Theoretical Computer Science 26 (1983) 83–103
North-Holland

NIVAT'S PROCESSING SYSTEMS: DECISION PROBLEMS RELATED TO PROTECTION AND SYNCHRONIZATION*

S. ISTRAIL and C. MASALAGIU
Department of Mathematics and Computer Centre, University 'Al. I. Cuza', Iaşi, Romania

Communicated by M. Nivat
Received March 1982
Revised April 1983

Abstract. The paper introduces a model for processing systems which provides 'environment' to the abstract notion of process as introduced by Nivat [13]. A basic component of the model is a protection mechanism which is general enough to capture as particular instances known protection strategies (e.g., take, grant, create, parameter passing) [5, 8, 9].

Decision problems associated with these systems are discussed for both cases: processes with infinite and finite behaviours. Solvability results are obtained for the safety problem: as a corollary we get the solvability result of Beauquier in the context of his processes [1]. Unsolvability results are also derived.

A concept of compatibility is introduced for processes acting in parallel subject to some synchronization condition. We show that the traversing from rational to algebraic systems can take the compatibility problem from solvable to unsolvable.

1. Introduction

The problem of controlling the access to information in large systems, manipulating many kinds of data, having different owners, is difficult and has various particular aspects. Every such system has some mechanism implementing security policies, being from this point of view a 'protection system' [15].

One of the most influencial model for protection systems is the model based on 'capabilities' [5, 15].

Starting with an idea of Beauquier [1] which considers a capability based protection system as the set of sequences of permitted 'actions' in the system, we give a general model for protection systems within the framework of the theory of processes, as developed by Nivat [13, 2, 17].

Our model provides 'environment' to the abstract notion of process, having finite or infinite behaviours, and being general enough to capture as particular instances the 'protocol' processes of Beauquier [1, 2] as well as other 'reasonable' protection mechanisms.

* A preliminary version was presented at the 8th Colloquium on Trees in Algebra and Programming, L'Aquila, Italy, March 9–11, 1983.

Decision problems associated with these systems are discussed for both cases: infinite and finite behaviours.

Solvability and unsolvability results are obtained for the safety problem and the compatibility problem. Our concept of compatibility is introduced for systems acting in parallel subject to some synchronization conditions.

Table 1 summarizes the decidability results obtained in this paper (S = solvable, U = unsolvable, ? = unknown).

Table 1
Decidability results.

System's class \ Decision problem	Safety				Compatibility			
	*	ω	*f	ωf	*	ω	*f	ωf
Rational	S	S	S	S	S	S	S	S
Algebraic	S	S	S	S	?	?	U	?
Context-sensitive	?	?	U	U	?	?	U	?
Recursive-enumerable	U	U	U	U	U	U	U	U

2. Preliminaries on infinite words and languages

Given a finite alphabet A and $N_+ = \{1, 2, \ldots\}$, a finite word over A can be thought of as being a partial function $f : N_+ \to A$ whose domain is

$$[n] = \{1, 2, \ldots, n\}.$$

For any $m \leq n$ we shall write

$$f[m] = f(1)f(2)\cdots f(m).$$

An infinite word over A will then be a total function $u : N_+ \to A$. We shall denote by $u[n]$ its initial segment of length n, i.e.,

$$u[n] = u(1)\cdots u(n).$$

$u[0]$ will always mean λ (the empty word).

Let A^ω be the set of infinite words over A and we put $A^\infty = A^\omega \cup A^*$.

The length of a finite word f will be denoted $|f|$.

The operator 'left factors' is defined by

$$\mathrm{FG}(f) = \{g \in A^* \mid g = f[n], n \leq |f|\} \quad \text{for } f \in A^*,$$

$$\mathrm{FG}(u) = \{u[n] \mid n \in N_+\} \quad \text{for } u \in A^\omega,$$

$$\mathrm{FG}(L) = \bigcup_{x \in L} \mathrm{FG}(x) \quad \text{for } L \subseteq A^\infty.$$

A basic notion in the theory of infinitary languages is that of adherence defined as follows:

$$\mathrm{Adh}(L) = \{u \in A^\omega \mid \mathrm{FG}(u) \subseteq \mathrm{FG}(L)\} \quad \text{for } L \subseteq A^\infty.$$

- $L \subseteq A^\infty$ is called *closed* if $\mathrm{Adh}(L) \subseteq L$.
- $L \subseteq A^\infty$ is called *rational* (*algebraic*) if $\mathrm{FG}(L)$ is regular (context-free).
- $L \subseteq A^\infty$ is called *central* if $L = \mathrm{FG}(\mathrm{Adh}(L))$.

Following Nivat [13] and interpreting A as the alphabet of 'actions', a *process p* is "a mechanism capable to do actions".

The set of infinite 'behaviours' of p is some fixed set $B^\omega(p) \subseteq A^\omega$ and the set of 'finite behaviours' is some fixed set $B^*(p) \subseteq A^*$. If $\boldsymbol{p} = \langle p_1, \ldots, p_k \rangle$ is a vector of processes, then $B^\omega(\boldsymbol{p}) = B^\omega(p_1) \times \cdots \times B^\omega(p_k)$ (respectively $B^*(\boldsymbol{p}) = B^*(p_1) \times \cdots \times B^*(p_k)$) and any $\boldsymbol{u} \in B^\omega(\boldsymbol{p})$ ($\boldsymbol{u} \in B^*(\boldsymbol{p})$) may be viewed as being an ω-word (word) over A^k (i.e., $A \times A \times \overset{k \text{ times}}{\cdots} \times A$.) ($B^*(\boldsymbol{p})$ is restricted to k-tuples of words of the same length).

We call a process rational (algebraic, closed) if its set of infinite behaviours is rational (algebraic, closed).

A set $S \subseteq A^k$ is called a 'condition of synchronization' or 'synchronization set'; given S, we can define the set of S-synchronized behaviours of $\boldsymbol{p}$ as $B_S^\omega(\boldsymbol{p}) \cup B_S^*(\boldsymbol{p})$ where

$$B_S^\omega(\boldsymbol{p}) = B^\omega(\boldsymbol{p}) \cap S^\omega \quad (\text{respectively, } B_S^*(\boldsymbol{p}) = B^*(\boldsymbol{p}) \cap S^*).$$

We shall denote by R, CF, CS the classes of regular, context-free, respectively context-sensitive languages of the Chomsky hierarchy.

The family of rational adherences and that of algebraic adherences possesses a representation of the form

$$L = \bigcup_{i=1}^{k} L_i \cdot (L_i')^\omega, \tag{1}$$

where L_i, L_i' are languages in the corresponding family.

Theorem 2.1 (Mc. Naughton). *Any rational adherence L can be represented as in* (1) *with* $L_i, L_i' \in R$, $1 \leq i \leq k$.

Theorem 2.2 (Nivat). *Any algebraic adherence L can be represented as in* (1) *with* $L_i, L_i' \in \mathrm{CF}$, $1 \leq i \leq k$.

More about infinitary languages and processes can be found in [12, 13, 17, 18].

We suppose the reader familiar with basic facts of formal language theory [4, 14, 3].

3. A model of protection in processing systems

A basic question concerning computing mechanisms is the following.

How can a 'mechanism' be programmed to supervise the development of a 'process' (i.e., a set of 'actions' performed sequential or in parallel) in such a way that the security requirements do not be violated? Much more, in what conditions can we tell something a priori on the possibility of executing 'illegal' actions?

