply the exhaustive reachability analysis to decide deadlock-freedom.
3. If there is a send cycle then, by Theorem 6.3, the protocol is not deadlock-free or has not the bounded channel property. □

Proof of 6.5. Use the exhaustive reachability analysis. If any global state $(S, (x_\alpha, \lambda))$ with $|x_\alpha| \geq k_0(k_1-1)+1$ is reachable then, by 6.2, the protocol is not affine and deadlock-free (i.e. it is not affine or it is not deadlock-free).

If no such global state is reachable, then the protocol has the bounded channel property, and deadlock-freedom can be decided. Then affinity can be decided by Bird's algorithm; or alternatively it can be decided by a modified reachability analysis, since the two state machines differ by a "finite balance". □

7 Undecidable problems

We have seen in the previous section that the following problems are algorithmically decidable:

- Given any pair of SR-machines, is it affine?
- Given any pair of affine SR-machines, is it deadlock-free and has it the bounded channel property?
- Given any pair of SR-machines with no send cycles, is it affine and deadlock-free?

In this section we shall see that, in contrast to the previous results, some very similar problems are undecidable. Brand and Zafiropulo [Bra] prove the undecidability of several problems of this kind by reduction to the halting problem for Turing machines. The proofs in this section are somewhat similar to those in [Bra], but it will be more convenient for us to use Post's tag systems instead of Turing machines. Every tag system can be encoded as a pair of SR-machines; the known undecidability results about tag systems yield the following theorem.

Theorem 7.1 *For pairs of communicating SR-machines, these problems are undecidable:*
(a) Given any protocol with no send cycles, is it deadlock-free?
(b) Given any deadlock-free protocol with no send cycles, has it the bounded channel property?
(c) Given any affine protocol, is it deadlock-free?
(d) Given any affine protocol, has it the bounded channel property?

Theorem 7.1 and the results in the previous section pinpoint the frontier between the decidable and the undecidable for pairs of communicating SR-machines. Next we turn to more general protocols, and explore connections between the decidability properties and the topology of the underlying communication graph.

For a directed graph G, denote by ∇G the corresponding undirected graph. Consider first any CFSM protocol (with communication graph G) for which ∇G has no cycles. (Of course, such protocols are hardly of any use. They allow no feedback.) As in Theorem 5.2, one can show that every path in the global state space starting and ending in global states with empty channels is locally equal to a path that uses only global states of the form $(S, (x_\xi : \xi \in E))$, $\sum_\xi |x_\xi| \leq 1$. It follows that the stable composite state problem (and, in particular, the deadlock problem) is decidable for these protocols.

On the other hand, all "practical" communication graphs lead to undecidable problems. The claim is made precise, for the stable composite state problem, in the following theorem.

Theorem 7.2 *If G is a directed graph such that ∇G has a cycle then the stable composite state problem is undecidable for the CFSM protocols with the communication graph G.*

The forthcoming proofs of 7.1 and 7.2 are based on known results about Post's tag systems; the results are collected in the appendix.

The principal steps in the proof of 7.1 are stated and proved separately in 7.3, 7.4 and 7.5.

Lemma 7.3 *For every tag system* $\mathbf{T} = (\Sigma, g, w_0)$ *there is a protocol of two communicating SR-machines* F_0 *and* F_1 *with no send cycles such that*
(a) the protocol is deadlock-free if and only if $s_n(\mathbf{T}) \neq \lambda$ *for all* n*;*
(b) the protocol has the bounded channel property if and only if there is a constant c *such that* $|s_n(\mathbf{T})| \leq c$ *for all* n*.*

Proof of 7.3. For each $b \in \Sigma$ create two new symbols b_α and b_β; define $M_\alpha = \{\ b_\alpha \mid b \in \Sigma\ \} \cup \{f\}$ and $M_\beta = \{\ b_\beta \mid b \in \Sigma\ \}$, where f is a new symbol. The machine F_0 has a single receive state h_0, which is also its initial state, and one send state p_b for every $b \in \Sigma$, with transitions $h_0 \stackrel{+b_\beta}{\to} p_b$ and $p_b \stackrel{-b_\alpha}{\to} h_0$. Thus F_0 is a repeater (or a perfect transmission demon): it sends b_α whenever it receives b_β.

The machine F_1 simulates the tag system. It first transmits the string w_0 (subscripted by β), and then it alternately receives any d_α, receives any b_α, and transmits $g(d)$ subscripted by β. The transition diagram of F_1 is schematically depicted in Fig. 7.1, where $g(d) = g_{d0} g_{d1} \ldots g_{dm(d)}$ for every $d \in \Sigma$, and $w_0 = d_0 d_1 \ldots d_m$. There is a transition $q \stackrel{+d_\alpha}{\to} q_d$ for every $d \in \Sigma$. Note also the "dummy" transition $q \stackrel{+f}{\to} h_1$; it will never be used, but it makes the transition diagram strongly connected. Neither F_0 nor F_1 has a send cycle and if $|g|^- > 0$ then they have no receive cycles.

The pair (F_0, F_1) simulates the tag system $\mathbf{T}$ in the following sense: For $w \neq \lambda$ we have $w = s_n(\mathbf{T})$ for some n if and only if the global state $((h_0, q), (w_\alpha, \lambda))$ is reachable; and $\lambda = s_n(\mathbf{T})$ for some n if and only if either $((h_0, q), (\lambda, \lambda))$ or $((h_0, q_d), (\lambda, \lambda))$ for some $d \in \Sigma$ is reachable.

This proves (a) and, in view of Lemma 6.7, also (b). □

Lemma 7.4 *For every pair of communicating SR-machines* F_0' *and* F_1' *we can construct an affine pair* F_0*,* F_1 *such that either both pairs are deadlock-free or none is.*

Proof of 7.4. Let the channel alphabets be M_α' and M_β'. Let $\#_\alpha$ and $\#_\beta$ be two new symbols (not in $M_\alpha' \cup M_\beta'$) and define $M_\alpha = M_\alpha' \cup \{\#_\alpha\}$ and $M_\beta = M_\beta' \cup \{\#_\beta\}$. We construct F_0 and F_1, with the corresponding relations $\mathbf{Z}_0$ and $\mathbf{Z}_1$ (defined in section 6) both equal to

$$\{\ (u_\alpha \#_\alpha, u_\beta \#_\beta) \mid u_\alpha \in M_\alpha'^*\ ,\ u_\beta \in M_\beta'^*\ \}^*\ .$$

First we modify F_0' and F_1' so that no transitions lead to the initial states h_0' and h_1'. This is arranged as follows in F_0' (and similarly in F_1'): Add a new state p_0. Add the transition $p \stackrel{e}{\to} p_0$ whenever $p \stackrel{e}{\to} h_0'$ in F_0', and add $p_0 \stackrel{e}{\to} p$ whenever $h_0' \stackrel{e}{\to} p$ in F_0'. Then delete all transitions leading to h_0'. The resulting diagram is not strongly connected, but otherwise it satisfies all the properties of an SR-machine. The deadlock-freedom is not changed by the modification.

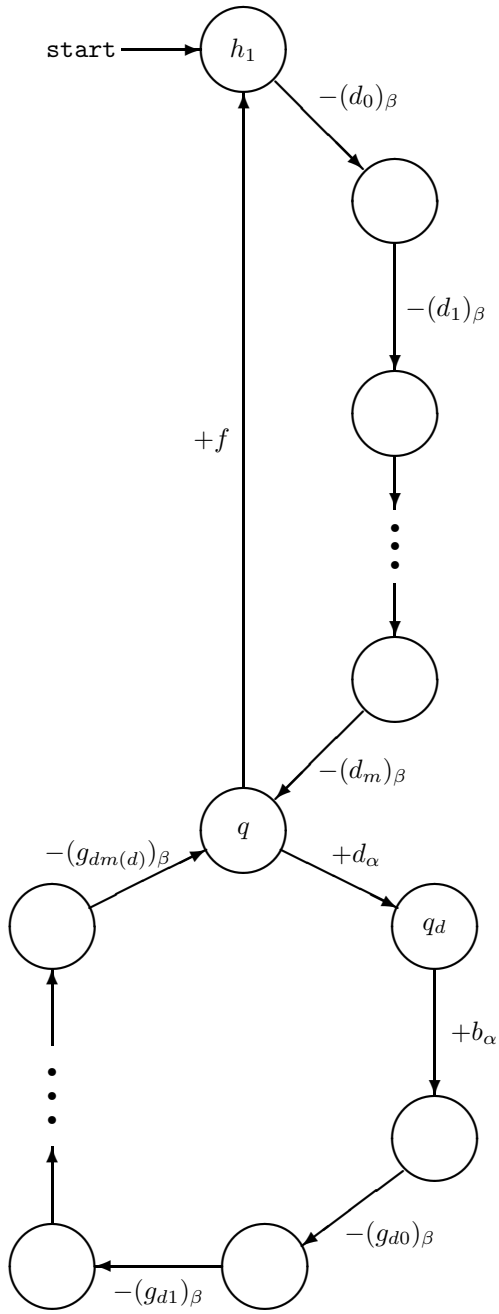


Fig. 7.1. The transition diagram of F_1.

The next step in the construction of F_0 is illustrated in Fig. 7.2 (it is again the same for F_1). Add two new send states s and s' and a new receive state r. For each send state p (including h'_0 if it is a send state) add the transition $p \stackrel{-\#_\alpha}{\to} r$, and add $p \stackrel{-b}{\to} s'$ whenever $p \stackrel{-b}{\to}$ not in F'_0. For each receive state p (including h'_0 if it is a receive state) add the transition $p \stackrel{+\#_\beta}{\to} s$, and add $p \stackrel{+b}{\to} s'$ whenever $p \stackrel{+b}{\to}$ not in F'_0. Also, add $s' \stackrel{-\#_\alpha}{\to} r$, $r \stackrel{+\#_\beta}{\to} h'_0$, $s \stackrel{-\#_\alpha}{\to} h'_0$; $s' \stackrel{-b}{\to} s'$ and $s \stackrel{-b}{\to} s$ for every $b \in M'_\alpha$; and $r \stackrel{+b}{\to} r$ for every $b \in M'_\beta$. Call the resulting SR-machine F_0, and call F_1 that constructed in the same way from F'_1.

If $h'_0 \stackrel{w}{\to} h'_0$ in F_0, $w \neq \lambda$, and if $h'_0 \stackrel{u}{\to} h'_0$ for no proper nonempty prefix u of w, then $\pi_\alpha(w) = u_\alpha \#_\alpha$ for some $u_\alpha \in M'^*_\alpha$ and $\pi_\beta(w) = u_\beta \#_\beta$ for some $u_\beta \in M'^*_\beta$. Conversely, for any $u_\alpha \in M'^*_\alpha$ and $u_\beta \in M'^*_\beta$ there exists w such that $h'_0 \stackrel{w}{\to} h'_0$, $\pi_\alpha(w) = u_\alpha \#_\alpha$ and $\pi_\beta(w) = u_\beta \#_\beta$. Therefore

$$\mathbf{Z}_0 \;=\; \{\, (u_\alpha \#_\alpha, u_\beta \#_\beta) \mid u_\alpha \in M'^*_\alpha \;,\; u_\beta \in M'^*_\beta \,\}^* \;.$$

For the same reason, $\mathbf{Z}_1$ is equal to the same relation. Hence F_0 and F_1 are affine.

Every reachable deadlocked global state for the pair (F'_0, F'_1) is reachable for (F_0, F_1). At the same time, no additional deadlocked global states are reachable for (F_0, F_1); if, for example, F_0 is in its state r and the channels are empty then F_1 must be in its state s, which is not a receive state. □

Lemma 7.5 *For every pair of communicating SR-machines F'_0 and F'_1 we can construct an affine pair F_0, F_1 such that either both pairs have the bounded channel property or none has.*

(Note that, in view of 6.4, the constructions in 7.4 and 7.5 cannot be combined. More precisely, it is not true that for every F'_0 and F'_1 we can construct an affine pair F_0, F_1 such that both the deadlock-freedom and the bounded-channel property are shared by the two pairs.)

Proof of 7.5. As in the proof of 7.4, we define $M_\alpha = M'_\alpha \cup \{\#_\alpha\}$ and $M_\beta = M'_\beta \cup \{\#_\beta\}$. We construct F_0 and F_1 such that the corresponding relations $\mathbf{Z}_0$ and $\mathbf{Z}_1$ are both equal to

$$\{\, (u_\alpha \#_\alpha \#_\alpha, u_\beta \#_\beta \#_\beta) \mid u_\alpha \in M'^*_\alpha \;,\; u_\beta \in M'^*_\beta \,\}^* \;.$$

Again we first arrange that no transitions lead to the initial states h'_0 and h'_1. The next step is shown, for F'_0, in Fig. 7.3. Add four new receive states r, r', r'' and r''' and two new send states s and s'. For each send state p (including h'_0 if it is a send state p) add the transition $p \stackrel{-\#_\alpha}{\to} r''$, and add $p \stackrel{-b}{\to} r$ whenever $p \stackrel{-b}{\to}$ not in F'_0. For each receive state p (including h'_0 if it is a receive state p) add the transition $p \stackrel{+\#_\beta}{\to} r'$, and add $p \stackrel{+b}{\to} r$ whenever $p \stackrel{+b}{\to}$ not in F'_0. Also, add $r \stackrel{+\#_\beta}{\to} r'$, $r' \stackrel{+\#_\beta}{\to} s$, $s \stackrel{-\#_\alpha}{\to} s'$, $r'' \stackrel{+\#_\beta}{\to} r'''$, $r''' \stackrel{+\#_\beta}{\to} s'$, $s' \stackrel{-\#_\alpha}{\to} h'_0$;

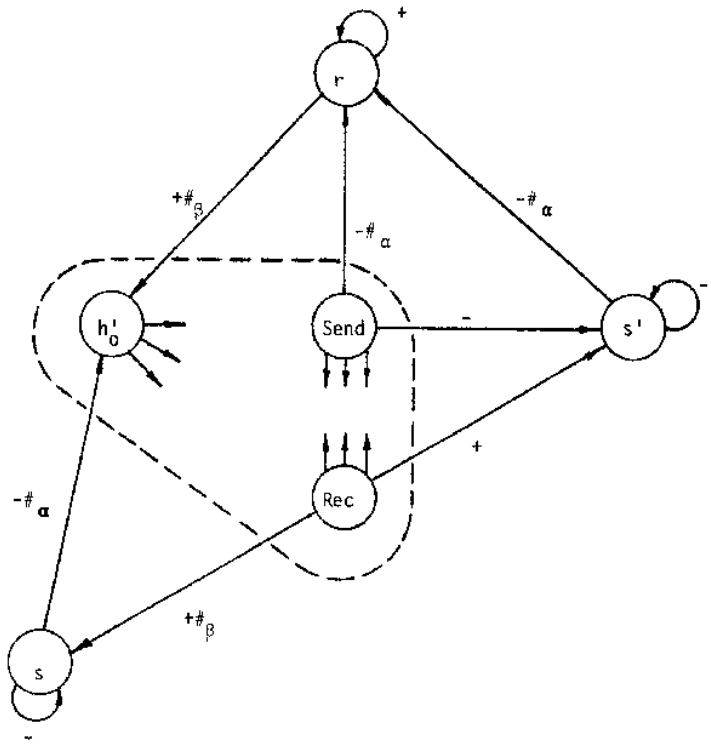


Fig. 7.2. The construction of F_0 in the proof of 7.4.