Let us introduce the basic concept of the paper.

Definition 3.1. *A processing system* is a 6-tuple $\mathrm{PS} = (\mathrm{Act}, Q, F, q_0, \mathcal{H})$ where $\mathcal{A} = (\mathrm{Act}, Q, F, q_0)$ is a finite automaton (without final states) and $\mathcal{H} = (\{Q_a\}_{a \in \mathrm{Act}}, H)$ is the 'capability legislation' and

(i) Act is the set of *actions*,

(ii) Q is the set of *states*,

(iii) $F : \mathrm{Act} \times Q \to 2^Q$ is the *transition function*,

(iv) $q_0 \in Q$ is the *initial state*,

(v) $Q_a \subseteq Q$ is the set of *states compatible with action a* for any $a \in \mathrm{Act}$,

(vi) H, the set of *histories*, is a prefix-closed subset of Act^*.

Two languages, related to a processing system are introduced:

- The set of *legal infinite behaviours* is given by

$$L_\omega(\mathrm{PS}) = \{u \in \mathrm{Act}^\omega \mid \forall i \geqslant 0: F(u[i], q_0) \cap Q_{u(i+1)} \neq \emptyset \text{ and } u[i] \in H\}.$$

- The set of *legal finite behaviours* is given by

$$L_*(\mathrm{PS}) = \{u \in \mathrm{Act}^* \mid (\forall i, 0 \leqslant i < |u| : F(u[i], q_0) \cap Q_{u(i+1)} \neq \emptyset) \text{ and } u \in H\}.$$

Remark. Because H is prefix-closed, in the definition of $L_*(\mathrm{PS})$ we have: $u \in H$, iff $\forall j, 0 \leqslant j < |u| : u[j] \in H$.

In fact, a processing system PS may be viewed as an action-sensitive construct. Every action changes the state of the system in the same way an input symbol changes the state of a finite automaton. In addition, the capability legislation implements a 'protection mechanism': denoting by Q_a the subset of Q consisting of those states in which action 'a' can occur, we require for a process to be composed from actions having a 'good' state as well as a 'good' history.

So, the process described by the sequences of actions, is protected from 'illegal' occurrences of actions which do not agree with compatible states or with admissible histories.

We shall associate two processes to a processing system:

- p_{PS} the *full process* with infinite behaviours $B_\omega(p_{\mathrm{PS}}) = L_\omega(\mathrm{PS})$ and finite behaviours $B_*(p_{\mathrm{PS}}) = L_*(\mathrm{PS})$.
- p'_{PS} the *historyless process*, having as infinite behaviours

$$B_\omega(p'_{\mathrm{PS}}) = \{u \in \mathrm{Act}^\omega \mid \forall i \geqslant 0, F(u[i], q_0) \cap Q_{u(i+1)} \neq \emptyset\}$$

and as finite behaviours

$$B_*(p'_{PS}) = \{u \in \mathrm{Act}^* \mid \forall i,\, 0 \leq i < |u| : F(u[i], q_0) \cap Q_{u(i+1)} \neq \emptyset\}$$

(i.e., we have dropped the restriction on histories).

Our model incorporates features presented in other protection models (Beauquier [1], Lipton and Snyder [9] and Harrison, Ruzzo and Ulmann [5]), providing us with a general framework to represent protection mechanisms and having decision procedures for enough complex classes.

As will be clear from Example 3.2, our mechanism is powerful enough to express protection strategies in the same way, as graph-rewriting rules do [8, 2] (e.g., take, grant, call, create, segment). Thus we can model sufficiently realistic systems.

Example 3.2. The states of our processing system PS will be graphs with vertices C, a set of objects/subjects (e.g., Editor, File, User 1) and edges E, labeled by names of actions (e.g., call, read, write-abbreviated c, r, w).

We interprete an action as being a triple, consisting of two labeled vertices together with a labeled edge joining them (see, e.g., Fig. 1).

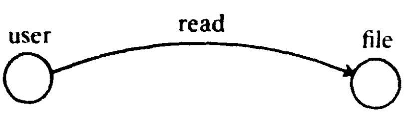

Fig. 1. The action 'a'.

If 'a' denotes this action, Q_a, the set of states in which 'a' can occur, will be the set of all graphs with vertices from C and edges from E which contain as subgraph the one which represents 'a' (as in Fig. 1).

Now, if states means graphs, state-transitions will be graph transformations by a set of graph-rewriting rules $\mathcal{R}$. We shall exemplify with GRANT: for $s_1, s_2, s_3 \in C$, and α the action (s_1, α, s_3), we consider the graph rewriting rule depicted in Fig. 2.

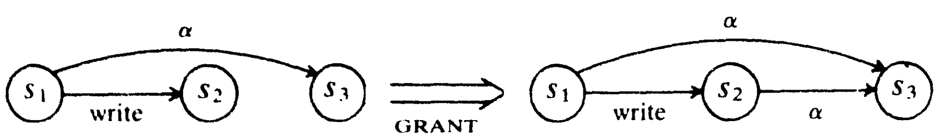

Fig. 2.

That is, "if s_1 can write on s_2 and it happens that it can do α to s_3, then s_1 grants s_2 the ability (the right) to do α to s_3". If one wants to capture a GRANT-mechanism our set $\mathcal{R}$ of rewriting rules will contain GRANT rules for any vertices and edges fulfilling the 'left-member' requirement; also we shall include an 'identity rule' (Fig. 3) for any edge.

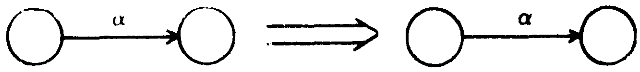

Fig. 3.

We shall consider five objects/subjects s_1, s_2, s_3, s_4, s_5 and the initial state q_0 will be given by Fig. 4.

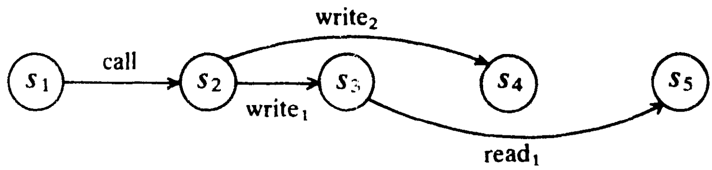

Fig. 4. The initial state q_0.

In our formalism we shall define the transition function by

$$F(a, q) = \{q' \mid q \underset{\text{rule}}{\Longrightarrow} q', \text{rule} \in \mathscr{R}\}.$$

H, the set of histories, will be constructed as follows. We want any write-action to be immediately preceded by a call-action, i.e., if $v \in H$, $v = v_1$ **write**$_i$ v_2, then $v = v_1'$ call **write**$_i$ v_2 (in abbreviated form $v = v_1' c w_i v_2$).

Hence we shall take $H = \mathrm{FG}((\{c, r_1, r_2\}^* \cdot (c\{w_1, w_2, w_3\})^* \cdot \{c, r_1, r_2\}^*)^*)$.

Fig. 5 presents (non-identical) transitions of states as well as sequences of legal actions in PS. We shall focus our attention on $L_\omega(\mathrm{PS})$.