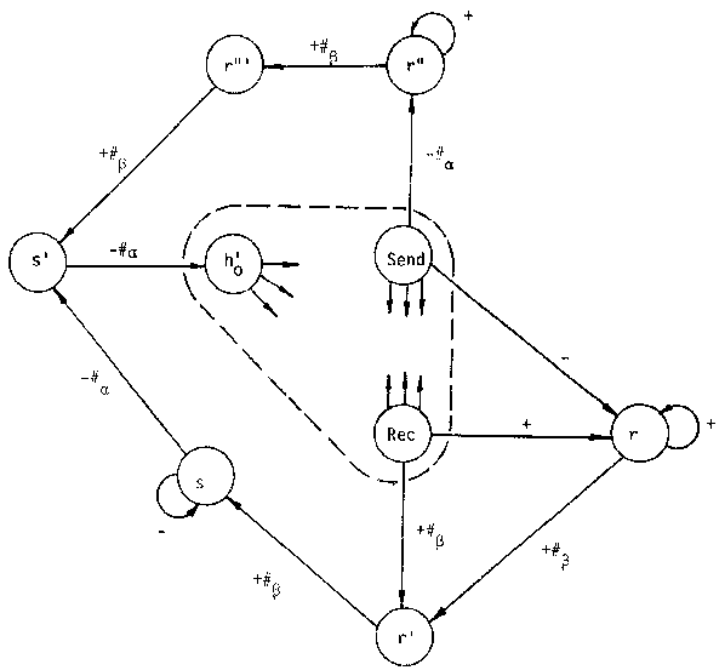


Fig. 7.3. The construction of F_0 in the proof of 7.5.

$r \stackrel{+b}{\to} r$ and $r'' \stackrel{+b}{\to} r''$ for every $b \in M'_\beta$; and $s \stackrel{-b}{\to} s$ for every $b \in M'_\alpha$. Call F_0 the resulting SR-machine, and call F_1 that constructed the same way from F'_1. As in the proof of 7.4 it now follows that

$$\mathbf{Z}_0 = \mathbf{Z}_1 = \{ (u_\alpha \#_\alpha \#_\alpha, u_\beta \#_\beta \#_\beta) \mid u_\alpha \in M'^*_\alpha \ , \ u_\beta \in M'^*_\beta \}^* \ .$$

The construction creates new reachable deadlocked global states. In fact, the protocol will never get over the states r' and r'''; hence no global state containing s or s' is reachable. It follows that the loop at s will never be entered and, therefore, the pair (F_0, F_1) has the bounded channel property if and only if (F'_0, F'_1) has. □

Proof of 7.1. (a) follows directly from Theorem A.1 (in the appendix) and 7.3(a).

Similarly, (b) follows from A.3, 7.3(a) and 7.3(b). (Observe that the construction 7.3 is such that if the tag system $\mathbf{T} = (\Sigma, g, w_0)$ satisfies $|g|^- > 0$ then the protocol has no receive cycles. Hence the problems (a) and (b) are undecidable even for the protocols with no send and no receive cycles.)

To prove the undecidability of (c), we combine the already proved case (a) with 7.4. Similarly, (d) follows from (b) and 7.5. □

The forthcoming Lemma 7.6 will simplify the proof of 7.2. Say that two finite directed graphs are *homeomorphic* if one can be transformed to the other by a finite sequence of elementary replacements, each of which either replaces an edge $0 \to 1$ by two edges $0 \to 2 \to 1$ (where 2 is a new vertex) or vice versa. For example, the two graphs in Fig. 7.4 are homeomorphic.

Lemma 7.6 *Let G and G' be two homeomorphic graphs. The problem "Is a given composite state stable?" is decidable for every CFSM protocol with the communication graph G if and only if it is decidable for every CFSM protocol with the communication graph G'.*

It will be obvious from the proof of 7.6 that the same result holds for the deadlock problem, the bounded-channel problem, etc.

Proof of 7.6. It is enough to prove the result under the assumption that G' is produced from G by a single elementary replacement, which replaces $0 \to 1$ by $0 \to 2 \to 1$. Assume this is the case.

Let the problem be decidable for every CFSM protocol with the communication graph G, and let $\mathbf{P}'$ be a protocol with the communication graph G'. Using the abstract flow control argument of sections 5 and 10 (with the highest priority at the node 2), we can confine ourselves to the global states in which the channel from 0 to 2 contains at most one symbol, and we do not lose any reachable global states of the form (S, C^0). Now we combine the state of the machine at 0, the state of the machine at 2, and the content of the channel

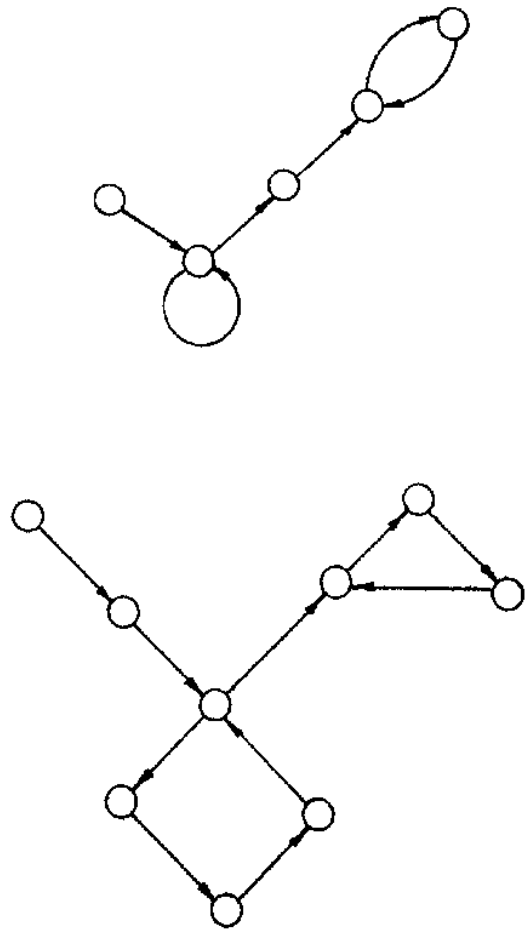


Fig. 7.4. Two homeomorphic graphs.

$0 \to 2$ into a single state; this transforms $\mathbf{P}'$ into a protocol with the communication graph G. It follows that the problem is decidable for G'.

Conversely, assume that the problem is decidable for G'. Every CFSM protocol with the communication graph G can be transformed into one with the communication graph G' by including a repeater (perfect transmission demon) at the node 2. It follows that the problem is decidable for G. $\square$

Proof of 7.2. Clearly it suffices to prove the undecidability for every graph G for which ∇G is a circle. When ∇G is a circle, there are two possibilities: Either G itself is a (directed) cycle or G is acyclic as a directed graph. Since every directed cycle is homeomorphic to the graph $0 \rightleftarrows 1$, the case of G being a cycle is taken care of by 7.1(a) (or 7.1(c)) and 7.6.

It remains to be proved that the stable composite state problem is undecidable for every acyclic graph G for which ∇G is a circle. The proof is based on the undecidability of *modified Post's correspondence problem* (MPCP). Recall [Hop] that an *instance* of MPCP consists of two lists $x = (x_0, x_1, \ldots, x_n)$ and $y = (y_0, y_1, \ldots, y_n)$ of strings over an alphabet Σ. The instance *has a solution* if there is a sequence of integers $j_1, j_2, \ldots, j_k$ such that

$$x_0 x_{j_1} \ldots x_{j_k} = y_0 y_{j_1} \ldots y_{j_k} \ ;$$

The sequence $j_1, j_2, \ldots, j_k$ is called a *solution* for the instance of MPCP. It is known that the problem "Given an instance of MPCP, has it a solution?" is undecidable ([Hop], 8.5).

Every acyclic graph G for which ∇G is a circle is homeomorphic to the graph in Fig. 7.5, for some $m \geq 0$. Hence the undecidability result follows from 7.6 and from this lemma:

Lemma 7.7 *For the graph G in Fig. 7.5 and for every instance of MPCP there exist a CFSM protocol with the communication graph G and a composite state S such that S is stable if and only the instance of MPCP has a solution.*

Proof of 7.7. Let Σ be the alphabet of the instance of MPCP. For every edge ξ in G, the channel alphabet M_ξ is defined to be $\{ b_\xi \mid b \in \Sigma \}$, where the symbols b_ξ are chosen so that the sets M_ξ are pairwise disjoint.

All the finite state machines except the one at 0 are simple comparators: Those at the even numbered nodes (except 0) send the same sequences of messages to both channels, those at the odd numbered nodes receive the same sequences from both channels. For example, the machine at 1 has the initial state h_1 and a separate state p_b for each $b \in \Sigma$, with transitions $h_1 \stackrel{+b_{\alpha_0}}{\to} p_b$ and $p_b \stackrel{+b_{\alpha_1}}{\to} h_1$.

The machine at 0 is capable of sending, for every infinite sequence of indices $j_1, j_2, \ldots$, the sequence of messages

$$(x_0)_{\alpha_{2m+1}} (x_{j_1})_{\alpha_{2m+1}} \ldots$$

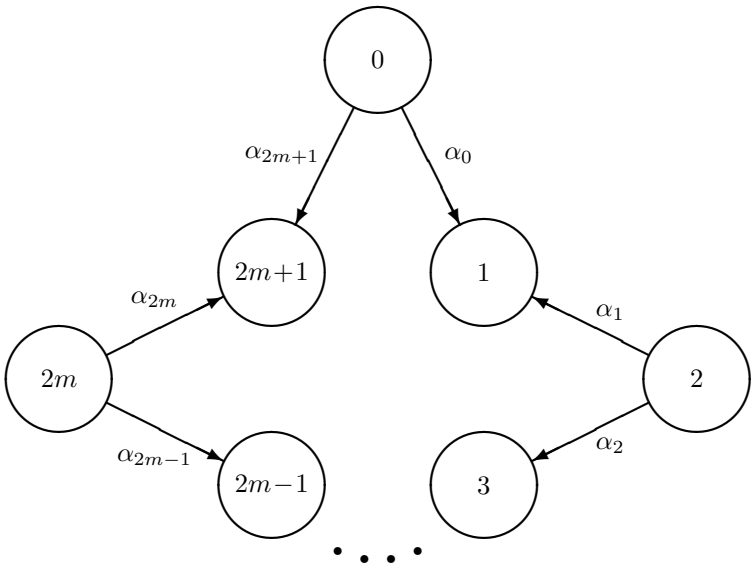


Fig. 7.5.

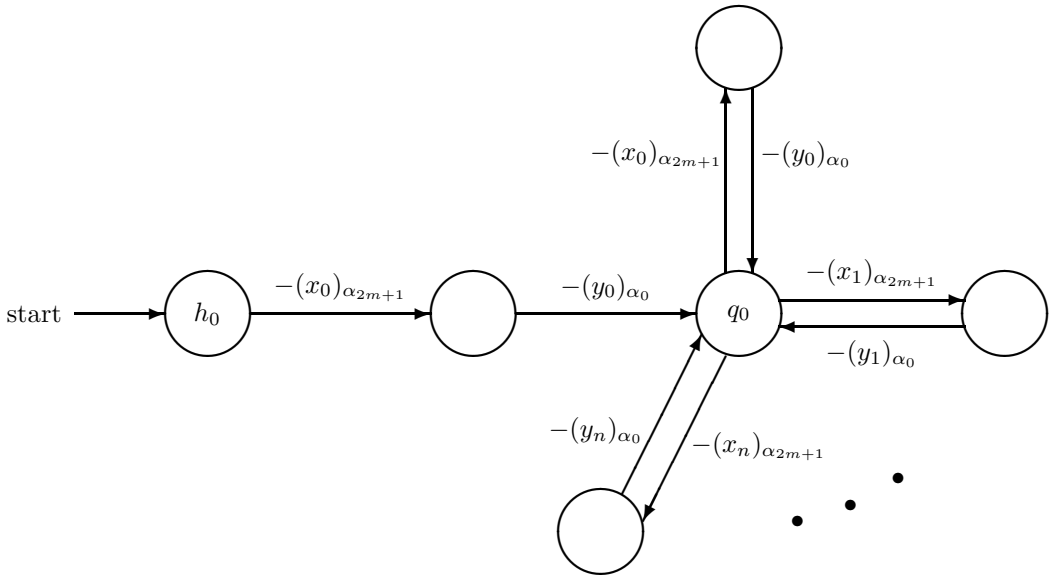


Fig. 7.6. A machine to simulate MPCP.

on the channel α_{2m+1}, and the sequence

$$(y_0)_{\alpha_0}(y_{j_1})_{\alpha_0}\cdots$$

on the channel α_0. A schematic transition diagram is in Fig. 7.6. The composite state $S = (q_0, h_1, h_2, \ldots, h_{2m+1})$ is stable if and only if there is solution of the MPCP.

This completes the proofs of 7.7 and 7.2. □

8 Rational channels for cyclic protocols

The results in the previous section show that general CFSM protocols can, with the help of their infinite channels, simulate arbitrary computation processes. It is for this reason that the reachability problems are undecidable. However, we are primarily interested in the protocols that use their channels more simply. Can we disqualify the CFSM protocols that, by using the channels as an infinite memory, simulate general computations? Can the "simple channel property" (or, more precisely, the property of "the channels being used in a simple manner") be formalized? One sufficient condition for this kind of channel simplicity is the bounded channel property. Two more general conditions are offered in this section.

The popular classification of verification techniques for communication protocols distinguishes between reachability analysis and program proofs [Bo1]. Traditionally, program proofs have been used to verify the protocol properties that are not amenable to reachability analysis. Our present aim is different: The primitive assertion proving technique proposed below is more powerful than the exhaustive reachability analysis, but it stays within the realm of reachability properties.

Rather than treating the reachability analysis and program proofs as two opposites, we shall regard the former as a simple special case of the latter. (Bochmann alludes to this perspective in [Bo2], p. 649.) In this view, illustrated by the following example, the reachability analysis of a bounded-channel CFSM protocol is a method for constructing and proving a set of simple assertions attached to composite states.

Example 8.1 The purpose of the protocol is to limit the total number of messages simultaneously in transit (ie. the total number of buffers needed). In the example, the limit is two. (Any other limit can be used. The larger the limit, the more states the finite state machines have.) The protocol assumes error-free channels. Data messages are transmitted in both directions. There are three message types:

`DATA`	data message,
`ACK`	acknowledgement of `DATA`,
`RELE`	releasing buffer.

Initially, each channel is allocated one buffer. Either transmitter can release a buffer, which is then used for transmissions in the opposite direction. The two finite state machines are identical. Fig. 8.1 shows their transmission diagrams and the communication graph.

Fig. 8.2 is the complete global state space of the protocol. (We write D=`DATA`, R=`RELE` and A=`ACK`.) The global state space is finite; from it one can read various reachability properties: The total number of messages in transit is at most two, the protocol is deadlock-free, etc. Fig. 8.3 shows a different data structure, which contains the same information as Fig. 8.2 (when Fig. 8.1 is known). The table in Fig. 8.3 lists, for each composite state, the set of all possible channel contents. We can regard each entry in the table as an *assertion*. For example, the entry $\{(\texttt{DATA}, \lambda), (\texttt{RELE}, \lambda), (\texttt{ACK}, \lambda), (\lambda, \texttt{DATA}), (\lambda, \texttt{RELE}), (\lambda, \texttt{ACK})\}$ at $(03, 10)$ asserts: If the state of Process 0 is 03 and the state of Process 1 is 10, then the channel content is $(\texttt{DATA}, \lambda)$ or $(\texttt{RELE}, \lambda)$ or $(\texttt{ACK}, \lambda)$ or $(\lambda, \texttt{DATA})$ or $(\lambda, \texttt{RELE})$ or $(\lambda, \texttt{ACK})$.