Considering the historyless process of PS, namely p'_{PS} we have

$$B_\omega(p'_{\mathrm{PS}}) = \{c, w_2, r_1\}^\omega \cup \{c, w_2, r_1\}^* \cdot w_1 \cdot \{c, w_1, w_2, r_1\}^\omega$$
$$\cup \{c, w_2, r_1\}^* \cdot w_1 \cdot \{c, w_1, w_2, r_1\}^* \cdot w_3 \cdot \{c, w_1, w_2, w_3, r_1, r_2\}^\omega,$$

which is a rational process.

For the other process p_{PS} (of legal behaviours), $B_\omega(p_{\mathrm{PS}})$ will be obtained as given by Fig. 5.

Notation. If V is an alphabet containing $c, w_1, \ldots, w_t$ and $V' = V \setminus \{w_1, \ldots, w_t\}$, then $[V]$ will denote the set

$$[V] = \left(V'^* \bigcup_{i=1}^{t} (cw_i)^* V'^* \right)^*.$$

Now we have

$$B_\omega(p_{\mathrm{PS}}) = [c, w_2, r_1]^\omega \cup [c, w_2, r_1] \cdot cw_1 \cdot [c, w_1, w_2, r_1]^\omega$$
$$\cup [c, w_2, r_1] \cdot cw_1 \cdot [c, w_1, w_2, r_1] \cdot cw_3 \cdot [c, w_1, w_2, w_3, r_1, r_2]^\omega.$$

Hence p_{PS} is a rational process.

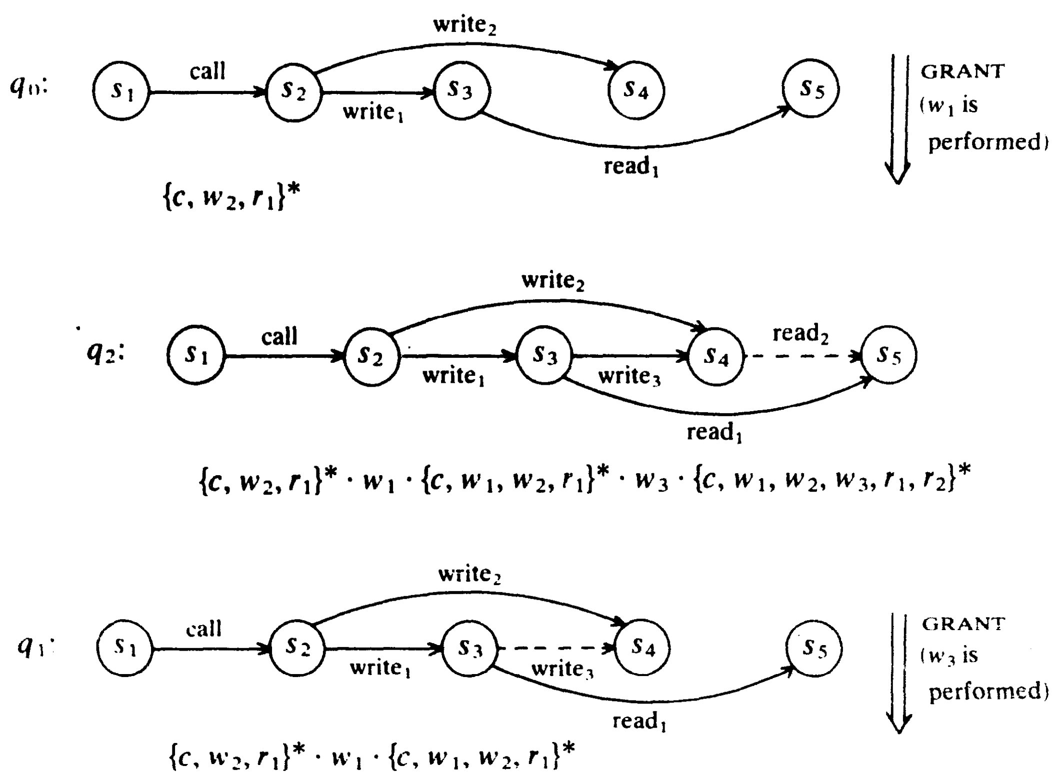

Fig. 5.

It is not too difficult to see that taking another history set H we can obtain p_{PS} algebraic.

Indeed, it is sufficient to ask that read—and write—actions to be (not necessarily immediately) preceded by a call-action.

Proposition 3.3. *For any processing system its historyless process is rational. I.e., the set* $R_{PS} = B_*(p'_{PS})$ *is regular.*

Proof. Let us consider the regular grammar $G = (V_N, V_T, x_0, P)$ where

(i) $V_N = \{x_0\} \cup \{(a, q_a) \mid a \in \text{Act}, q_a \in Q_a\}$,

(ii) $V_T = \{\bar{y} \mid y \in V_N \setminus \{x_0\}\}$,

(iii) P consists of the following rules $((a, q) \in V_N)$:

- $x_0 \to \overline{(a, q_0)}$ if $a \in$ Act and $q_0 \in Q_a$,
- $(a, q) \to \overline{(a, q)}(a', q')$ for all q', a', with $q' \in F(a, q) \cap Q_{a'}$,
- $(a, q) \to \lambda$.

It is easy to observe that

$$B_*(p'_{PS}) = h(L(G)),$$

where h is a homomorphism defined by

$$h(\overline{(a, q)}) = a \quad \text{for all } a, q. \qquad \square$$

Remarks. (1) $L_*(\mathrm{PS}) = R_{\mathrm{PS}} \cap H$.

(2) $\lambda \in L_*(\mathrm{PS})$.

Proposition 3.4. *Let* $\mathrm{PS} = (\mathrm{Act}, Q, F, q_0, \mathcal{H})$ *be a processing system where* $\mathcal{H} = (\{Q_a\}_{a \in \mathrm{Act}}, H)$. *Then*

$$\mathrm{Adh}(H) \cap \mathrm{Adh}(R_{\mathrm{PS}}) = \mathrm{Adh}(H \cap R_{\mathrm{PS}}) = L_\omega(\mathrm{PS}). \tag{2}$$

Proof. (a) $\mathrm{Adh}(H) \cap \mathrm{Adh}(R_{\mathrm{PS}}) = \mathrm{Adh}(H \cap R_{\mathrm{PS}})$.

We always have: $\mathrm{Adh}(H \cap R_{\mathrm{PS}}) \subseteq \mathrm{Adh}(H) \cap \mathrm{Adh}(R_{\mathrm{PS}})$. The converse inclusion follows from the fact that H and R_{PS} are prefix-closed.

(b) $L_\omega(\mathrm{PS}) = \mathrm{Adh}(H \cap R_{\mathrm{PS}})$.

Let $u \in L_\omega(\mathrm{PS})$. $\forall i$, $u[i] \in R_{\mathrm{PS}} \cap H = \mathrm{FG}(R_{\mathrm{PS}} \cap H)$ and hence $\mathrm{FG}(u) \subseteq \mathrm{FG}(R_{\mathrm{PS}} \cap H)$ yielding $u \in \mathrm{Adh}(R_{\mathrm{PS}} \cap H)$. Conversely, let u belong to $\mathrm{Adh}(R_{\mathrm{PS}} \cap H)$. We have $\mathrm{FG}(u) \subseteq \mathrm{FG}(R_{\mathrm{PS}} \cap H) = R_{\mathrm{PS}} \cap H$ which shows that $\forall i$, $u[i] \in R_{\mathrm{PS}} \cap H$, i.e., $u \in L_\omega(\mathrm{PS})$. $\square$

Remark. From this proposition we have

$$\mathrm{Adh}(L_*(\mathrm{PS})) = L_\omega(\mathrm{PS}) \tag{3}$$

which seems a very natural link between the two kinds of behaviours as already emphasized by Nivat [13], hence $L_\omega(\mathrm{PS})$ is a closed language.