The assertions in Fig. 8.3 can be written more compactly. E.g. the entry at $(03, 13)$ is the relation $\{(x, y) \mid |x| + |y| = 2\}$, the entry at $(00, 13)$ is the relation $\{(x, y) \mid |x| + |y| = 1\}$, etc. Quite simply, the protocol implements a distributed counter. However, an automatic assertion verifier would have to be considerably more intelligent to understand such descriptions.

From the table in Fig. 8.3 we can read, for example, that the composite state $(01, 13)$ is stable, and that no message can arrive at 02. (End of Example 8.1.)

In this view, the exhaustive reachability analysis is a method for constructing and verifying the correctness of tables whose entries are finite sets of channel contents. One can argue that the table, or a portion of it, should be a part of the protocol description, because it offers an additional insight into the structure of the protocol. This is especially true if the protocol has not the bounded channel property. In that case the entries in the table are infinite sets, and the complete table cannot be constructed by the exhaustive reachability analysis. If the table is supplied together with the CFSM description then the analysis algorithm need not construct the table, it merely has to verify its correctness (consistency).

The distinctive feature of the exhaustive reachability analysis is that the domain of assertions (the language that they are formulated in) is extremely simple, and therefore analysis can be efficiently automated. On the other hand, the method has several limitations. Here we address its inability to analyze protocols with unbounded channels.

Generally speaking, the way to overcome the limitations of any assertion proving system is to extend the domain of assertions; in doing so we trade simplicity for power. A natural extension of the exhaustive reachability analysis is to use more general relations, instead of finite ones, in the assertions. Two important families of relations have been extensively studied in the last ten years, the *recognizable* and the *rational* relations; their basic properties can be found in [Ber] and [Eil]. Every finite relation is recognizable and every recognizable relation is rational.

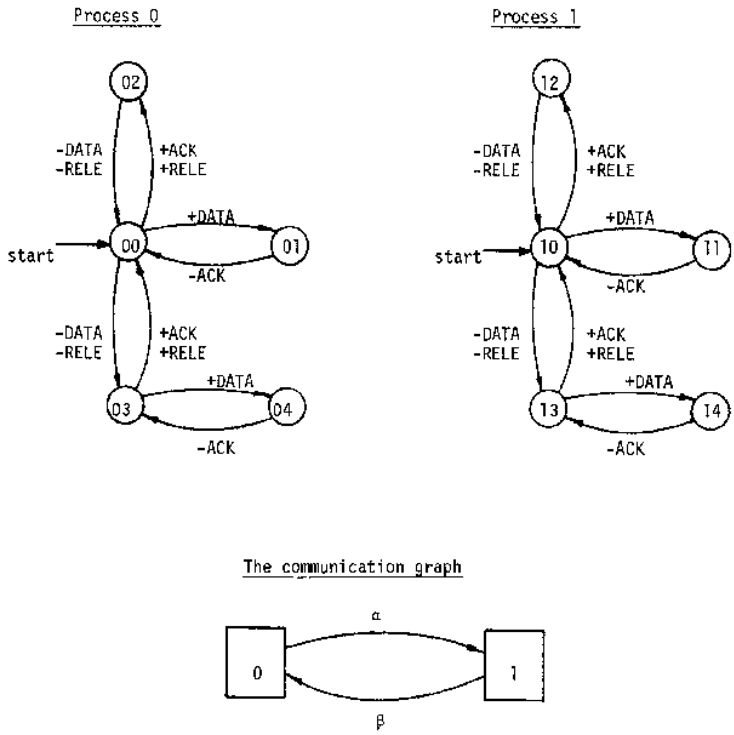


Fig. 8.1. A simple flow control protocol.

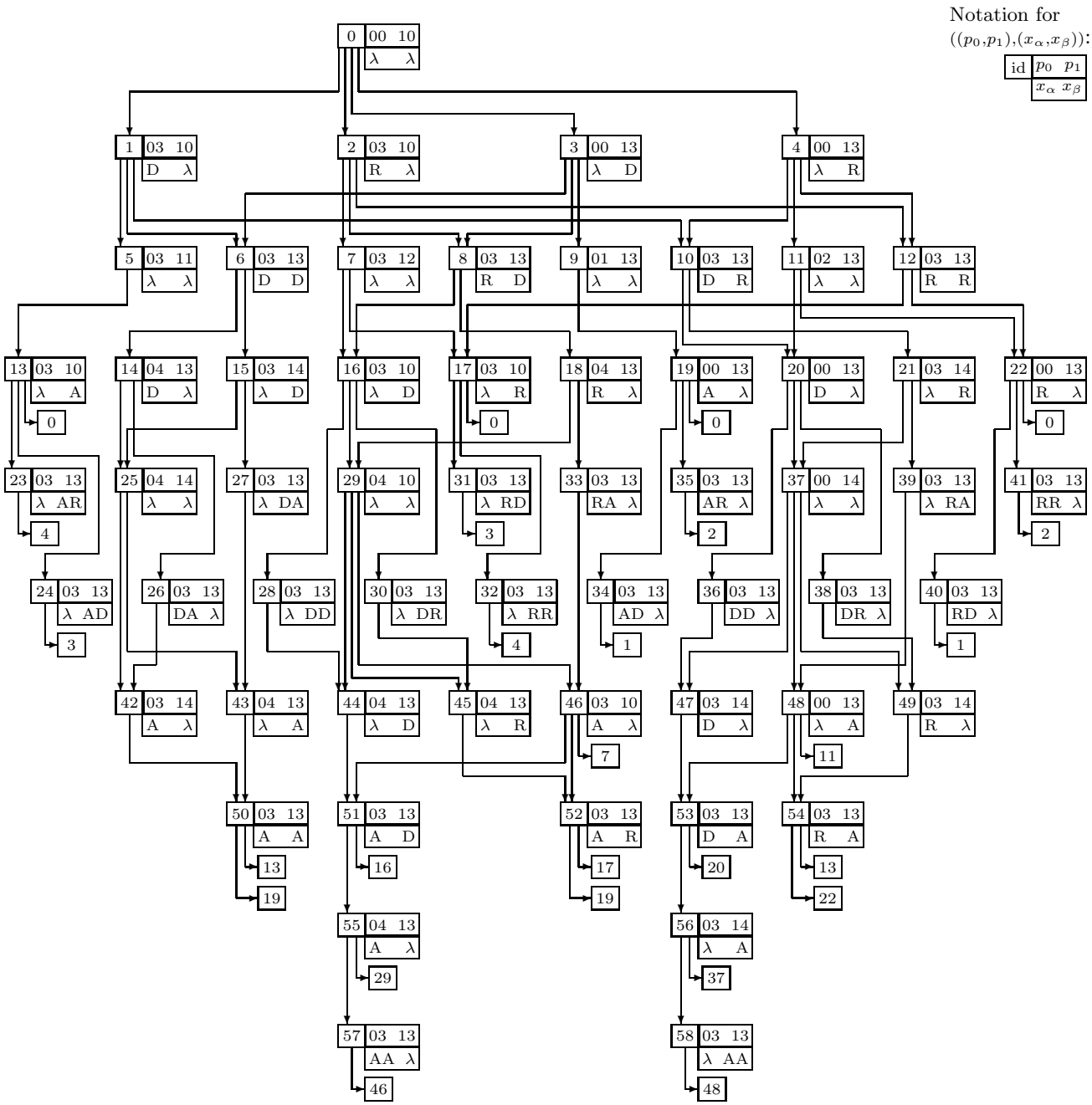


Fig. 8.2. The global state space.

	10	11	12	13	14
00	(λ,λ)	$\emptyset$	$\emptyset$	(λ,D), (λ,R), (λ,A), (λ,D), (λ,R), (λ,A)	(λ,λ)
01	$\emptyset$	$\emptyset$	$\emptyset$	(λ,λ)	$\emptyset$
02	$\emptyset$	$\emptyset$	$\emptyset$	(λ,λ)	$\emptyset$
03	(λ,D), (λ,R), (λ,A), (D,λ), (R,λ), (A,λ)	(λ,λ)	(λ,λ)	(λ,DD), (λ,DR), (λ, DA), (λ,RD), (λ,RR), (λ,RA), (λ,AD), (λ,AR), (λ,AA), (D,D), (D,R), (D,A), (R,D), (R,R), (R,A), (A,D), (A,R), (A,A), (DD,λ), (DR,λ), (DA,λ), (RD,λ), (RR,λ), (RA,λ), (AD,λ), (AR,λ), (AA,λ)	(λ,D), (λ,R), (λ,A), (D,λ), (R,λ), (A,λ)
04	(λ,λ)	$\emptyset$	$\emptyset$	(λ,D), (λ,R), (λ,A), (D,λ), (R,λ), (A,λ)	(λ,λ)

Fig. 8.3. Another description of the global state space.

We are going to extend the assertion domain by using recognizable and rational relations in place of finite ones. We gain power (ability to analyze protocols with unbounded channels), while not losing all the simplicity: The assertion verifier will have to be smarter but still fairly simple.

Definition 8.2 Let **P** be a CFSM protocol. Say that **P** has the *rational channel property* if the relation

$$\mathbf{L}(S) = \{\ C \mid (S^0, C^0) \vdash^* (S, C)\ \} \ \subseteq \bigtimes_{\xi \in E} M_\xi^*$$

is rational for each composite state $S \in \bigtimes_{j \in N} K_j$. Say that **P** has the *recognizable channel property* if $\mathbf{L}(S)$ is recognizable for each S.

Thus the bounded channel property implies the recognizable channel property, which in turn implies the rational channel property.

In this section we concentrate on cyclic protocols. We return to general CFSM protocols

in the next section. As we have seen in Theorem 5.2, cyclic protocols have the property that unbounded channel growth can be confined to a single channel.

Theorem 8.3 *For any cyclic CFSM protocol* $\mathbf{P}$ *the following four conditions are equivalent:*
(a) $\mathbf{P}$ *has the recognizable channel property;*
(b) $\mathbf{P}$ *has the rational channel property;*
(c) for every $\beta \in E$ *and for every composite state* S*, the set*

$$Q_\beta(S) \;=\; \{\, x_\beta \in M_\beta^* \mid (x_\xi : \xi \in E) \in \mathbf{L}(S) \;\text{ and }\; x_\xi = \lambda \;\text{ for }\; \xi \neq \beta \,\}$$

is regular;
(d) there exists $\beta \in E$ *such that the set* $Q_\beta(S)$ *is regular for every* S*.*

Thus the recognizable and the rational channel property coincide for cyclic protocols. We shall see later that this is not the case in general.

The sets $Q_\beta(S)$ of Theorem 8.3 are consistent, in this sense: If $(S, (x_\xi : \xi \in E)) \vdash^* (S', (x'_\xi : \xi \in E))$, $x_\xi = x'_\xi = \lambda$ for $\xi \neq \beta$, and $x_\beta \in Q_\beta(S)$ then $x'_\beta \in Q_\beta(S')$. At the same time, there is an efficient algorithm to decide whether a given family of *regular* sets $Q(S)$, indexed by $S \in \bigtimes_{j \in N} K_j$, is consistent (with respect to β and $\mathbf{P}$).

Moreover, a consistent family $Q(S)$ such that $\lambda \in Q(S^0)$ and $\lambda \notin Q(S)$ constitutes a *proof* that (S, C^0) is not reachable from (S^0, C^0)(i.e. that S is not a stable state). Consequently, if a cyclic protocol has the rational channel property then for each non-stable S there is an *automatically verifiable* proof that S is not stable.

The foregoing discussion is summed up in Definition 8.4 and Theorems 8.5 and 8.6.

Definition 8.4 Let $\mathbf{P}$ be a CFSM protocol, $\beta \in E$, and let $Q(S) \subseteq M_\beta^*$ for every $S \in \bigtimes_{j \in N} K_j$. Say that the sets $Q(S)$ are *consistent (with respect to* $\mathbf{P}$ *and* β*)* if

$$(S, (x_\xi : \xi \in E)) \vdash^* (S', (x'_\xi : \xi \in E)), \;\; x_\xi = x'_\xi = \lambda \;\text{ for }\; \xi \neq \beta, \;\text{ and }\; x_\beta \in Q(S)$$

imply $x'_\beta \in Q(S')$.

Theorem 8.5 *There is an algorithm to decide whether any given family of regular sets* $Q(S)$ *is consistent (with respect to a given cyclic* $\mathbf{P}$ *and a given* β*).*

Theorem 8.6 *Let* $\mathbf{P}$ *be a cyclic CFSM protocol with the rational channel property, and let* $\beta \in E$*. A composite state* S' *is not stable if and only if there is a consistent family of regular sets* $Q(S)$*,* $S \in \bigtimes_{j \in N} K_j$*, such that* $\lambda \in Q(S^0)$ *and* $\lambda \notin Q(S')$*.*

The following corollary to 8.5 and 8.6 shows that the rational channel property indeed prevents, in an essential way, the cyclic protocol from using channels as a general infinite memory.

Corollary 8.7 *The deadlock problem is algorithmically decidable for cyclic CFSM protocols with the rational channel property.*

The algorithm in the proof of 8.7 (at the end of this section) is awfully inefficient; it exhaustively searches for the proof of deadlock-freedom. However, once the proof is known, it can be efficiently verified. Therefore it makes sense to require that the protocol designer supply the proof (in the form of channel expressions) as a part of the protocol description. The description of a protocol by means of CFSM augmented with channel expressions will be exhibited in Example 8.9. The description is substantially abridged with the help of the simple result in the forthcoming Theorem 8.8. It says that one need not supply the sets $Q(S)$ for all S; it is sufficient to describe $Q(S)$ for sufficiently many S, and all the other sets $Q(S)$ can be automatically computed.

Theorem 8.8 *Let* $\mathbf{P}$ *be a cyclic CFSM protocol and* $\beta \in E$*. For each* $j \in N$*, let* $V_j \subseteq K_j$ *be a set of states such that* $h_j \in V_j$ *and if* $p_j \xrightarrow{-b} q_j$*,* $p_j, q_j \in K_j$*, then* $q_j \in V_j$*. Then there is an algorithm to decide whether any given family of regular sets indexed by* $S \in \bigtimes_{j \in N} V_j$ *can be extended to a consistent family of sets* $Q(S)$ *indexed by* $S \in \bigtimes_{j \in N} K_j$*. Moreover, if the family can be extended than the smallest such sets* $Q(S)$ *are regular and can be automatically constructed.*

The proofs of the results in this section come after the following example, which illustrates the proposed proof method.

Example 8.9 This is a variation of the alternating bit protocol described in section 2. In the present version both stations take turns in transmitting and receiving data packets. The communication graph is again as in Fig. 8.4.

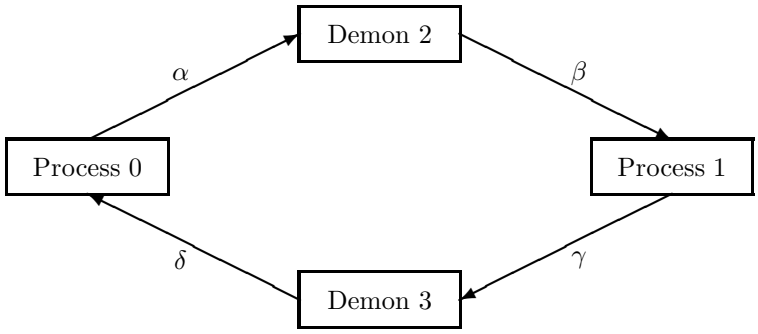


Fig. 8.4. The communication graph.