4. Decision problems

This section is devoted to the study of the 'safety problem' and the 'compatibility problem' for processing systems. Decidability results are obtained for both problems.

Definition 4.1. A processing system is called *central* if $H \cap R_{\mathrm{PS}}$ is a central language. It is called *rational* (*algebraic*, *context-sensitive*) if H is regular (context-free, context-sensitive).

Remark. The 'centrality hypothesis' means in fact that

$$\mathrm{FG}(L_\omega(\mathrm{PS})) = L_*(\mathrm{PS}). \tag{4}$$

The safety problem

To get insight why we call the problem 'safety problem' instead of, say, 'occurrence problem' or 'alphabet problem' let us consider the environment provided by Example 3.2. There, an action means that a subject/object does something to another subject/object. So, if for example 'a' will be interpreted as 's reads s'', then the safety problem asks if, in some legal behaviour of PS, s will eventually can read s' at some moment.

In this interpretation we obtain the well-known concept of 'safety' as used in [8, 2]. Usually the question of safety will stand for illegal actions to see whether it happens to be a behaviour containing them.

The safety problem for infinite behaviours (ω-safety problem). Given a processing system $\mathrm{PS} = (\mathrm{Act}, Q, F, q_0, \mathcal{H})$ and an action $a \in \mathrm{Act}$, does there exist a behaviour $u \in L_\omega(\mathrm{PS})$ such that $u = wau'$ for some $w \in \mathrm{Act}^*$ and $u' \in \mathrm{Act}^\omega$?

If for a given PS and 'a' the answer to the ω-safety problem is "no", we shall say that "PS is *ω-safe for a*".

We shall denote by

$$\mathrm{SAFETY}^{\omega}_{\mathrm{PS},a} = \mathrm{FG}(L_\omega(\mathrm{PS})) \cap \mathrm{Act}^* \cdot a. \tag{5}$$

Let us remark that PS is ω-safe for 'a' if and only if $\mathrm{SAFETY}^{\omega}_{\mathrm{PS},a} = \emptyset$.

The safety problem for finite behaviours ($*$-safety problem). Given a processing system $\mathrm{PS} = (\mathrm{Act}, Q, F, q_0, \mathcal{H})$ and action $a \in \mathrm{Act}$, does there exist a behaviour $w \in L_*(\mathrm{PS})$ such that $w = w'aw''$ for some $w', w'' \in \mathrm{Act}^*$?

This time we have to deal with the set $X = H \cap R_{\mathrm{PS}} \cap \mathrm{Act}^* \cdot a \cdot \mathrm{Act}^*$, but because $X = \emptyset$ iff $H \cap R_{\mathrm{PS}} \cap \mathrm{Act}^* \cdot a = \emptyset$, we shall denote by

$$\mathrm{SAFETY}^{*}_{\mathrm{PS},a} = H \cap R_{\mathrm{PS}} \cap \mathrm{Act}^* \cdot a. \tag{6}$$

In an analogous way as for the ω-safety problem, we see that PS is $*$-safe for 'a' iff $\mathrm{SAFETY}^{*}_{\mathrm{PS},a} = \emptyset$.

Theorem 4.2. *Let* PS *be a central processing system and* $a \in \mathrm{Act}$. *Then* PS *is ω-safe for a iff* PS *is $*$-safe for a.*

Proof. Because of (4), (5) and (6) we have

$$\mathrm{SAFETY}^{\omega}_{\mathrm{PS},a} = \mathrm{SAFETY}^{*}_{\mathrm{PS},a}. \quad \square \tag{7}$$

Theorem 4.3. *The $*$-safety problem for algebraic processing systems is solvable.*

Proof. The theorem follows from (6) and the fact that H is context-free. $\square$

Corollary 4.4. *The ω-safety problem for central algebraic processing systems is solvable.*

Remark. For non-central Nivat-processing systems PS we have in general

$$\mathrm{SAFETY}^{\omega}_{\mathrm{PS},a} \subsetneq \mathrm{SAFETY}^{*}_{\mathrm{PS},a} \quad \forall a$$

and it is difficult to treat $\mathrm{FG}(L_\omega(\mathrm{PS}))$.

However we can prove the following theorem.

Theorem 4.5. *The ω-safety problem for algebraic processing systems is solvable.*

Proof. By Proposition 3.4 we have

$$L_\omega(\mathrm{PS}) = \mathrm{Adh}(L_*(\mathrm{PS})),$$

that is, $L_\omega(\mathrm{PS})$ equals the adherence of a context-free language and hence it has a Nivat representation

$$L_\omega(\mathrm{PS}) = \bigcup_{i=1}^{p} L_i \cdot (L_i')^\omega$$

where L_i and L_i', $i = 1, \ldots, p$ are context-free languages (see Theorem 2.2). Then

$$\mathrm{FG}(L_\omega(\mathrm{PS})) = \bigcup_{i=1}^{p} [\mathrm{FG}(L_i) \cup L_i(L_i')^* \cdot \mathrm{FG}(L_i')]$$

is a context-free language. So our result follows because the emptiness problem for context-free languages is decidable. □

We shall discuss some extensions of the problem in Section 5.

Synchronization and compatibility

We shall consider the compatibility problem for processing systems, showing that in the rational case the problem is decidable. Our notion of compatibility is more general than that of Beauquier (viewed only as inclusion between the sets of behaviours)—see Section 5.

Our concept is formulated in the context of parallelism and synchronization. Within this framework we feel that the notion captures an interesting phenomenon.

Let us suppose that we have two processing systems with a common Act set:

$$\mathrm{PS}_i = (\mathrm{Act}, Q_i, F_i, q_{0i}, \mathcal{H}_i), \quad i = 1, 2$$

and a 'synchronization' set $S \subseteq \mathrm{Act} \times \mathrm{Act}$.

The ω-compatibility problem of PS_1 with PS_2. For every infinite behaviour of PS_1 does there exist an infinite behaviour of PS_2, such that the two processes p_{PS_1}, p_{PS_2} can 'cooperate' (i.e., run in parallel and satisfy the synchronization conditions) during these behaviours?

If the answer is "yes", we shall say that "PS_1 is ω-compatible with PS_2".