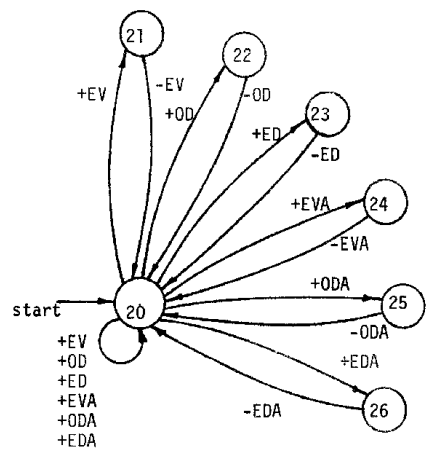


Fig. 8.5. Demon 2.

Demon 2 and Demon 3 are identical. Demon 2 is defined in Fig. 8.5; Demon 3 differs only in state numbers (30, 31, ... instead of 20, 21, ...). Processes 0 and 1 are defined in Fig. 8.6. They differ only in the starting state.

Theorem 8.8 applies for these sets V_j:
$V_0 = \{00, 01, 02, 04\}$,
$V_1 = \{10, 11, 12, 14\}$,
$V_2 = \{20\}$,
$V_3 = \{30\}$.
This reduces the number of the sets $Q(S)$ that have to be specified from $6 \times 6 \times 7 \times 7 = 1764$ to $4 \times 4 \times 1 \times 1 = 16$. The sets $Q_\alpha(S)$ for $S \in \bigtimes_{j=0}^{3} V_j$ are listed in Fig. 8.7. (Recall that, in agreement with the notation in Theorem 8.3, $Q_\alpha(S)$ is the set of all possible contents of the channel from Process 0 to Demon 2 when the other channels are empty.) Each K_j, $j = 0, 1, 2, 3$, contains one receive state: the protocol is deadlock-free if and only if the global state $((04, 14, 20, 30), C^0)$ is unreachable. Since the $(04, 14, 20, 30)$ entry in Fig. 8.7 is the empty set, the protocol is deadlock-free.
(End of Example 8.9.)

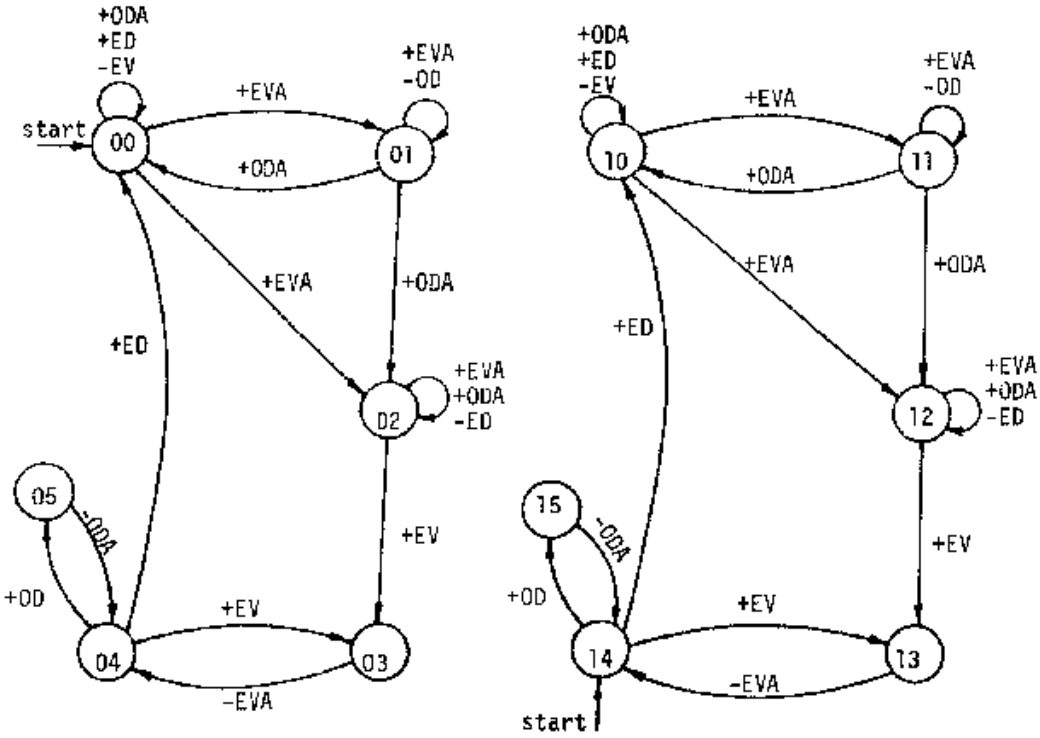


Fig. 8.6. Another alternating bit protocol.

p	q			
	10	11	12	14
00	$\emptyset$	$\emptyset$	$\mathtt{EVA}^*\ \mathtt{EV}^* \cup \mathtt{ODA}^*\ \mathtt{EV}^*$	$\mathtt{OD}^*\ \mathtt{EV}^*$
01	$\emptyset$	$\emptyset$	$\emptyset$	$\mathtt{EV}^*\ \mathtt{OD}^*$
02	$\mathtt{ED}^*$	$\emptyset$	$\emptyset$	$\mathtt{EV}^*\ \mathtt{ED}^* \cup \mathtt{OD}^*\ \mathtt{ED}^*$
04	$\mathtt{ED}^*\ \mathtt{EVA}^* \cup \mathtt{ODA}^*\ \mathtt{EVA}^*$	$\mathtt{EVA}^*\ \mathtt{ODA}^*$	$\mathtt{EVA}^* \cup \mathtt{ODA}^*$	$\emptyset$

Fig. 8.7. $Q_\alpha((p,q,20,30))$ **for** $p \in \{00,01,02,04\}$ **and** $q \in \{10,11,12,14\}$**.**

The proofs of the results in this section follow. Several proofs use the "priority argument" informally; it could be formalized as in the proof of 10.1.

First we establish two lemmas that will be needed in the proof of 8.3.

Lemma 8.10 *Let M_1 and M_2 be two alphabets. If $R \subseteq M_1^*$ is a regular set and $\mathbf{L} \subseteq M_1^* \times M_2^*$ is a rational relation then the relation*

$$\mathbf{L} \setminus R \;=\; \{\ (x,y) \in M_1^* \times M_2^* \ \mid\ \exists z : zx \in R \quad \text{and} \quad (z,y) \in \mathbf{L}\ \}$$

is recognizable.

Proof. Let $F = (K, M_1, T, h, A)$ be a deterministic finite automaton accepting R; we use the notation of [Hop]. For each $p \in K$, denote R_{hp} the language accepted by $(K, M_1, T, h, \{p\})$, and R_{pA} the language accepted by (K, M_1, T, p, A). Define

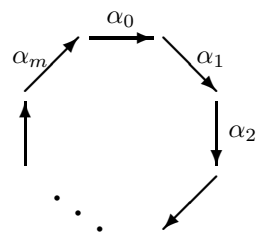
$$\mathbf{L}(R_{hp}) \;=\; \{\ y \in M_2^* \mid \exists x \in R_{hp} : (x,y) \in \mathbf{L}\ \}\ .$$

Now

$$\mathbf{L} \setminus R \;=\; \bigcup_{p \in K} R_{pA} \times \mathbf{L}(R_{hp})$$

and each $\mathbf{L}(R_{hp})$ is regular. It follows that $\mathbf{L} \setminus R$ is recognizable. □

Lemma 8.11 *Let $\mathbf{P}$ be a cyclic CFSM protocol with the communication graph $G = (N, E)$ where $E = \{\alpha_0, \alpha_1, \ldots, \alpha_m\}$, $-\alpha_0 = +\alpha_m$, $-\alpha_1 = +\alpha_0$, …, $-\alpha_m = +\alpha_{m-1}$:*



If (S', C') is a reachable global state such that $C' = (x_\xi : \xi \in E)$, $x_{\alpha_i} = \lambda$ for $k+1 < i \le m$, then there are a reachable global state (S, C) and a path Γ from (S, C) to (S', C') such that
(a) $C = (y_\xi : \xi \in E)$, $y_{\alpha_i} = \lambda$ for $k < i \le m$;
(b) $\mathrm{Im}_i(\Gamma)$ is a trivial path (of length 0) for each $i \ne +\alpha_k$.

Proof. We use the same priority argument as in the proof of 5.2. There is a path from (S^0, C^0) to (S', C'); rearrange it by giving the lowest priority to the node $+\alpha_k = -\alpha_{k+1}$. Let Γ be the longest suffix of the rearranged path for which (b) holds. Let (S, C) be the starting global state of Γ. Then $C = (y_\xi : \xi \in E)$ must satisfy (a): If $y_{\alpha_{k+1}} \ne \lambda$ then Γ could be made one step longer; if $y_{\alpha_i} \ne \lambda$ for some $i > k+1$ then Γ could not lead to (S', C'). $\square$

Proof of 8.3. Clearly (a)⇒(b) and (c)⇒(d). To prove the implication (b)⇒(c), observe that

$$Q_\beta(S) \times \bigtimes_{\substack{\xi \in E \\ \xi \ne \beta}} \{\lambda\} \;=\; \mathbf{L}(S) \cap \{\, (x_\xi : \xi \in E) \mid x_\xi = \lambda \;\text{ for }\; \xi \ne \beta \,\} \,.$$

The relation $\{(x_\xi : \xi \in E) \mid x_\xi = \lambda \;\text{ for }\; \xi \ne \beta\}$ is recognizable, hence the right hand side is rational ([Ber], p. 57). Since $Q_\beta(S)$ is a homomorphic image of the left hand side, it follows that $Q_\beta(S)$ is regular.

It remains to be shown that (d)⇒(a) (this is the only part of the proof that uses the fact that $\mathbf{P}$ is cyclic). Assume, without loss of generality, that $E = \{\alpha_0, \alpha_1, \ldots, \alpha_m\}$, $-\alpha_0 = +\alpha_m$, $-\alpha_1 = +\alpha_0$, …, $-\alpha_m = +\alpha_{m-1}$, and $\beta = \alpha_0$. By induction on k we show that the relation

$$\mathbf{L}_k(S) \;=\; \{\, (x_\xi : \xi \in E) \in \mathbf{L}(S) \mid x_{\alpha_i} = \lambda \;\text{ for }\; k+1 \le i \le m \,\}$$

is recognizable for $0 \le k \le m$ and every S. As $\mathbf{L}_m(S) = \mathbf{L}(S)$, this proves (a).

Induction basis: $\mathbf{L}_0(S) = Q_\beta(S) \times \bigtimes_{\substack{\xi \in E \\ \xi \ne \beta}} \{\lambda\}$ and $Q_\beta(S)$ is regular, hence $\mathbf{L}_0(S)$ is recognizable (for every S).

Induction step: Assume that $0 \le k < m$ and $\mathbf{L}_k(S)$ is recognizable for every S. Thus

$$\mathbf{L}_k(S) \;=\; \bigcup_{\nu=0}^{r(S)} \bigtimes_{i=0}^{m} Q_{\nu i}(S) \,,$$

where every set $Q_{\nu i}(S)$ is regular, $Q_{\nu i}(S) \subseteq M^*_{\alpha_i}$, and $Q_{\nu i} = \{\lambda\}$ for $k < i \leq m$.

For each S', the relation $\mathbf{L}_{k+1}(S')$ can be expressed in terms of the relations $\mathbf{L}_k(S)$, $S \in \bigtimes_{j \in N} K_j$, and the finite state machine $F_n = (K_n, \Sigma_n, T_n, h_n)$, where $n = +\alpha_k = -\alpha_{k+1}$: Write $S' = (p_j : j \in N)$ and for every $q \in K_n$ denote by $\mathbf{R}(q) \subseteq M^*_{\alpha_k} \times M^*_{\alpha_{k+1}}$ the rational relation defined by the transducer

$$(K_n, M_{\alpha_k}, M_{\alpha_{k+1}}, T_n, q, \{p_n\}) \ .$$

Let $S'(q) = (q_j : j \in N)$ where $q_j = p_j$ for $j \neq n$ and $q_n = q$. By Lemma 8.11,

$$\mathbf{L}_{k+1}(S') \ = \ \bigcup_{q \in K_n} \ \bigcup_{\nu=0}^{r(S'(q))} \left[\bigtimes_{\substack{i=0 \\ i \neq k,k+1}}^{m} Q_{\nu i}(S'(q)) \ \times \ (\mathbf{R}(q) \setminus Q_{\nu k}(S'(q))) \right]$$

in the notation of Lemma 8.10. Hence $\mathbf{L}_{k+1}(S')$ is recognizable by 8.10. □

The next two lemmas, 8.12 and 8.13, are used in the proof of 8.5.

Lemma 8.12 *A family of sets $Q(S)$ is consistent (with respect to a cyclic protocol $\mathbf{P}$ and an edge $\beta \in E$) if and only if the following three conditions are satisfied:*
(a) If $(S, (x_\xi : \xi \in E)) \overset{+b}{\vdash} (S', (x'_\xi : \xi \in E))$, $b \in M_\beta$, $x_\xi = x'_\xi = \lambda$ for $\xi \neq \beta$, and $x_\beta \in Q(S)$ then $x'_\beta \in Q(S')$.
(b) If $(S, (x_\xi : \xi \in E)) \overset{-b}{\vdash} (S', (x'_\xi : \xi \in E))$, $b \in M_\beta$, $x_\xi = x'_\xi = \lambda$ for $\xi \neq \beta$, and $x_\beta \in Q(S)$ then $x'_\beta \in Q(S')$.
(c) If there is a path Γ from (S, C^0) to (S', C^0) whose no step is labelled $+b$ or $-b$, $b \in M_\beta$, then $Q(S) \subseteq Q'(S')$.

Proof. Observe that (c) is equivalent to the following, formally stronger, condition:
(d) If there is a path Γ from $(S, (x_\xi : \xi \in E))$ to $(S', (x'_\xi : \xi \in E))$, $x_\xi = x'_\xi = \lambda$ for $\xi \neq \beta$, $x_\beta \in Q(S)$ and no step in Γ is labelled $+b$ or $-b$, $b \in M_\beta$, then $x_\beta = x'_\beta \in Q(S')$.