Notation

$$L_S^\omega(PS_1, PS_2) = \{v \in S^\omega \mid v = (u_{11}, u_{21})(u_{12}, u_{22}) \cdots (u_{1n}, u_{2n}) \cdots,$$
$$u_i = u_{i1}u_{i2} \cdots u_{in} \cdots \in L_\omega(PS_i), i = 1, 2\}.$$

The infinite word v above will be denoted by

$$\langle u_1, u_2\rangle.$$

Also, if w and z are finite words of the same length $w = w_1 \cdots w_n$, $z = z_1 \cdots z_n$, then $\langle w, z\rangle$ will denote the word

$$(w_1, z_1)(w_2, z_2) \cdots (w_n, z_n).$$

Thus

$$L_S^\omega(PS_1, PS_2) = \{v \in S^\omega \mid v = \langle u_1, u_2\rangle, u_i \in L_\omega(PS_i), i = 1, 2\}.$$

We shall define here the 'finite'-analogous for $L_S^\omega(PS_1, PS_2)$. Namely, we put

$$L_S^*(PS_1, PS_2) = \{w \in S^\omega \mid w = (u_{11}, u_{21}) \cdots (u_{1n}, u_{2n}), u_i = u_{i1} \cdots u_{in}$$
$$\in L_*(PS_1), i = 1, 2\}$$
$$= \{\langle u_1, u_2\rangle \mid u_i \in L_*(PS_i), i = 1, 2, |u_1| = |u_2|\}.$$

We can extend our notation to languages in the following way:

$$\langle L_1, L_2\rangle = \{\langle u_1, u_2\rangle \mid u_1 \in L_1, u_2 \in L_2\} \qquad \text{for } L_1, L_2 \subseteq A^\omega,$$
$$\langle L_1, L_2\rangle = \{\langle u_1, u_2\rangle \mid u_1 \in L_1, u_2 \in L_2, |u_1| = |u_2|\} \qquad \text{for } L_1, L_2 \subseteq A^*.$$

Now we can reformulate the ω-compatibility problem:

$$\forall u_1 \in L_\omega(PS_1), \exists u_2 \in L_\omega(PS_2): \quad \langle u_1, u_2\rangle \in L_S^\omega(PS_1, PS_2)\ ?.$$

Defining the homomorphism π_1 (first projection) by $\pi_1((a, b)) = a$ for any $(a, b) \in S$, we can extend it to infinite words as follows: if $v \in S^\omega$, we define $\pi_1(v)$ as the limit of the sequence $\pi_1(v[n])_{n \geq 1}$, i.e., the unique u' such that, for any $n \geq 1$, $\pi_1(v[n])$ is a prefix of it.

Reformulating again, the ω-compatibility problem becomes

$$\pi_1(L_S^\omega(PS_1, PS_2)) = L_\omega(PS_1)\ ?.$$

Proposition 4.6. *Given two processing systems* PS_1, PS_2 and a synchronization set S, *there exists a processing system* PS *such that*

$$L_\omega(\mathrm{PS}) = L_S^\omega(\mathrm{PS}_1, \mathrm{PS}_2), \qquad L_*(\mathrm{PS}) = L_S^*(\mathrm{PS}_1, \mathrm{PS}_2).$$

Proof. We take

$$\mathrm{PS} = (\mathrm{Act} \times \mathrm{Act}, Q_1 \times Q_2, F, (q_{01}, q_{02}), \mathscr{H})$$

where F is given by

$$F((a_1, a_2), (q_1, q_2)) = \{(h_1, h_2) \mid h_i \in F_i(a_i, q_i), i = 1, 2\}$$

and

$$\mathscr{H} = (\{Q_{(a_1,a_2)}\}_{(a_1,a_2) \in \mathrm{Act}^2}, H)$$

with

$$Q_{(a_1,a_2)} = \begin{cases} Q_{a_1} \times Q_{a_2}, & (a_1, a_2) \in S, \\ \emptyset, & (a_1, a_2) \notin S, \end{cases}$$

$$H = \{\langle w_1, w_2\rangle \mid w_i \in H_i, i = 1, 2, |w_1| = |w_2|\}.$$

Let us prove that $L_\omega(\mathrm{PS}) = L_S^\omega(\mathrm{PS}_1, \mathrm{PS}_2)$.

For $v \in L_\omega(\mathrm{PS})$, $v = \langle u_1, u_2\rangle$ the following two conditions are satisfied:

(a) $\forall j, F(v[j], (q_{01}, q_{02})) \cap Q_{v(j+1)} \neq \emptyset$.

(b) $\forall j, v[j] \in H$.

We can refine them by

(a') $\quad \forall j, \quad \{(h_1, h_2) \mid h_i \in F_i(u_i[j], q_{0i}), i = 1, 2\} \cap$
$$\cap\, Q_{(u_1(j+1), u_2(j+1))} \neq \emptyset.$$

(b') $\quad \forall j, \quad u_i[j] \in H_i, i = 1, 2.$

But $Q_{(u_1(j), u_2(j))} \neq \emptyset$ iff

(a'') $\quad (u_1(j), u_2(j)) \in S$.

From (a') it follows that for every j

(a''') $\quad F_i(q_{0i}, u_i[j]) \cap Q_{u_i(j+1)} \neq \emptyset, \quad i = 1, 2.$

As a conclusion, $v \in L_\omega(\mathrm{PS})$ iff (a''), (a''') and (b') hold which in turn holds iff $v \in L_S^\omega(\mathrm{PS}_1, \mathrm{PS}_2)$.

The $*$-case is obtained in an analogous way. $\square$

Remark. From Proposition 4.6 it follows that

$$\pi_1(L_S^\omega(\mathrm{PS}_1, \mathrm{PS}_2)) = \pi_1(L_\omega(\mathrm{PS})).$$

So our ω-compatibility problem will be

$$L_\omega(\mathrm{PS}_1) = \pi_1(L_\omega(\mathrm{PS}))\ ?. \tag{$\bigstar$}$$

Because we always have

$$\pi_1(L_\omega(\mathrm{PS})) \subseteq L_\omega(\mathrm{PS}_1),$$

it follows that ($\bigstar$) is equivalent with

$$L_\omega(\mathrm{PS}_1) \subseteq \pi_1(L_\omega(\mathrm{PS}))\ ?\ .$$

In order to obtain our decidability result we need a theorem of Nivat [13] which we present for the case of two processing systems.

Theorem 4.7 (Nivat [13]). *If p_1, p_2 are two closed rational processes and S is a synchronization set, then $B_S^\omega(p_1, p_2)$ is a closed rational language.*

Theorem 4.8. *Let* PS_1, PS_2 *be two rational processing systems, and S a synchronization set.*

Then the ω-compatibility problem of PS_1 *with* PS_2 *is decidable.*

Proof. As we have noted before, $L_\omega(\mathrm{PS}_i)$, $i = 1, 2$, are closed sets. Because they are rational adherences, they possess a McNaughton representation (see Theorem 2.1) yielding that $\mathrm{FG}(L_\omega(\mathrm{PS}_i)) \in R$, that is, $L_\omega(\mathrm{PS}_i)$ are rational. From the Nivat Theorem 4.7, $L_S^\omega(\mathrm{PS}_1, \mathrm{PS}_2) = B_S^\omega(p_{\mathrm{PS}_1}, p_{\mathrm{PS}_2})$ is also a closed rational set. Moreover, the equality shows that $B_S^\omega(p_{\mathrm{PS}_1}, p_{\mathrm{PS}_2}) = L_\omega(\mathrm{PS})$ is an adherence, namely $\mathrm{Adh}(L_*(\mathrm{PS}))$. Because π_1 is a faithful sequential mapping, π_1 commutes with Adh (property 9 in [16]) and hence $\pi_1(L_\omega(\mathrm{PS}))$ is closed.

As both members are closed sets, their inclusion $L_\omega(\mathrm{PS}_1) \subseteq \pi_1(L_\omega(\mathrm{PS}))$ is equivalent with $\mathrm{FG}(L_\omega(\mathrm{PS}_1)) \subseteq \mathrm{FG}(\pi_1(L_\omega(\mathrm{PS})))$. Now, the theorem follows because the inclusion in R is decidable. □

The $*$-compatibility problem of PS_1 with PS_2. For every finite behaviour of PS_1 does there exist a finite behaviour of PS_2 such that the two processes p_{PS_1} and p_{PS_2} can 'cooperate' during these behaviours?