It is clear that (a), (b) and (d) each are necessary for the consistency of $Q(S)$. To prove that the three conditions together are also sufficient, take any path Γ in the global state space, say from $(S, (x_\xi : \xi \in E))$ to $(S', (x'_\xi : \xi \in E))$, such that $x_\xi = x'_\xi = \lambda$ for $\xi \neq \beta$ and $x_\beta \in Q(S)$. Using the priority argument again, rearrange Γ so that $y_\xi = y'_\xi = \lambda$ for $\xi \neq \beta$ whenever $(S_1, (y_\xi : \xi \in E)) \overset{+b}{\vdash} (S_2, (y'_\xi : \xi \in E))$ or $(S_1, (y_\xi : \xi \in E)) \overset{-b}{\vdash} (S_2, (y'_\xi : \xi \in E))$ is a step in the rearranged path. Thus the rearranged path is a concatenation of paths to each of which either (a) or (b) or (d) applies. It follows that $x'_\beta \in Q(S')$. □

Lemma 8.13 *Let* $\mathbf{P}$ *be a cyclic CFSM protocol and* $\beta \in E$*. Then there is an algorithm to find, for any composite state* S*, every composite state* S' *for which there is a path from* (S, C^0) *to* (S', C^0) *with no step labelled* $+b$ *or* $-b$*,* $b \in M_\beta$*.*

Proof. Construct the following directed graph H. The nodes of H are the composite states of $\mathbf{P}$. There is an edge in H from S_1 to S_2 iff there exists $\xi \in E$, $\xi \neq \beta$, such that $S_1 = (p_j : j \in N)$, $S_2 = (q_j : j \in N)$, $p_j = q_j$ for $j \neq +\xi, -\xi$, and $p_{-\xi} \stackrel{-b}{\to} q_{-\xi}$, $p_{+\xi} \stackrel{+b}{\to} q_{+\xi}$ for some $b \in M_\xi$. Now S' can be reached from S by a directed path in H if and only if there is a path Γ from (S, C^0) to (S', C^0) in the global state space such that no step of Γ is labelled $+b$ or $-b$, $b \in M_\beta$. Hence the property can be decided by the standard reachability (transitive closure) algorithm in the graph H. □

Proof of 8.5. To prove that there is an algorithm to decide the consistency of a family of regular sets $Q(S)$, we construct algorithms to decide the properties (a), (b) and (c) in Lemma 8.12.

It is easy to check (a). The condition says that if $p \stackrel{+b}{\to} q$ in $F_{+\beta}$, $S = (p_j : j \in N)$, $S' = (q_j : j \in N)$, $p_j = q_j$ for $j \neq +\beta$, $p_{+\beta} = p$ and $q_{+\beta} = q$, then

$$\{\ x \mid bx \in Q(S)\ \} \ \subseteq\ Q(S')\ .$$

The inclusion is algorithmically decidable for regular sets $Q(S)$ and $Q(S')$.

A similar algorithm decides (b).

The algorithm to decide (c) has two components. The first, based on the algorithm in Lemma 8.13, finds every pair of composite states S and S' for which there is a path Γ from (S, C^0) to (S', C^0) whose no step is labelled $+b$ or $-b$, $b \in M_\beta$. The second component of the algorithm checks the inclusion $Q(S) \subseteq Q(S')$. □

Proof of 8.6. Let $\mathbf{P}$ be a cyclic CFSM protocol and $\beta \in E$. If a composite state S' is not stable then there is a consistent family of regular sets, namely the sets $Q_\beta(S)$ of Theorem 8.3, such that $\lambda \in Q_\beta(S^0)$ and $\lambda \notin Q_\beta(S')$.

Conversely, if S' is stable then $(S^0, C^0) \vdash^* (S', C^0)$; hence for any consistent family of sets $Q(S)$, regular or not, such that $\lambda \in Q(S^0)$, we must have $\lambda \in Q(S')$. □

Proof of 8.7. An algorithm to decide the deadlock problem combines two semialgorithms, one of which always terminates.

The first searches for a deadlock, using the exhaustive reachability analysis. It terminates whenever the protocol allows a deadlock.

The second semialgorithm searches for a proof of deadlock-freedom in the form of a consistent family of regular sets $Q(S)$ such that $\lambda \in Q(S^0)$ and $\lambda \notin Q(S)$ whenever S consists

solely of receive states. It terminates if the protocol is deadlock-free. $\square$

Proof of 8.8. Construct a finite state automaton F with λ-transitions as follows: The states of F are the composite states of $\mathbf{P}$. There is a λ-transition from S_1 to S_2 in F iff the graph H in the proof of 8.13 has an edge from S_1 to S_2. There is a transition from $S_1 = (p_j : j \in N)$ to $S_2 = (q_j : j \in N)$ labelled b, $b \in M_\beta$, iff $p_j = q_j$ for $j \neq +\beta$ and $p_{+\beta} \stackrel{+b}{\to} q_{+\beta}$. Write $S' \stackrel{w}{\to} S$, $w \in M_\beta^*$, if the automaton F can move from S' to S by reading w.

For a given family of regular sets $Q(S)$, $S \in \bigtimes_{j \in N} V_j$, define

$$Q(S) \;=\; \{\, y \in M_\beta^* \mid \exists\, S' \in \bigtimes_{j \in N} V_j \quad \exists\, x \in M_\beta^* \;:\; S' \stackrel{x}{\to} S \;\text{ and }\; xy \in Q(S') \,\}$$

for every $S \in \bigtimes_{j \in N} K_j - \bigtimes_{j \in N} V_j$.

Both the given sets and the newly defined ones are regular. In view of 8.5, it is now sufficient to prove this lemma:

Lemma 8.14 *If there is a consistent family $Q'(S)$, $S \in \bigtimes_{j \in N} K_j$, such that $Q'(S) = Q(S)$ for every $S \in \bigtimes_{j \in N} V_j$, then*
(a) $Q(S) \subseteq Q'(S)$ for every $S \in \bigtimes_{j \in N} K_j$; and
(b) the family $Q(S)$, $S \in \bigtimes_{j \in N} K_j$, is consistent.

Proof. (a) Let $x \in Q(S)$, $S \in \bigtimes_{j \in N} K_j - \bigtimes_{j \in N} V_j$. Write $C = (x_\xi : \xi \in E)$ where $x_\beta = x$ and $x_\xi = \lambda$ for $\xi \neq \beta$. From the definition of $Q(S)$ it follows that there are $S' \in \bigtimes_{j \in N} V_j$ and $C' = (x'_\xi : \xi \in E)$ such that $x'_\xi = \lambda$ for $\xi \neq \beta$, $x'_\beta \in Q(S') = Q'(S')$ and $(S', C') \vdash^* (S, C)$. Hence $x \in Q'(S)$ and, since $x \in Q(S)$ is arbitrary, $Q(S) \subseteq Q'(S)$.

(b) Let $C = (x_\xi : \xi \in E)$, $C' = (x'_\xi : \xi \in E)$, $x_\xi = x'_\xi = \lambda$ for $\xi \neq \beta$, $x'_\beta \in Q(S')$ and $(S', C') \vdash^* (S, C)$. It is to be shown that $x_\beta \in Q(S)$. We distinguish three cases:
I. $S \in \bigtimes_{j \in N} V_j$; then the inclusion in (a) and the consistency of $Q'(S)$ imply that $x_\beta \in Q(S)$.
II. $S \in \bigtimes_{j \in N} K_j - \bigtimes_{j \in N} V_j$ and $S' \in \bigtimes_{j \in N} V_j$. Since $(S', C') \vdash^* (S, C)$, there is a path Γ from (S', C') to (S, C) in the global state space. We assume, again, that $E = \{\alpha_0, \alpha_1, \ldots, \alpha_m\}$, $-\alpha_0 = +\alpha_m$, $-\alpha_1 = +\alpha_0$, …, $-\alpha_m = +\alpha_{m-1}$, and $\beta = \alpha_0$. As before, we rearrange the path Γ by using the highest priority at $+\alpha_m$, the next at $+\alpha_{m-1}$, etc., with the lowest priority at $+\alpha_0$. In the rearranged path, let Γ_0 be the longest prefix whose last step is

labelled $-b$, $b \in M_\beta$, and let Γ_1 be the remaining suffix of the path. Thus Γ_1 is the longest suffix whose no step is labelled $-b$, $b \in M_\beta$, and the path $\Gamma_0\Gamma_1$ is locally equal to Γ. The path Γ_0 leads from (S', C') to (S'', C''), say, with $C'' = (x''_\xi : \xi \in E)$. From the choice of priorities it follows that $S'' \in \bigtimes_{j \in N} V_j$ and $x''_\xi = \lambda$ for $\xi \neq \beta$. At the same time, Γ_1 defines a sequence of transitions from S'' to S in the automaton F; let $y \in M^*_\beta$ be the corresponding input of F, i.e. $S'' \xrightarrow{y} S$. Then $yx_\beta = x''_\beta \in Q(S'')$ and, therefore, $x_\beta \in Q(S)$.
III. $S, S' \in \bigtimes_{j \in N} K_j - \bigtimes_{j \in N} V_j$. By the definition of $Q(S')$, there are $S'' \in \bigtimes_{j \in N} V_j$ and $C'' = (x''_\xi : \xi \in E)$ such that $x''_\xi = \lambda$ for $\xi \neq \beta$, $x''_\beta \in Q(C'')$ and $(S'', C'') \vdash^* (S', C')$. Hence $(S'', C'') \vdash^* (S, C)$ and the result follows from the already proved case II.

This completes the proofs of 8.14 and 8.8. □

9 Recognizable channels for general protocols

By Theorem 8.3, the rational and the recognizable channel properties are equivalent for cyclic protocols. We begin this section by showing that the two properties differ in general.

Example 9.1 The communication graph is $0 \underset{\beta}{\overset{\alpha}{\rightrightarrows}} 1$; both M_α and M_β contain a single symbol: $M_\alpha = \{d\}$, $M_\beta = \{b\}$. The transition diagrams of the two finite state machines are in Fig. 9.1. We have

$$\begin{aligned}
\mathbf{L}((00,10)) &= \{(d^n, b^n) \mid n \geq 0\} \\
\mathbf{L}((00,11)) &= \{(d^n, b^{n+1}) \mid n \geq 0\} \\
\mathbf{L}((01,10)) &= \{(d^{n+1}, b^n) \mid n \geq 0\} \\
\mathbf{L}((01,11)) &= \{(d^n, b^n) \mid n \geq 0\}
\end{aligned}$$

All these relations are rational, but none is recognizable.
(End of Example 9.1.)

The results in section 8 (particularly Corollary 8.7) suggest the following problem.

Open problem 9.2 *Is there an algorithm to decide whether an arbitrary CFSM protocol with the rational channel property is deadlock-free?*

The present section gives a partial solution: There is an algorithm to decide deadlock-freedom for the CFSM protocols with the recognizable channel property. (This also yields another proof of 8.7.) The key property of recognizable relations needed in this theory, and not possessed by rational relations, is the decidability of inclusion.

The following Definition 9.3 and Theorems 9.4 through 9.7 are analogous to 8.4, 8.5, 8.6, 8.7 and 8.8. The results will be proved at the end of the section.

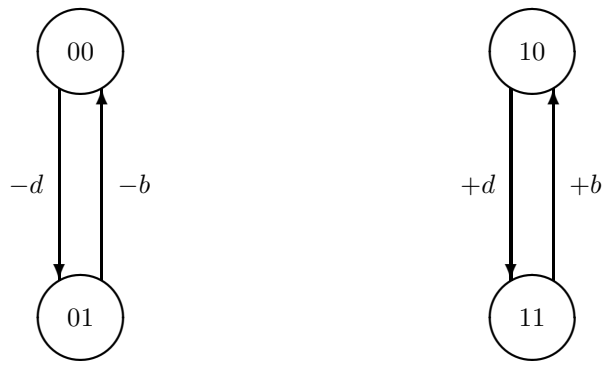


Fig. 9.1.

Definition 9.3 Let $\mathbf{P}$ be a CFSM protocol, and let $\mathbf{R}(S) \subseteq \bigtimes_{\xi \in E} M_\xi^*$ for $S \in \bigtimes_{j \in N} K_j$. Say that the relations $\mathbf{R}(S)$ are *consistent* (with respect to $\mathbf{P}$) if $(S, C) \vdash^* (S', C')$ and $C \in \mathbf{R}(S)$ imply $C' \in \mathbf{R}(S')$.

Theorem 9.4 *There is an algorithm to decide whether any given family of recognizable relations $\mathbf{R}(S)$ is consistent (with respect to a given $\mathbf{P}$).*

Theorem 9.5 *Let $\mathbf{P}$ be a CFSM protocol with the recognizable channel property. A global state (S', C') is not reachable if and only if there is a consistent family of recognizable relations $\mathbf{R}(S)$, $S \in \bigtimes_{j \in N} K_j$, such that $C^0 \in \mathbf{R}(S^0)$ and $C' \notin \mathbf{R}(S')$.*

Corollary 9.6 *The simple reachability problems (such as the deadlock problem) are algorithmically decidable for the CFSM protocols with the recognizable channel property.*

Theorem 9.7 *Let $\mathbf{P}$ be a CFSM protocol. For each $j \in N$ let $V_j \subseteq K_j$ be a set of states such that $h_j \in V_j$ and if $p_j \stackrel{-b}{\to} q_j$, $p_j, q_j \in K_j$, then $q_j \in V_j$. There is an algorithm to decide whether any given family of recognizable relations indexed by $S \in \bigtimes_{j \in N} V_j$ can be extended to a consistent family of relations $\mathbf{R}(S)$ indexed by $S \in \bigtimes_{j \in N} K_j$. Moreover, if the family can be extended than the smallest such sets $\mathbf{R}(S)$ are recognizable and can be automatically constructed.*

Theorem 9.7 should be compared with the similar result in the next theorem, which is analogous to placing intermediate assertions in program loops, as in the Floyd-Hoare invariant assertion method [Man].

Recall that a *feedback vertex set* in a directed graph is a set of vertices that intersects every directed cycle in the graph. Theorem 9.8 refers to feedback vertex sets in the *product graph* (of the protocol $\mathbf{P}$). The nodes of the graph are the composite states of $\mathbf{P}$, and the edge $S \to S'$ is in the graph iff there exists $i \in N$ such that $S = (p_j : j \in N)$, $S' = (q_j : j \in N)$, $p_j = q_j$ for $j \neq i$, and the edge $p_i \to q_i$ is in the transition diagram of F_i.

Theorem 9.8 *Let V be a feedback vertex set in the product graph of a CFSM protocol $\mathbf{P}$. There is an algorithm to decide whether any given family of recognizable sets $\mathbf{R}(S)$ indexed by $S \in V$ can be extended to a consistent family of sets $\mathbf{R}(S)$ indexed by $S \in \bigtimes_{j \in N} K_j$. Moreover, if the given family can be extended then the smallest such sets $\mathbf{R}(S)$, $S \in \bigtimes_{j \in N} K_j - V$, are recognizable and can be automatically constructed.*

The results of this section are to be used to construct automatically verifiable proofs of reachability properties for the CFSM protocols with the recognizable channel property on general communication graphs, in the same way as the results in section 8 are used for cyclic protocols. The proofs are again in the form of tables; the entries are recognizable relations. Theorems 9.7 and 9.8 help us limit the size of the tables.

The method in this section is in fact more general than the method of regular sets in section 8. Indeed, we can construct a proof that a general global state (S, C) is unreachable, whereas previously we could only prove that (S, C^0) is unreachable (i.e. that S is not stable). We can even decide certain second-order reachability properties:

Theorem 9.9 *Let $\mathbf{P}$ be a CFSM protocol with the recognizable channel property. Let $b \in M_\beta$, $p_i \in K_i$, $i = +\beta$. The message b cannot arrive at p_i if and only if there is a consistent family of recognizable relations $\mathbf{R}(S)$, $S \in \bigtimes_{j \in N} K_j$, such that $C^0 \in \mathbf{R}(S^0)$ and if $(x_\xi : \xi \in E) \in \mathbf{R}((p_j : j \in N))$ then x_β does not begin with b.*

Corollary 9.10 *The problem "Can b arrive at p_i?" is algorithmically decidable for the CFSM protocols with the recognizable channel property.*

Now we prove 9.4 through 9.9.

Proof of 9.4. Although the consistency of a family $\mathbf{R}(S)$, $S \in \bigtimes_{j \in N} K_j$, is defined in terms of the relation $\vdash^*$, it can be equivalently defined in terms of $\vdash$: The relations $\mathbf{R}(S)$ are consistent if and only if $(S, C) \vdash (S', C')$ and $C \in \mathbf{R}(S)$ imply $C' \in \mathbf{R}(S')$. In other words, $\mathbf{R}(S)$ are consistent if and only if these two conditions hold:
(a) If $S = (p_j : j \in N)$, $S' = (q_j : j \in N)$, $i = +\beta$, $p_j = q_j$ for $j \neq i$, and $p_i \xrightarrow{+b} q_i$ in F_i, then

$$\{ \ (x'_\xi : \xi \in E) \ \mid \ \exists (x_\xi : \xi \in E) \in \mathbf{R}(S) : x_\xi = x'_\xi \text{ for } \xi \neq \beta \text{ and } x_\beta = b x'_\beta \ \} \ \subseteq \ \mathbf{R}(S') \ .$$

(b) If $S = (p_j : j \in N)$, $S' = (q_j : j \in N)$, $i = -\beta$, $p_j = q_j$ for $j \neq i$, and $p_i \stackrel{-b}{\to} q_i$ in F_i, then

$$\{ \ (x'_\xi : \xi \in E) \ \mid \ \exists (x_\xi : \xi \in E) \in \mathbf{R}(S) : x_\xi = x'_\xi \text{ for } \xi \neq \beta \text{ and } x_\beta b = x'_\beta \ \} \ \subseteq \ \mathbf{R}(S') \ .$$

Since these inclusions are decidable for recognizable relations, both (a) and (b) are decidable. □

Proof of 9.5. The proof is similar to that of 8.6. If (S', C') is not reachable, then the relations $\mathbf{L}(S)$ of Definition 8.2 fulfill the condition. Namely, $\mathbf{L}(S)$ are consistent, $C^0 \in \mathbf{L}(S^0)$ and $C' \notin \mathbf{L}(S')$.

Conversely, if (S', C') is reachable then no consistent family of relations $\mathbf{R}(S)$, recognizable or not, satisfies $C^0 \in \mathbf{R}(S^0)$ and $C' \notin \mathbf{R}(S')$. □

Proof of 9.6. As in the proof of 8.7, we combine two semialgorithms, one of which always terminates.

Given a global state (S', C'), the first semialgorithm searches for a path from (S^0, C^0) to (S', C'). It terminates whenever (S', C') is reachable.

The second semialgorithm searches for a proof of non-reachability of (S', C'), in the form of a consistent family of recognizable relations $\mathbf{R}(S)$ such that $C^0 \in \mathbf{R}(S^0)$ and $C' \notin \mathbf{R}(S')$. Since the protocol has the recognizable channel property, the semialgorithm terminates whenever (S', C') is not reachable. □

Proof of 9.7. Define

$$W^+(q,p) \ = \ \{ \ b_0 b_1 \ldots b_n \mid b_i \in M_\xi \ \text{ for } \ 0 \leq i \leq n \ \text{ and } \ q \xrightarrow{+b_0+b_1\ldots+b_n} p \ \}$$

for $q, p \in K_{+\xi}$, and

$$\mathbf{W}^+(S', S) \ = \ \bigtimes_{j \in N} W^+(q_j, p_j)$$

for $S' = (q_j : j \in N)$ and $S = (p_j : j \in N)$. For a given family of recognizable relations $\mathbf{R}(S)$, $S \in \bigtimes_{j \in N} V_j$, define

$$\mathbf{R}(S) \ = \ \bigcup_{S' \in \bigtimes_{j \in N} V_j} \{ \ (y_\xi : \xi \in E) \mid \exists (x_\xi : \xi \in E) \in \mathbf{W}^+(S', S) \ : \ (x_\xi y_\xi : \xi \in E) \in \mathbf{R}(S') \ \}$$

for $S \in \bigtimes_{j \in N} K_j - \bigtimes_{j \in N} V_j$. All the relations $\mathbf{R}(S)$, $S \in \bigtimes_{j \in N} K_j$, are recognizable, and Theorem 9.7 follows from this lemma:

Lemma 9.11 *If there is a consistent family* $\mathbf{R}'(S)$, $S \in \bigtimes_{j\in N} K_j$, *such that* $\mathbf{R}'(S) = \mathbf{R}(S)$ *for every* $S \in \bigtimes_{j\in N} V_j$, *then*

(a) $\mathbf{R}(S) \subseteq \mathbf{R}'(S)$ *for every* $S \in \bigtimes_{j\in N} K_j$*; and*

(b) the family $\mathbf{R}(S)$, $S \in \bigtimes_{j\in N} K_j$, *is consistent.*

Proof of 9.11. (a) Let $C \in \mathbf{R}(S)$, $S \in \bigtimes_{j\in N} K_j - \bigtimes_{j\in N} V_j$. By the definition of $\mathbf{R}(S)$, there is $S' \in \bigtimes_{j\in N} V_j$ such that $(S', C') \vdash^* (S, C)$ for some $C' \in \mathbf{R}(S')$. Since the relations $\mathbf{R}'(S)$ are consistent, it follows that $C \in \mathbf{R}'(S)$. Hence $\mathbf{R}(S) \subseteq \mathbf{R}'(S)$.
(b) Let $(S', C') \vdash^* (S, C)$ and $C' \in \mathbf{R}(S')$. We want to prove that $C \in \mathbf{R}(S)$. We distinguish three cases:
I. $S \in \bigtimes_{j\in N} V_j$. Then the inclusion in (a) and the consistency of $\mathbf{R}'(S)$ imply $C \in \mathbf{R}(S)$.
II. $S \in \bigtimes_{j\in N} K_j - \bigtimes_{j\in N} V_j$ and $S' \in \bigtimes_{j\in N} V_j$. Since $(S', C') \vdash^* (S, C)$, there is a path Γ from (S', C') to (S, C). There are two paths Γ' and Γ'' such that $\Gamma'\Gamma''$ is locally equal to Γ, the end state (S'', C'') of Γ' satisfies $S'' \in \bigtimes_{j\in N} V_j$, and all the steps in Γ'' are receptions (i.e. are labelled $+b$). Since $\mathbf{R}'(S)$ are consistent and $\mathbf{R}'(S') = \mathbf{R}(S')$ and $\mathbf{R}'(S'') = \mathbf{R}(S'')$, it follows that $C'' \in \mathbf{R}(S'')$. The path Γ'' defines a vector $(x_\xi : \xi \in E) \in \mathbf{W}^+(S'', S)$, and with $C = (y_\xi : \xi \in E)$ we have $(x_\xi y_\xi : \xi \in E) = C'' \in \mathbf{R}(S'')$. By the definition of $\mathbf{R}(S)$ we get $C \in \mathbf{R}(S)$.
III. $S, S' \in \bigtimes_{j\in N} K_j - \bigtimes_{j\in N} V_j$. By the definition of $\mathbf{R}(S')$, there is a global state (S'', C'') such that $S'' \in \bigtimes_{j\in N} V_j$, $(S'', C'') \vdash^* (S', C')$ and $C'' \in \mathbf{R}(S'')$. Hence $(S'', C'') \vdash^* (S, C)$ and we apply the already proved case II. □

Proof of 9.8. We start with the given recognizable relations $\mathbf{R}(S)$, $S \in V$, and first define relations $\mathbf{R}(S)$, for $S \notin V$, as follows. For $S \in \bigtimes_{j\in N} K_j - V$, let $\mathbf{R}(S)$ be the set of all those $C \in \bigtimes_{\xi\in E} M_\xi^*$ for which there are $S' \in V$, $C' \in \mathbf{R}(S')$, and a path from (S', C') to (S, C) such that no composite state S'' along the path (except S') belongs to V. Since V is a feedback vertex set, no such path can pass through the same composite state twice. Hence the length of all such paths is bounded, and therefore the sets $\mathbf{R}(S)$, $S \in \bigtimes_{j\in N} K_j - V$, are recognizable and automatically constructible. The result now follows from this lemma:

Lemma 9.12 *If there is a consistent family* $\mathbf{R}'(S)$, $S \in \bigtimes_{j\in N} K_j$, *such that* $\mathbf{R}'(S) = \mathbf{R}(S)$

for every $S \in V$, then
(a) $\mathbf{R}(S) \subseteq \mathbf{R}'(S)$ for every S; and
(b) the family $\mathbf{R}(S)$, $S \in \bigtimes_{j \in N} K_j$, is consistent.

Proof of 9.12 is similar to, but simpler than, that of 9.11.
(a) Let $C \in \mathbf{R}(S)$, $S \in \bigtimes_{j \in N} K_j - V$. There are $S' \in V$ and $C' \in \mathbf{R}(S')$ such that $(S', C') \vdash^* (S, C)$. Since the relations $\mathbf{R}'(S)$ are consistent, $C \in \mathbf{R}'(S)$. Hence $\mathbf{R}(S) \subseteq \mathbf{R}'(S)$.
(b) Let $(S', C') \vdash^* (S, C)$ and $C' \in \mathbf{R}(S')$. We want to prove that $C \in \mathbf{R}(S)$. We distinguish three cases:
I. $S \in V$. Then the inclusion in (a) and the consistency of $\mathbf{R}'(S)$ imply $C \in \mathbf{R}(S)$.
II. $S \notin V$ and $S' \in V$. There is a path Γ from (S', C') to (S, C). Let Γ'', from (S'', C'') to (S, C), be the shortest suffix of Γ such that $S'' \in V$. Thus $\Gamma = \Gamma'\Gamma''$, Γ' leads from (S', C') to (S'', C''), and $C'' \in \mathbf{R}(S'')$. No composite state along Γ'' (except S'') belongs to V, hence $C \in \mathbf{R}(S)$ by the definition of $\mathbf{R}(S)$.
III. $S \notin V$ and $S' \notin V$. By the definition of $\mathbf{R}(S')$, there is a global state (S'', C'') such that $S'' \in V$, and a path from (S'', C'') to (S', C'). Hence there is a path from (S'', C'') to (S, C), and the result follows from case II. □

Proof of 9.9. If b cannot arrive at p_i then the relations $\mathbf{L}(S)$ of Definition 8.2 fulfill the condition. Conversely, if there is a consistent family of recognizable relations $\mathbf{R}(S)$, $S \in \bigtimes_{j \in N} K_j$, such that $C^0 \in \mathbf{R}(S^0)$ and x_β does not begin with b whenever $(x_\xi : \xi \in E) \in \mathbf{R}((p_j : j \in N))$, then, by Theorem 9.5, no global state $((p_j : j \in N), (x_\xi : \xi \in E))$ in which x_β begins with b is reachable. In other words, b cannot arrive at p_i. □

Proof of 9.10. Again it is sufficient to show that if b cannot arrive at p_i then there is an algorithmically verifiable proof. This follows from the previous results in this section and from the following: There is an algorithm to decide, for every recognizable relation $\mathbf{R} \subseteq \bigtimes_{\xi \in E} M_\xi^*$, every $\beta \in E$ and every $b \in M_\beta$, whether there is $(x_\xi : \xi \in E) \in \mathbf{R}$ such that x_β begins with b. □

10 Abstract flow control in general graphs

We now return to the idea of abstract flow control, introduced in section 5 for cyclic graphs. Recall that our first aim is to limit the number of locally equal paths to be examined by the reachability algorithms. This alone is easily achieved; we can order all nodes of the

communication graph by assigning them distinct priorities, and thus select a *unique* path in every class of locally equal paths.

However, not all such priority assignments are of equal value. Our second aim is to choose locally equal paths that use a small number of global states. Two methods for making the choice, leading to two different priority schemes, are described in this section. Then the priority arguments are applied to give a partial solution of the reachability problem for the rational channel CFSM protocols.

Let Γ be a path in the global state space of a CFSM protocol. Suppose that $\Phi(S,C)$ is a proposition applicable to every global state (S,C); that is, $\Phi(S,C)$ is a (true or false) statement for every (S,C). Say that $\Phi(S,C)$ is true *frequently along* Γ if $\Phi(S,C)$ is true for at least one of every two consecutive global states along Γ. In particular, if β is an edge in the communication graph and the statement "if $C = (x_\xi : \xi \in E)$ then $x_\beta = \lambda$" is true frequently along Γ, then the transmissions and receptions on the channel β are tightly coupled in the execution described by Γ; in other words, every symbol sent on β is received at once (in the next step).

The first result uses collections of noncrossing boundaries in the communication graph; the concept is somewhat similar to the laminar collection of (or valuation on) directed cuts in a directed graph, in the sense of Lucchesi and Younger [Luc]. Let $G = (N,E)$ be a directed graph. For $A \subseteq N$ denote

$$\begin{aligned} \partial^-(A) &= \{\, \xi \in E \mid +\xi \in A \ \text{ and } \ -\xi \notin A \,\} \\ \partial^+(A) &= \{\, \xi \in E \mid -\xi \in A \ \text{ and } \ +\xi \notin A \,\} \end{aligned}$$

and call the sets $\partial^-(A)$ and $\partial^+(A)$ the *negative* and the *positive boundary* of A.

A set Ψ of subsets of N is *smooth* if for all $A, B \in \Psi$ we have (i) $A \subseteq B$ or (ii) $B \subseteq A$ or (iii) $A \cap B = \emptyset$ and $\partial^-(A \cup B) = \partial^-(A) \cup \partial^-(B)$.

Theorem 10.1 *Let $G = (N,E)$ be the communication graph of a CFSM protocol and let Ψ be a smooth set of subsets of N. For every path that ends in a global state with empty channels, there exists a locally equal path along which the following is frequently true:*

$$\forall A \in \Psi \ \ \exists \beta \in \partial^-(A) \ : \ \text{ if } \ C = (x_\xi : \xi \in E) \ \text{ then } \ x_\beta = \lambda \, .$$

Proof of 10.1. Order the sets in Ψ in a sequence $A_0, A_1, \ldots, A_n$ such that if $A_i \subseteq A_j$ then $i \le j$. Set $B_0 = A_0$ and $B_k = A_k - \bigcup_{i=0}^{k-1} A_i$ for $k > 0$. Let Γ be a path ending in a global state with empty channels. We rearrange Γ by executing the processes in B_0 with the highest priority, those in B_1 with the second highest, etc.

Formally, if Γ contains two adjacent steps

$$\Gamma_1 \ \ : \ \ (S_1, C_1) \overset{e_1}{\vdash} (S_2, C_2)$$

$$\Gamma_2 \quad : \quad (S_2, C_2) \overset{e_2}{\vdash} (S_3, C_3)$$

such that $\mathrm{Im}_{i_1}(\Gamma_1)$ and $\mathrm{Im}_{i_2}(\Gamma_2)$ are nontrivial paths, $i_1 \in B_{j_1}$, $i_2 \in B_{j_2}$, $j_1 > j_2$, and if it is *not* the case that $e_2 = +b$, $b \in M_\beta$, $C_1 = (x_\xi : \xi \in E)$ and $x_\beta = \lambda$, then we replace the subpath $\Gamma_1\Gamma_2$ in Γ by the path $(S_1, C_1) \overset{e_2}{\vdash} (S_4, C_4) \overset{e_1}{\vdash} (S_3, C_3)$ for a suitable (S_4, C_4). We repeat the same with the new path, etc., until no further transformation is possible. Let Γ' be the path constructed by this process. We wish to show that

$$\forall A \in \Psi \;\; \exists \beta \in \partial^-(A) \;:\; \text{ if } \; C = (x_\xi : \xi \in E) \; \text{ then } \; x_\beta = \lambda$$

frequently along Γ'.