If the answer is "yes", we shall say that "PS_1 is $*$-compatible with PS_2".

The problem can be rephrased as

$$\pi_1(L_S^*(\mathrm{PS}_1, \mathrm{PS}_2)) = L_*(\mathrm{PS}_1)?$$

in the same way we considered it in the ω-case.

With PS given by Proposition 4.6, the $*$-compatibility is captured in the question:

$$L_*(\mathrm{PS}_1) \subseteq \pi_1(L_*(\mathrm{PS}))?. \tag{8}$$

Theorem 4.9. *Let* PS_1 *and* PS_2 *be any central processing systems and* PS *given by Proposition* 4.6. *Then* $L_*(\mathrm{PS}_1) \subseteq \pi_1(L_*(\mathrm{PS}))$ *iff* $\mathrm{FG}(L_\omega \mathrm{PS}_1)) \subseteq \pi_1(\mathrm{FG}(L_\omega(\mathrm{PS})))$.

Proof. ($\Rightarrow$) $L_*(\mathrm{PS}_1) \subseteq \pi_1(L_*(\mathrm{PS}))$ implies

$$\mathrm{Adh}(L_*(\mathrm{PS}_1)) \subseteq \mathrm{Adh}(\pi_1(L_*(\mathrm{PS}))).$$

Because PS is closed, again by Property 9 in [16] and by Proposition 3.4 we have

$$\mathrm{Adh}(L_*(\mathrm{PS}_1)) \subseteq \pi_1(\mathrm{Adh}(L_*(\mathrm{PS}))) \quad \text{and} \quad L_\omega(\mathrm{PS}_1) \subseteq \pi_1(L_\omega(\mathrm{PS})).$$

Hence $\mathrm{FG}(L_\omega(\mathrm{PS}_1)) \subseteq \pi_1(\mathrm{FG}(L_\omega(\mathrm{PS})))$.

($\Leftarrow$) $\mathrm{FG}(L_\omega(\mathrm{PS}_1)) \subseteq \pi_1(\mathrm{FG}(L_\omega(\mathrm{PS})))$ yields

$$\mathrm{FG}(\mathrm{Adh}(L_*(\mathrm{PS}_1))) \subseteq \pi_1(\mathrm{FG}(\mathrm{Adh}(L_*(\mathrm{PS})))).$$

PS_1 being central it follows that

$$L_*(\mathrm{PS}_1) \subseteq \pi_1(\mathrm{FG}(\mathrm{Adh}(L_*(\mathrm{PS}))))$$
$$\subseteq \pi_1(\mathrm{FG}(L_*(\mathrm{PS}))) = \pi_1(L_*(\mathrm{PS})). \qquad \square$$

Corollary 4.10. *The* $*$*-compatibility problem for central, rational processing systems is decidable.*

However, the result holds in a more general case.

Theorem 4.11. *Let* PS_1 *and* PS_2 *be two rational processing systems. Then the* $*$*-compatibility problem of* PS_1 *with* PS_2 *is solvable.*

Proof. We have $L_*(\mathrm{PS}_i) = H_i \cap R_{\mathrm{PS}_i}$, $i = 1, 2$, which implies that $L_*(\mathrm{PS}_i)$ is regular when H_i is regular ($i = 1, 2$). What remains to prove is that H, the history set of PS is regular. (PS is given by Proposition 4.6.)

If S is the synchronization set, then

$$H = \{v \in S^* \mid v = \langle u_1, u_2 \rangle, |u_1| = |u_2|, u_i \in H_i, i = 1, 2\}$$

and we know that H_1 and H_2 are regular. Let G_1, G_2 be two regular grammars in normal form, i.e., having rules $x \to ay$, $x \to b$ and respectively $x' \to a'y'$, $x' \to b'$ (where

x, y, x', y' are nonterminals and $a, b, a', b' \in$ Act, that is, terminals). Following a standard construction it is not difficult to see that a grammar with rules $(x, x') \to (a, a')(y, a'), (x, x') \to (b, b')$ will generate exactly $H \setminus \{\lambda\}$.

Now (8) is decidable, being an inclusion between regular sets. $\square$

5. Finally legal behaviours and Beauquier processes

In Beauquier [1, 2] a protection system is identified with the set of all finite sequences of actions permitted in it.

Within this framework he obtained important solvability results for the safety problem (formulated in similar terms) and the compatibility problem viewed only as inclusion between the sets of behaviours, i.e., not in the parallelism and synchronization context.

His model contains some protection mechanisms expressed in somewhat informal terms and this implies that any generalization requires new proofs of the solvability results, if at all possible.

Considering 'finally legal behaviours' in our Nivat's processing systems we obtain that Beauquier's processes are particular instances of ours.

In this general framework we re-obtain Beauquier's result of the solvability of the safety problem, now for algebraic systems. For the compatibility problem we show that his result is the best one (to date) because the natural extension from Dyck sets to context-free sets takes the problem for solvable to unsolvable. Note that our concept of compatibility subsumes the Beauquier's one by simply taking $S = \{(a, a) \mid a \in \text{Act}\}$.

Let us consider the context provided by Example 3.2 and imagine that all actions are only of two types: 'ask for' ('[') and 'satisfy' (']'). An action of the form

$$s \xrightarrow{m} 0$$

has in Beauquier systems two 'parts':

$\langle 0, m, [\rangle$ (read: "someone asks for permission to do m to 0")

and

$\langle 0, m,]\rangle$ (read: "it is permitted to do m to 0").

Because the subject s who asks for permission to do m to 0 (and then getting this permission) must be uniquely determined, a behaviour (i.e., a finite sequence of this protocolar version of actions, named *events* by Beauquier) is required to be a restrained Dyck word over the alphabet of parentheses:

$$\text{Act} = \{\langle 0, m, \$\rangle \mid 0 \in \mathcal{O}, m \in M, \$ \in \{[,]\}\}$$

where $\mathcal{O}$ is the object/subject set and M the set of modalities.

In this way his events are actions; the access matrix will be a graph and the transformations in the access matrix can be modeled by graph rewriting rules.

All these features can be captured in our processing systems, except the fact that the history set H need not be prefix-closed.

Definition 5.1. A *generalized processing system* (GPS) is a processing system for which we drop the restriction on H to be prefix-closed, i.e., H is an arbitrary set. The languages

$$L_*(\text{GPS}) = \{u \in \text{Act}^* \mid (\forall i, 0 \leq i < |u|: \\ F(u[i], q_0) \cap Q_{u(i+1)} \neq \emptyset), \text{ and } u \in H\} \tag{9}$$

and

$$L_\omega(\text{GPS}) = \text{Adh}(L_*(\text{PS})) \tag{10}$$

will be referred to respectively as the set of *finally legal finite behaviours* (of P_{GPS}) and the set of *finally legal infinite behaviours* (of P_{GPS}). The languages studied by Beauquier in connection with his system are of the form $L_*(\text{GPS})$. We consider also infinite behaviours in our systems, and the results obtained turn out to provide new information about the Beauquier's systems as well.

The safety and the compatibility problem for finally legal behaviours can be formulated in analogous terms. They will be referred to as the $*f/\omega f$-safety/compatibility problems.