If not then there are two consecutive global states (S, C) and (S', C') in Γ' and two sets $A, A' \in \Psi$ such that $C = (x_\xi : \xi \in E)$, $C' = (x'_\xi : \xi \in E)$ and

$$\begin{aligned} \forall \xi \in \partial^-(A) \quad &: \quad x_\xi \neq \lambda \\ \forall \xi \in \partial^-(A') \quad &: \quad x'_\xi \neq \lambda \end{aligned}$$

First observe that we can assume, without loss of generality, that $A = A'$. Indeed, if the move from (S, C) to (S', C') is a reception on a channel $\beta \in \partial^-(A)$ then

$$\forall \xi \in \partial^-(A') \;:\; x_\xi \neq \lambda,$$

and if the move from (S, C) to (S', C') is not a reception on a channel in $\partial^-(A)$ then

$$\forall \xi \in \partial^-(A) \;:\; x'_\xi \neq \lambda.$$

Now assume $A = A'$. Since Γ' ends in a global state with empty channels, there is a later step $\overset{+b}{\vdash}$ in Γ', for some $b \in M_\beta$, $\beta \in \partial^-(A)$. Taking the first such step, say $(S_1, C_1) \overset{+b}{\vdash} (S_2, C_2)$, we get a contradiction with the construction of Γ': We have $b \in M_\beta$, $\beta \in \partial^-(A)$ and from the properties of Ψ it follows that $+\beta \in B_i$, $-\beta \in B_j$, $i < j$. Hence the step $(S_1, C_1) \overset{+b}{\vdash} (S_2, C_2)$ could be exchanged with the previous step in Γ', contrary to the assumption that no further transformation is applicable to Γ'. $\square$

Observe that Theorem 5.2 follows immediately from 10.1. Indeed, with the notation of 5.2, there is a smooth set Ψ of subsets of N such that $\{\beta\} = \partial^+(A)$ for every $A \in \Psi$ and

$$\{\; \partial^-(A) \mid A \in \Psi \;\} \;=\; \{\; \{\xi\} \mid \xi \in E - \{\beta\} \;\}\;.$$

Fig. 10.1 shows such a set Ψ for a cyclic protocol whose graph has four nodes.

The priorities in the proof of Theorem 10.1 are interpreted in the "standard" way: A node executes (i.e. its finite state machine makes a move) if and only if it is not blocked (waiting for input) and all the nodes with higher priorities are blocked.

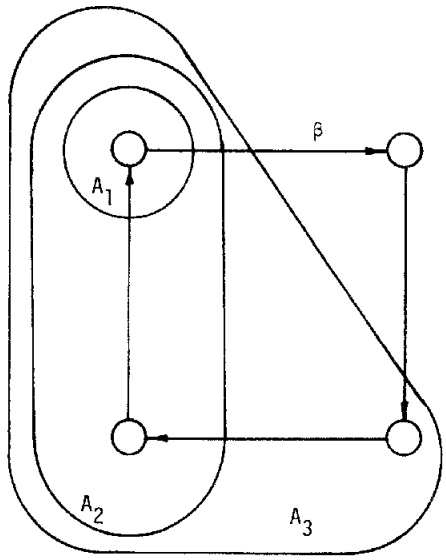
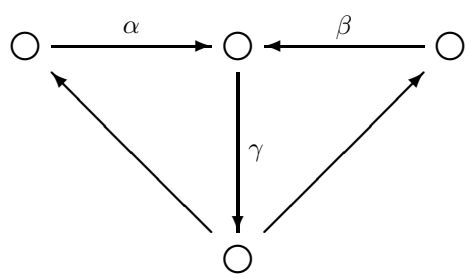


Fig. 10.1. The smooth set $\{ A_1, A_2, A_3 \}$**.**

A different priority scheme arises when we apply Theorem 10.1 recursively, in a divide-and-conquer manner.

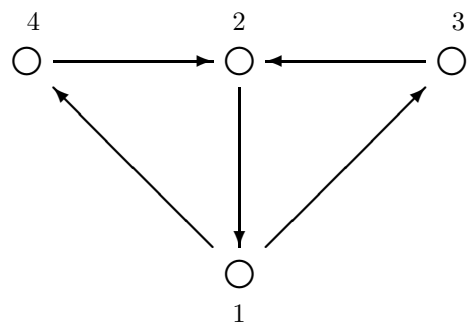
Example 10.2 Consider the following communication graph.



Every execution that begins and ends with empty channels can be reordered so that α, β and γ are frequently empty. However, such a reordering cannot be achieved by the standard priority scheme.

Instead, we first apply Theorem 10.1 to the set $\{\{+\gamma\}\}$, to make γ frequently empty. Then we restrict all subsequent reorderings to the remaining nodes of the graph; we next apply 10.1 to the set $\Psi = \{\{-\beta, +\beta\}\}$; this makes α frequently empty. Then, in the graph with the two nodes $-\beta$ and $+\beta$, we apply 10.1 to $\Psi = \{\{+\beta\}\}$, to make β frequently empty.

Another way of describing the new execution is to say that the nodes are ordered $+\gamma$, $-\gamma$, $-\beta$, $-\alpha$, from the highest to the lowest priority. However, the priorities now have a different meaning. In the standard scheme, the unblocked process with the highest priority executes. In the present scheme, that unblocked process executes on which the process with the highest priority is (directly or indirectly) blocked. In our example, the priorities are as follows:



If 2 is blocked on 4 and both 4 and 3 are unblocked, then 4 (not 3) executes; in the standard scheme, 3 would execute.
(End of Example 10.2.)

Clearly the priority schemes, as well as any other abstract flow control methods, improve the efficiency of the exhaustive reachability analysis by reducing the number of global states that the analysis must enumerate. It is difficult to make any quantitative claims about the efficiency gains because, as Brand and Zafiropulo [Bra] note when they evaluate two analysis methods, "in both approaches a protocol can be analyzed successfully only if its behavior is far from the worst case, as is true for protocols designed in practice." However, in the context of the theory developed in this paper we can prove qualitative claims about the *existence* of algorithms (rather than their *cost*).

We have already seen (in section 8) how a priority scheme can be used to construct an algorithm to solve the deadlock problem for the cyclic protocols with the rational channel property. In the remainder of this section we shall see, on two examples, that the same can be done for some other communication graphs.

Theorem 10.3 *The problem "Is a given composite state stable?" is algorithmically decidable for the CFSM protocols with the rational channel property and the communication graph* $0 \underset{\beta}{\overset{\alpha}{\rightrightarrows}} 1$.

Theorem 10.4 *The problem "Is a given composite state stable?" is algorithmically decidable for the CFSM protocols with the rational channel property and the communication graph in Fig. 10.2(a).*

The same result can be proved for the graphs in Fig. 10.2(b), (c), (d) and other similar ones. On the other hand, it is not known (to the author) whether the stable composite state problem is decidable for the CFSM protocols with the rational channel property and the communication graphs in Fig. 10.3(a) and (b).

In the forthcoming proofs, we say that a family of relations $\mathbf{R}(S)$, $S \in \bigtimes_{j \in N} K_j$, is *consistent relative to a restriction* if this condition holds: if $(S, C) \vdash (S', C')$, $C \in \mathbf{R}(S)$, and both (S, C) and (S', C') satisfy the restriction, then $C' \in \mathbf{R}(S')$.

Proof of 10.3. If a composite state is stable then its stability is verified by the exhaustive reachability analysis. Thus it suffices to construct a semialgorithm that verifies non-stability and terminates whenever the composite state is not stable. We show that there is an algorithmically verifiable proof of non-stability for every non-stable composite state; the semialgorithm then simply generates proof candidates until it finds a correct one.

Choosing $\Psi = \{\{1\}\}$ in Theorem 10.1, we can restrict our attention to the paths along which frequently α or β is empty. Thus for every non-stable composite state S' there is a proof of non-stability of S', in the form of a family of relations $\mathbf{R}(S)$, $S \in \bigtimes_{j \in N} K_j$, that are consistent relative to the restriction "$|x_\alpha| \leq 1$ or $|x_\beta| \leq 1$" and such that $C^0 \in \mathbf{R}(S^0)$ and $C^0 \notin \mathbf{R}(S')$. The consistency is algorithmically verifiable when the relations are recognizable; hence the result follows from this lemma:

Lemma 10.5 *If* $\mathbf{R} \subseteq M_\alpha^* \times M_\beta^*$ *is a rational relation then the relation*

$$\mathbf{R}' = \{\ (x_\alpha, x_\beta) \in \mathbf{R} \mid |x_\alpha| \leq 1 \text{ or } |x_\beta| \leq 1\ \}$$

is recognizable.

Proof of 10.5. For every $x \in M_\alpha^*$ the relation

$$\mathbf{R}^\beta(x) = \{\ (x_\alpha, x_\beta) \in \mathbf{R} \mid x_\alpha = x\ \}$$

is recognizable; similarly, for every $y \in M_\beta^*$ the relation

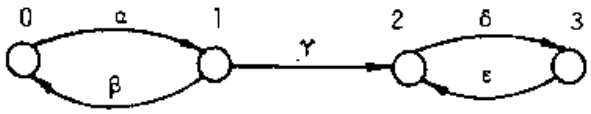
$$\mathbf{R}^\alpha(y) = \{\ (x_\alpha, x_\beta) \in \mathbf{R} \mid x_\beta = y\ \}$$

is recognizable. Since the relation $\mathbf{R}'$ is equal to

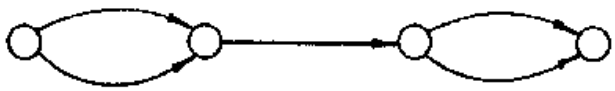
$$\mathbf{R}^\beta(\lambda) \cup \mathbf{R}^\alpha(\lambda) \cup \bigcup_{x_\alpha \in M_\alpha} \mathbf{R}^\beta(x_\alpha) \cup \bigcup_{x_\beta \in M_\beta} \mathbf{R}^\alpha(x_\beta)\ ,$$

it is recognizable.

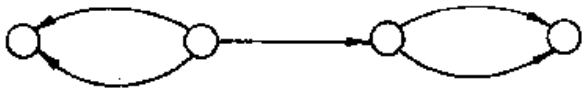
This completes the proofs of 10.5 and 10.3. □



(a)



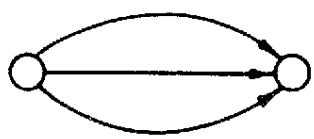
(b)



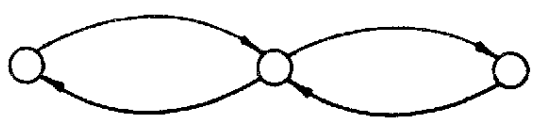
(c)

(d)

Fig. 10.2.



(a)



(b)

Fig. 10.3.

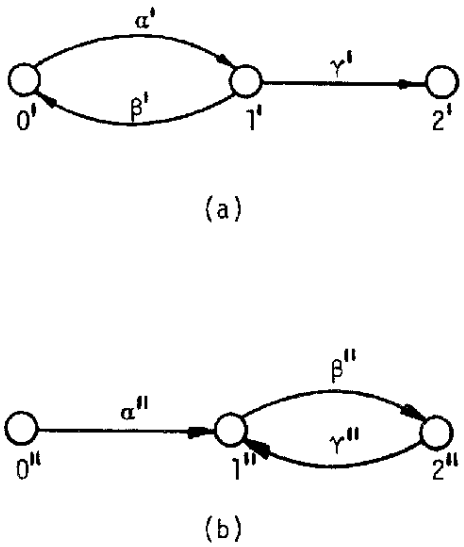


Fig. 10.4.

In the forthcoming proof of 10.4 we split the graph in Fig. 10.2(a) into the two graphs in Fig. 10.4. For the given CFSM protocol $\mathbf{P}$ (with the communication graph in Fig. 10.2(a)) and for an arbitrary deterministic (complete) finite automaton F over the alphabet M_γ, we define two protocols $\mathbf{P}'(F)$ and $\mathbf{P}''(F)$ as follows: The protocol $\mathbf{P}'(F)$ has the communication graph of Fig. 10.4(a), the finite state machines at the nodes $0'$ and $1'$ are the same as those at 0 and 1 in $\mathbf{P}$ and the machine at $2'$ is F (with every label in its transition diagram prefixed by $+$). The communication graph of $\mathbf{P}''(F)$ is as in Fig. 10.4(b), the finite state machine at $0''$ is F (with every label prefixed by $-$) and the machines at $1''$ and $2''$ are the same as those at 2 and 3 in $\mathbf{P}$.

Lemma 10.6 *Let $\mathbf{P}$ be a CFSM protocol with the communication graph in Fig. 10.2(a), and let (p_0, p_1, p_2, p_3) be a composite state of $\mathbf{P}$. Assume that there exist a deterministic finite automaton (over M_γ) and a set U of its states such that*
(a) if p is a state of F, $p \notin U$, then (p_0, p_1, p) is not stable for $\mathbf{P}'(F)$; and
(b) if $p \in U$ then (p, p_2, p_3) is not stable for $\mathbf{P}''(F)$.
Then (p_0, p_1, p_2, p_3) is not stable (for $\mathbf{P}$).

Proof of 10.6. Suppose that $S' = (p_0, p_1, p_2, p_3)$ is stable, i.e. $(S^0, C^0) \vdash^* (S', C^0)$. We use higher priority for the nodes 0 and 1 to get two paths Γ_0 and Γ_1 and a channel content $C' = (x_\xi : \xi \in E)$ such that
(1) $x_\xi = \lambda$ for $\xi \neq \gamma$ (where γ is the edge from 1 to 2 in Fig. 10.2(a));
(2) Γ_0 leads from (S^0, C^0) to $((p_0, p_1, h_2, h_3), C')$;
(3) Γ_1 leads from $((p_0, p_1, h_2, h_3), C')$ to (S', C^0); and
(4) the images $\mathrm{Im}_2(\Gamma_0)$, $\mathrm{Im}_3(\Gamma_0)$, $\mathrm{Im}_0(\Gamma_1)$, $\mathrm{Im}_1(\Gamma_1)$ are all trivial paths.

Let F be any deterministic finite automaton over M_γ, and U a set of its states. Let p be the state of F to which F moves from its initial state by reading x_γ. The composite state (p_0, p_1, p) is stable for $\mathbf{P}'(F)$, and (p, p_2, p_3) is stable for $\mathbf{P}''(F)$. Thus (a) and (b) in 10.6 cannot be both true. □

The crucial step in the proof of 10.4 is the following lemma, which (together with 10.6) shows that for every non-stable composite state of $\mathbf{P}$ there is an algorithmically verifiable proof of its non-stability.

Lemma 10.7 *Let $\mathbf{P}$ be a CFSM protocol with the rational channel property and the communication graph in Fig. 10.2(a). If a composite state (p_0, p_1, p_2, p_3) of $\mathbf{P}$ is not stable then there exist a deterministic finite automaton F and a set U of its states such that*
(a) there is a family of recognizable relations $\mathbf{R}'(S')$ indexed by the composite states S' of the protocol $\mathbf{P}'(F)$, consistent relative to the restriction "$|x_{\alpha'}| \leq 1$ and $|x_{\gamma'}| \leq 1$", such that $C^0 \in \mathbf{R}'(S^0)$ and $C^0 \notin \mathbf{R}'(p_0, p_1, p))$ for every state p of F not in U;

(b) there is a family of recognizable relations $\mathbf{R}''(S'')$ indexed by the composite states S'' of the protocol $\mathbf{P}''(F)$, consistent relative to the restriction "$|x_{\alpha''}| \leq 1$ and $|x_{\gamma''}| \leq 1$", such that $C^0 \in \mathbf{R}''(S^0)$ and $C^0 \notin \mathbf{R}''(p, p_2, p_3))$ for every state $p \in U$.