Theorem 5.2. *The $*f$-safety problem for (finally legal finite behaviours of) algebraic generalized processing systems is decidable.*

Proof. It is immediate from the fact that $L_{*f}(\text{GPS}) = H \cap R_{\text{PS}}$ and the analogous $\text{SAFETY}_{\text{GPS},a}$ again equals $H \cap R_{\text{GPS}} \cap \text{Act}^* \cdot a$, and H is context-free. □

Theorem 5.3. *The ωf-safety problem for (finally legal infinite behaviours of) algebraic generalized processing systems is decidable.*

Proof. From (10), following the same way as in the proof of Theorem 4.5. □

On the context-sensitive level the safety problem is unsolvable. Actually we can prove a stronger result.

Theorem 5.4. *The *f-safety problem for central context-sensitive generalized processing systems is unsolvable.*

Proof. We shall prove first that for an arbitrary GPS there exists a central GPS' such that

$$\forall a, \quad \text{SAFETY}^{*}_{\text{GPS},a} = \emptyset \Leftrightarrow \text{SAFETY}^{\omega}_{\text{GPS}',a} = \emptyset. \tag{11}$$

Let GPS $= (\text{Act}, Q, F, q_0, \mathcal{H})$ be a generalized processing system and $a' \notin \text{Act}$. Then GPS' $= (\text{Act}', Q', F', q_0', \mathcal{H}')$ where

- $\text{Act}' = \text{Act} \cup \{a'\}$,
- $Q' = Q \cup \{q'\}$, $q' \notin Q$,
- F' is defined as

$$F'(a', q') = \{q'\},$$

$$F'(a', q) = \emptyset \qquad \text{for } q \neq q'$$

$$F'(a, q) = F(a, q) \cup \{q'\} \quad \text{for } a \neq a' \text{ and } q \in Q,$$

$$F'(a, q') = \emptyset \qquad \text{for } a \neq a',$$

- $q_0' = q_0$,
- $\mathcal{H}' = (\{Q_a'\}_{a \in \text{Act}'}, H')$. Here $H' = \text{FG}(H) \cdot (a')^*$ and

$$Q_a' = Q_a \quad \text{for } a \neq a', \qquad Q_{a'}' = \{q'\}.$$

Let us note that

$$L_*(\text{GPS}') = \text{FG}(L_*(\text{GPS})) \cdot (a')^* \quad \text{and} \quad R_{\text{GPS}'} = R_{\text{GPS}} \cdot (a')^*.$$

We have

$$\begin{aligned}
&\text{FG}(\text{Adh}(L_*(\text{GPS}'))) = \\
&\quad = \text{FG}(\text{Adh}(\text{FG}(L_*(\text{GPS})) \cdot (a')^*)) \\
&\quad = \text{FG}(\text{Adh}(\text{FG}(L_*(\text{GPS}))) \cup \text{FG}(L_*(\text{GPS})) \cdot (a')^{\omega}) \\
&\quad = \text{FG}(\text{Adh}(L_*(\text{GPS}))) \cup \text{FG}(L_*(\text{GPS})) \cdot (a')^* \\
&\quad = \text{FG}(L_*(\text{GPS})) \cdot (a')^* = L_*(\text{GPS}').
\end{aligned}$$

So GPS' is central. Now

$$\begin{aligned}
\text{SAFETY}^{*}_{\text{GPS}',a} &= \text{FG}(H) \cdot (a')^* \cap R_{\text{GPS}} \cdot (a')^* \cap (\text{Act}')^* \cdot a \\
&= \text{FG}(H) \cap R_{\text{GPS}} \cap \text{Act}^* \cdot a.
\end{aligned}$$

Because

$$\text{SAFETY}^{*}_{\text{GPS},a} = H \cap R_{\text{GPS}} \cap \text{Act}^* \cdot a \quad \text{and} \quad H = \emptyset \Leftrightarrow \text{FG}(H) = \emptyset,$$

it follows that (11) holds.

To end the proof we shall show that the $*f$-safety problem is unsolvable for arbitrary context-sensitive GPS.

Let C be an arbitrary context-sensitive set over Act and $a' \notin \text{Act}$. We consider the generalized processing system

$$\text{GPS} = (\text{Act} \cup \{a'\}, Q, F, q_0, \mathscr{H}),$$

$\mathscr{H} = (\{Q_a\}_{a \in \text{Act} \cup \{a'\}}, H)$ where $Q_a = Q_{a'} = Q$, $\forall a \in \text{Act}$ and $H = a'C$. We take F such that we have $R_{\text{GPS}} = (\text{Act} \cup \{a'\})^*$ and so

$$\text{SAFETY}^{*}_{\text{GPS},a'} = a'C.$$

It is true that $\text{SAFETY}^{*}_{\text{GPS},a'} = \emptyset$ iff $C = \emptyset$.

The emptiness problem for context-sensitive sets being unsolvable [14], the theorem follows. □

Let us note that we can easily prove an analogue of Theorem 4.2 for generalized processing systems and hence we have the following theorem.

Theorem 5.4. *The ωf-safety problem for central context-sensitive generalized processing systems is unsolvable.*

Remarks. (i) The construction in the proof of Proposition 4.6 holds when H is not prefix-closed as well. Indeed, defining the history control as in the definition of GPS, the entire construction becomes meaningful for GPS.

(ii) In the proof of Theorem 4.11, the construction yields H of the same type as H_i $(i = 1, 2)$ even if the H_i $(i = 1, 2)$ are context-free and non-prefix-closed.

As a conclusion we get a variant of Theorem 4.11 for GPS.

Theorem 5.5. *The $*f$-compatibility problem for rational generalized processing systems is solvable.*

The proof of the next theorem somewhat parallels that of Theorem 4.8. However, we need a special construction to get the desired inclusion solvable.

Theorem 5.6. *The ωf-compatibility problem for rational generalized processing systems is solvable.*

Proof. Let us consider GPS_1, GPS_2 two generalized processing systems and S a synchronization set.

We have that $L_\omega(\mathrm{GPS}_i)$, $i=1,2$, are closed sets and rational adherences. That is, $\mathrm{FG}(L_\omega(\mathrm{GPS}_i))\in \boldsymbol{R}$, and so $L_\omega(\mathrm{GPS}_i)$ are rational. Nivat's theorem shows that $L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2)=B_S^\omega(p_{\mathrm{GPS}_1},p_{\mathrm{GPS}_2})$ is a closed rational set.

Considering a similar GPS-construction as in Proposition 4.6, we have

$$L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2)\neq L_\omega(\mathrm{GPS})=\mathrm{Adh}(L_*(\mathrm{GPS})).$$

That is, in general, $L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2)$ is not an adherence, and so, we cannot derive from this that $\pi_1(L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2))$ is a closed set.

However, the fact can be obtained in another way. Consider GPS'_1, GPS'_2 two generalized processing systems and S' a synchronization set given as follows. Let 'a' be a new action. GPS'_i is obtained from GPS_i by adding the new action 'a' which can be followed only by other a's and taking $H'_i=H_ia^*$, $i=1,2$. We also put $S'=S\cup(a\times S)\cup(S\times a)\cup(a,a)$. Let GPS' be the corresponding system, analogous to that given by Proposition 4.6.

We have

$$L_{S'}^\omega(\mathrm{GPS}'_1,\mathrm{GPS}'_2)=L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2)\cup E(a,a)^\omega$$

and

$$L_{S'}^\omega(\mathrm{GPS}'_1,\mathrm{GPS}'_2)=L_\omega(\mathrm{GPS}')=\mathrm{Adh}(L_*(\mathrm{GPS}')).$$

We are not interested in detailing E'; the main point is that all its ω-words have, except for finite prefixes, the $(a,a)^\omega$ termination $(E'=E\cdot(a,a)^\omega)$.