Proof of 10.7. Let Q be the set $Q_\gamma((p_0, p_1, h_2, h_3))$ of Theorem 8.3; that is,

$$Q \;=\; \{\, x_\gamma \in M_\gamma^* \mid (S^0, C^0) \vdash^* ((p_0, p_1, h_2, h_3), (x_\xi : \xi \in E)) \;\text{ and }\; x_\xi = \lambda \;\text{ for }\; \xi \neq \gamma \,\}.$$

Since $\mathbf{P}$ has the rational channel property, Q is regular. There is a deterministic finite automaton F to recognize Q; let U be the set of accepting states of F. To define the relations $\mathbf{R}'((q_0, q_1, p))$ and $\mathbf{R}''((p, q_2, q_3))$, we use the relations $\mathbf{L}(S)$ of Definition 8.2. Denote h_0' the initial state of F.

Define

$$\begin{aligned}
&\mathbf{R}'((q_0, q_1, p)) = \\
&\{\, (x_{\alpha'}, x_{\beta'}, x_{\gamma'}) \;\mid\; \exists\, (x_\alpha, x_\beta, x_\gamma, x_\delta, x_\varepsilon) \in \mathbf{L}((q_0, q_1, h_2, h_3)) \;\; \exists\, y \in M_\gamma^* \;: \\
&\qquad\qquad x_{\alpha'} = x_\alpha \,, x_{\beta'} = x_\beta \,, x_\delta = x_\varepsilon = \lambda \,, |x_{\alpha'}| \leq 1 \,, |x_{\gamma'}| \leq 1 \,, \\
&\qquad\qquad h_0' \xrightarrow{y} p \;\text{ in }\; F \text{ and }\; y x_{\gamma'} = x_\gamma \,\}
\end{aligned}$$

for every composite state (q_0, q_1, p) of $\mathbf{P}'(F)$. If (p, q_2, q_3) is a composite state of $\mathbf{P}''(F)$ such that a state in U can be reached from p in F then define

$$\begin{aligned}
&\mathbf{R}''((p, q_2, q_3)) = \\
&\{\, (x_{\alpha''}, x_{\beta''}, x_{\gamma''}) \;\mid\; \forall\, y \in M_\gamma^* \;\; \forall\, p' \in U \;:\; p \xrightarrow{y} p' \implies \\
&\qquad\qquad \exists\, (x_\alpha, x_\beta, x_\gamma, x_\delta, x_\varepsilon) \in \mathbf{L}((p_0, p_1, q_2, q_3)) \;: x_\alpha = x_\beta = \lambda \,, \\
&\qquad\qquad x_{\beta''} = x_\delta \,, x_{\gamma''} = x_\varepsilon \,, |x_{\gamma''}| \leq 1 \,, |x_{\alpha''}| \leq 1 \,, \text{ and }\; x_{\alpha''} y = x_\gamma \,\} \,,
\end{aligned}$$

and if no state in U can be reached from p, define

$$\mathbf{R}''((p, q_2, q_3)) = \{\, (x_{\alpha''}, x_{\beta''}, x_{\gamma''}) \;\mid\; |x_{\alpha''}| \leq 1 \text{ and }\; |x_{\gamma''}| \leq 1 \,\} \,.$$

Since $\mathbf{P}$ has the rational channel property, $\mathbf{L}(S)$ are rational, and therefore $\mathbf{R}'(S')$ and $\mathbf{R}''(S'')$ are recognizable.

The consistency of $\mathbf{R}'(S')$ and $\mathbf{R}''(S'')$ follows from the consistency of $\mathbf{L}(S)$ and from the definition of F and U. It also follows from the definition of F and U that if $C^0 \in \mathbf{R}'((p_0, p_1, p))$ then $p \in U$, and that if $C^0 \in \mathbf{R}''((p, p_2, p_3))$ then $p \notin U$. This completes the proof of 10.7. □

Proof of 10.4. As in the proof of 10.3, it suffices to show that for every non-stable composite state there is an algorithmically verifiable proof of its non-stability. By 10.7 and 10.6 there is such a proof, consisting of F, U, the family $\mathbf{R}'(S')$ and the family $\mathbf{R}''(S'')$.

Indeed, if p is a state of F not in U then the family $\mathbf{R}'(S')$ is a proof that (p_0, p_1, p) is not stable for $\mathbf{P}'(F)$ (by the priority argument applied to the graph in Fig. 10.4(a), every stable composite state is reachable by a path along which frequently α' and γ' are empty). Similarly, the priority argument applied to the graph in Fig. 10.4(b) shows that the family $\mathbf{R}''(S'')$ is a proof that (p, p_2, p_3) is not stable for $\mathbf{P}''(F)$ whenever $p \in U$. □

11 Recapitulation and conclusions

The theory of communicating finite state machines, or, more precisely, of finite state machines connected by unbounded queues, is emerging as a valuable tool for the specification and correctness analysis of communication protocols operating over channels with indefinite delays. Although the CFSM model is very simple, it is rich enough to encompass certain basic protocol properties, which are expressed as reachability properties in the global state space.

The reachability properties cannot be automatically verified in the class of all CFSM protocols; in other words, the reachability problems are (algorithmically) undecidable. However, since the usefulness of the model is greatly enhanced by its amenability to automated analysis, it is well worthwhile to look for classes of CFSM protocols in which the problems are decidable. Traditionally, the emphasis has been on the class of the protocols with the bounded channel property.

The present paper advances our understanding of the question "What makes the reachability problems in the CFSM theory undecidable?" The paper contributes three new concepts to the theory: Affinity of SR-machines, simple-channel properties, and abstract flow control.

The results about affine SR-machines point out close ties between the traditional automata theory and the theory of CFSM protocols. It is also shown (in section 6) that, although many interesting properties of communicating SR-machines are undecidable, some become decidable under additional restrictions (affinity in this case).

Similarly, the results about simple-channel (recognizable channel and rational channel) properties demonstrate how *some* protocols with unbounded channels can be automatically analyzed, although the problems are undecidable for *general* protocols. The simple-channel restrictions formally express the observation that common protocols do not make use of the full generality of the CFSM model. "Protocols with unbounded channels usually use them in a simple manner, which makes them worth considering" ([Bra], p. 10). The results in this paper suggest a new formalism for protocol description (CFSM augmented with channel expressions) together with algorithms for automated analysis of the protocols so described.

It should be pointed out that a proof of, say, deadlock-freedom in the form of a table of

recognizable relations can be potentially advantageous even for a protocol with the bounded channel property. Indeed, it can happen that the reachable global states are separated from the deadlocked ones by a consistent family of recognizable relations that are described by short expressions, while at the same time the complete list of all reachable global states is very large.

The theory of "recognizable proofs" (i.e. proofs based on recognizable relations) is all ready for use; the theory of "rational proofs", on the other hand, is not well understood. The key open question is whether reachability problems are algorithmically decidable for protocols with the rational channel property. The problem is answered in the affirmative for cyclic protocols in section 8, and for several other simple communication graphs in section 10.

The aim of the abstract flow control, as defined and studied in this paper, is to limit the redundancy in the global state space, thereby improving the efficiency of the algorithms that decide the reachability properties. Abstract flow control methods should exploit the topology of the communication graph, as do the two priority schemes proposed in section 10.

In section 10 it is shown how the priority schemes lead to qualitative gains: They allow us to construct algorithms for solving reachability problems for the rational-channel CFSM protocols with some communication graphs. Abstract flow control methods yield quantitative gains as well, but these are difficult to estimate in any meaningful way for general protocols. Perhaps a fruitful approach would be to study *algorithms* for finding optimal abstract flow control methods, or, for the sake of concreteness, optimal priority assignments. For example, one can formulate the optimization problem of finding (for an arbitrary communication graph) the priority assignment that minimizes a cost function, which measures the number of "needlessly reachable" global states. But that, as Kipling says, is another story.

Acknowledgement. The work reported here is a part of my Ph.D. thesis supervised by K. Culik II. I wish to thank him for encouragement and many fruitful discussions.

A Appendix: Post's tag systems

The tag systems are first mentioned by Post [Pos] as a source of possibly undecidable problems. The undecidability is actually proved by Minsky ([Mi1], [Mi2]).

A *tag system* is a 3-tuple $\mathbf{T} = (\Sigma, g, w_0)$ where Σ is a finite alphabet, g is a function from Σ to Σ^* and $w_0 \in \Sigma^*$. Define

$$\begin{aligned} |g|^- &= \min \{ \ |g(b)| \ \mid \ b \in \Sigma \ \} \ , \\ |g|^+ &= \max \{ \ |g(b)| \ \mid \ b \in \Sigma \ \} \ . \end{aligned}$$

For every positive integer (*deletion number*), the tag system defines a function from Σ^* to Σ^*; in what follows we only consider the function corresponding to the deletion number 2. The function, denoted $f_{\mathbf{T}}$, is defined by

(a) if $|w| \leq 1$ then $f_{\mathbf{T}}(w) = \lambda$; and

(b) if $w = b_0 b_1 \ldots b_n$, $n \geq 1$, then $f_{\mathbf{T}}(w) = b_2 \ldots b_n g(b_0)$.

The *sequence of* $\mathbf{T}$, denoted $\{s_n(\mathbf{T})\}_{n=0}^{\infty}$, is defined by $s_0(\mathbf{T}) = w_0$, and $s_{n+1}(\mathbf{T}) = f_{\mathbf{T}}(s_n(\mathbf{T}))$, $n \geq 0$.

Theorem A.1 *There is no algorithm to decide, for every tag system* $\mathbf{T} = (\Sigma, g, w_0)$ *with* $|g|^- = 1$ *and* $|g|^+ = 3$*, whether* $s_n(\mathbf{T}) = \lambda$ *for some* n*.*

Proof: See Theorem 5 in [Wan].

Theorem A.2 *There is no algorithm to decide, for every tag system* $\mathbf{T} = (\Sigma, g, w_0)$ *with* $|g|^- = 1$ *and* $|g|^+ = 3$*, whether* $|s_n(\mathbf{T})| \leq c$ *for some constant* c *and every* n*.*

Proof. If there were such an algorithm, we could construct an algorithm to decide the problem $s_n(\mathbf{T}) = \lambda$ of Theorem A.1 as follows: For a given $\mathbf{T}$, first decide whether $s_n(\mathbf{T}) \leq c$ for some c and all n. If this is not the case then $s_n(\mathbf{T}) \neq \lambda$ for all n. If, on the other hand, the sequence of $\mathbf{T}$ is bounded then generate the successive strings $s_n(\mathbf{T})$ until $s_{m_0}(\mathbf{T}) = s_{m_1}(\mathbf{T})$ for some m_0 and m_1, $m_0 \neq m_1$; now if $s_{m_0}(\mathbf{T}) = \lambda$ then the problem is decided, and if $s_{m_0}(\mathbf{T}) \neq \lambda$ then $s_n(\mathbf{T}) \neq \lambda$ for all n. $\square$

Theorem A.3 *There is no algorithm to decide, for every tag system* $\mathbf{T} = (\Sigma, g, w_0)$ *such that* $|g|^- = 1$*,* $|g|^+ = 3$ *and* $s_n(\mathbf{T}) \neq \lambda$ *for all* n*, whether* $|s_n(\mathbf{T})| \leq c$ *for some constant* c *and every* n*.*

Proof. For every tag system $\mathbf{T} = (\Sigma, g, w_0)$ choose a symbol $\# \notin \Sigma$ and define

$$\begin{aligned} \Sigma' &= \Sigma \cup \{\#\}\,, \\ w_0' &= w_0\#\#\,, \\ g'(b) &= g(b) \quad \text{for} \quad b \in \Sigma\,, \\ g'(\#) &= \#\#\,. \end{aligned}$$

Then the tag system $\mathbf{T}' = (\Sigma', g', w_0')$ is bounded (i.e. $|s_n(\mathbf{T}')| \leq c$ for some c and all n) if and only if $\mathbf{T}$ is. Moreover, $s_n(\mathbf{T}') \neq \lambda$ for all n, because every $s_n(\mathbf{T}')$ contains the subsequence $\#\#$.

Thus if we had an algorithm to decide the boundedness for every $\mathbf{T}' = (\Sigma', g', w_0')$ such that $|g|^- = 1$, $|g|^+ = 3$ and $s_n(\mathbf{T}') \neq \lambda$ for all n, then we would also have an algorithm to decide boundedness for every $\mathbf{T} = (\Sigma, g, w_0)$ such that $|g|^- = 1$ and $|g|^+ = 3$, in contradiction to A.2. $\square$

References

[Ber] J. Berstel: Transductions and context-free languages, B. G. Teubner Stuttgart (1979).

[Bir] M. Bird: The equivalence problem for deterministic two-tape automata, J. Comput. System Sci. 7 (1973) 218-236.

[Bo1] G. V. Bochmann and C. Sunshine: Formal methods in communication protocol design, I.E.E.E. Trans. Comm. COM-28 (1980), 624-631.

[Bo2] G. V. Bochmann: A general transition model for protocols and communication services, I.E.E.E. Trans. Comm. COM-28 (1980), 643-650.

[Bra] D. Brand and P. Zafiropulo: On communicating finite state machines, IBM RZ 1053 (1981).

[Eil] S. Eilenberg: Automata, languages and machines, Vol. A, Academic Press (1974).

[Gou] M. G. Gouda: Protocol machines – towards a logical theory of communication protocols, Univ. of Waterloo Ph. D. thesis (1977).

[Hop] J. E. Hopcroft and J. D. Ullman: Introduction to automata theory, languages and computation, Addison-Wesley (1979).

[Kan] R. Kannan and R. J. Lipton: The orbit problem is decidable, Proc. 12th Annual ACM Symp. on Theory of Computing (1980), 252-261.

[Luc] C. L. Lucchesi and D. H. Younger: A minimax theorem for directed graphs, J. London Math. Soc. (2) 17 (1978), 369-374.

[Man] Z. Manna and R. Waldinger: The logic of computer programming, IEEE Trans. on Software Engineering, SE-4 (1978), 199-229.

[May] E. Mayr: An algorithm for the general Petri net reachability problem, Proc. 13th Annual ACM Symp. on Theory of Computing (1981), 238-246.

[Mi1] M. Minsky: Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines, Ann. Math. 74 (1961), 437-455.

[Mi2] M. Minsky: Computation – finite and infinite machines, Prentice-Hall (1967).

[Pos] E. Post: Formal reduction of the combinatorial decision problems, Amer. J. Math. 65 (1943), 196-215.

[Rub] J. Rubin and C. H. West: An improved protocol validation technique, IBM RZ 1024 (1980).

[Wan] H. Wang: Tag systems and lag systems, Math. Annalen 152 (1963), 65-74.

[Zaf] P. Zafiropulo, C. H. West, H. Rudin, D. D. Cowan and D. Brand: Towards analyzing and synthesizing protocols, I.E.E.E. Trans. Comm. COM-28 (1980), 651-661.