Applying π_1 to the first relation we get

$$\pi_1(L_{S'}^\omega(\mathrm{GPS}'_1,\mathrm{GPS}'_2))=\pi_1(L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2))\cup\pi_1(E(a,a)^\omega).$$

By our second relation, π_1 commutes with Adh:

$$\mathrm{Adh}(\pi_1(L_*(\mathrm{GPS}')))=\pi_1(L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2))\cup\pi_1(E(a,a)^\omega),$$

from which we have

$$\pi_1(L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2))=\mathrm{Adh}(\pi_1(L_*(\mathrm{GPS}')))\cap S^\omega. \tag{12}$$

Now we can prove that $\pi_1(L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2))$ is closed.

Indeed, let $u\in\mathrm{Adh}(\pi_1(L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2))$. Then

$$\begin{aligned}\mathrm{FG}(u)&\subseteq\mathrm{FG}(\pi_1(L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2)))= \quad (\text{by }(12))\\ &=\mathrm{FG}(\mathrm{Adh}(\pi_1(L_*(\mathrm{GPS}'))\cap S^\omega)\\ &\subseteq\mathrm{FG}(\pi_1(L_*(\mathrm{GPS}')))\cap\mathrm{FG}(S^\omega).\end{aligned}$$

That is,

$$u\in\mathrm{Adh}(\pi_1(L_*(\mathrm{GPS}')))\cap S^\omega= \quad (\text{by }(12)) \quad \pi_1(L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2)).$$

The ωf-compatibility problem is given as follows:

$$L_\omega(\mathrm{GPS}_1)\subseteq\pi_1(L_S^\omega(\mathrm{GPS}_1,\mathrm{GPS}_2))?$$

which by closedness can be rephased as

$$\mathrm{FG}(L_\omega(\mathrm{GPS}_1)) \subseteq \mathrm{FG}(\pi_1(L_S^\omega(\mathrm{GPS}_1, \mathrm{GPS}_2)))?$$

As an inclusion between regular sets, it can be decided. □

Theorem 5.7. *Let* GPS_1 *and* GPS_2 *be two algebraic generalized processing systems. The* $*f$*-compatibility problem for* GPS_1 *with* GPS_2 *is unsolvable.*

Proof. Similarly as in the proof of Proposition 4.6 we can obtain

$$\begin{aligned} L_*(\mathrm{GPS}) &= L_S^*(\mathrm{GPS}_1, \mathrm{GPS}_2) \\ &= \{w \in S^* \mid w = (u_{11}, u_{21}) \cdots (u_{1n}, u_{2n}), \\ &\qquad u_i = u_{i1} \cdots u_{in} \in L_*(\mathrm{GPS}_i),\ i = 1, 2\} \end{aligned}$$

and H is context-free if H_1 and H_2 are context-free.

Let L_1, L_2 be two arbitrary context-free languages over some alphabet Act. Consider $\mathrm{GPS}_i = (\mathrm{Act}, Q_i, F_i, q_{0i}, \mathscr{H}_i)$, $i = 1, 2$, with $\mathscr{H}_i = (\{Q_a^i\}_{a \in \mathrm{Act}}, H_i)$, $Q_a^i = Q_i$, $F_i(a, q) = Q_i$, $\forall a \in \mathrm{Act}$ and $H_i = L_i$, $i = 1, 2$.

It is easy to see that $R_{\mathrm{GPS}_1} = R_{\mathrm{GPS}_2} = \mathrm{Act}^*$ and so

$$L_*(\mathrm{GPS}_i) = R_{\mathrm{GPS}_i} \cap L_i = L_i.$$

Now, we shall consider the synchronization set $S = \{(a, a) \mid a \in \mathrm{Act}\}$. Then

$$\begin{aligned} H &= \{w \in S^* \mid w = \langle u_1, u_2 \rangle, u_i \in H_i\} \\ &= \{w \in S^* \mid w = \langle u, u \rangle, u \in H_1 \cap H_2\}. \end{aligned}$$

The question "$L_*(\mathrm{GPS}_1) \subseteq \pi_1(L_*(\mathrm{GPS}))$?" is in fact reduced to

$$L_1 \subseteq \pi_1(H) \Leftrightarrow L_1 \subseteq L_1 \cap L_2 \Leftrightarrow L_1 \subseteq L_2$$

and the theorem follows because the inclusion problem for context-free languages is unsolvable. □

References

[1] J. Beauquier, Sur la compatibilité des systèmes de sécurité, in: B. Robinet, ed., *Proc. 3rd Internat. Symp. on Programming*, Paris (1978) pp. 110–125.

[2] J. Beauquier and M. Nivat, Applications of formal language theory to problems of security and synchronization, in: R. Book, ed., *Formal Language Theory* (Academic Press, New York, 1980) pp. 407–453.

[3] J. Berstel, *Transductions and Context-Free Languages* (Teubner, Stuttgart, 1979).

[4] M.A. Harrison, *Introduction to Formal Language Theory* (Addison Wesley, Reading, MA, 1978).

[5] M.A. Harrison, W.L. Ruzzo and J.D. Ullman, Protection in operating systems, *Comm. ACM* **19** (9) (1976).

[6] S. Istrail and C. Masalagiu, On protection in 'intelligent' processing systems, in: C. Ignat, ed., *New Directions in Artificial Intelligence* (Universitatea 'Al. I. Cuza', Iaşi, 1981).

[7] N.D. Jones and W.T. Laaser, Complete problems for deterministic polynomial times, *Theoret. Comput. Sci.* **3** (1977) 105–117.
[8] R.J. Lipton and L. Snyder, A linear time algorithm for deciding subject security, Res. Rept. 72, Yale University, 1976.
[9] R.J. Lipton and L. Snyder, On synchronization and security, *Foundations of Secure Computations* (1978) pp. 367–385.
[10] C. Masalagiu, Characterizing regularity of languages by data protection systems, *Annales de l'Université, Iaşi* **XXVIII** (2) (1982) to appear.
[11] M. Nivat, Languages algébraiques de mots infinis, *R.A.I.R.O. Theoret. Inform.* **12** (3) 1978).
[12] M. Nivat, Infinite words, infinite trees, infinite computations, in: J.W. de Bakker and J. van Leeuwen, eds., *Foundations of Computer Science III, Part 2* (Mathematical Center, Amsterdam, 1979) pp. 3–53.
[13] M. Nivat, Sur la synchronisation des processus, *Revue Technique Thomson-CSF* **11** (4) (1979) 899–919.
[14] A. Salomaa, *Formal Languages* (Academic Press, New York, 1973).
[15] C. Wood, E.B. Fernandez and R.C. Summers, Data base security: Requirements, policies and models, *IBM Systems J.* **19** (2) (1980) 229–253.
[16] L. Boasson and M. Nivat, Adherences of languages, *J. CSS* **20** (1980) 285–309.
[17] A. Arnold and M. Nivat, Comportements de processus, *Coll. AFCET*, Paris (1982) pp. 35–68.
[18] S. Istrail, Some remarks on nonalgebraic adherences, *Theoret. Comput. Sci.* **21** (1982) 341–349.