Theoretical Computer Science 8 (1979) 379–391

COMMUTATION PROPERTIES AND GENERATING SETS CHARACTERIZE SLICES OF VARIOUS SYNCHRONIZATION PRIMITIVES

Danny DOLEV

Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel

Communicated by Michael Rabin
Received January 1977

Abstract. Various synchronization primitives are described by adding and testing integer vectors, or by using "Petri Nets". A slice represents a local behavior, described by permissible sequences of distinct actions of the system.

We present a double characterization of slices defined by various synchronization primitives: In terms of generating sets and dually in terms of commutation properties. A typical form of a commutation property is: The set of sequences of past actions which disallow a certain coming action is closed under certain permutations. The synchronization primitives treated here include various systems which lie between PV and Vector Replacement Systems or Petri Nets.

1. Introduction

Various synchronization primitives can be described by adding and testing of integer vectors, each primitive imposes some restrictions on the form of the vectors. The study of the behavior of synchronization primitives leads to the study of slices or finite languages which are prefix closed. The letters of the alphabet Σ of the language represent actions of the form "test a vector and add a vector". Words of the language represent feasible (non-blocked) sequences of actions. The restriction of slices means we consider only words composed of distinct letters. The papers of Lipton [11, 12] explain how and why the comparison of slices plays an important role in the study of synchronization primitives.

We characterize the slices of: Vector Replacement Systems (VRS in short, due to Keller [10]), Unrestricted VRS (U-VRS, see Definition 2), Vector Addition Systems (VAS, see [9]), Petri Nets (PN, see [8]), variants of PN (see [5]), PVgeneral (PVg, see [4]) and PVmultiple (PVm, see [15]). We extend the notion of generating sets ([6], see also [3]) and use commutation relations (see [1]) to characterize the slices which are generated by the above systems.

By the definitions of VAS and VRS one can easily see that VAS $\subseteq$ VRS ($A \subseteq B$ means that every slice which is defined by the system A can be defined by the system B. If, moreover, there is a B-slice which is not an A-slice then $A \subsetneq B$). Lipton, Snyder and Zalcstein [14] proved VAS⊊VRS (with respect to Σ-slices, which is a restricted family of slices). They also gave relations between some other synchronization primitives, but the relations between PVm, PVg and PVc (PVchunck, defined in [16]) was left open in their work. In an earlier report [1] we proved PVg ≡ PVm⊊PVc (again with respect to Σ-slices) and using commutation properties we also characterized Σ-slices of VAS, VRS, PN and Loopless–PN. In the present article we simplify the constructions and extend the characterization to general slices defined by the above systems.

We prove the following results (in sense of slices):

(i) Prefix-Permutable ≡ Unrestricted–VRS ≡ VRS ≡ GPN ≡ PN. A slice π is Prefix-Permutable is $\alpha \in \pi$ and $\alpha\sigma \notin \pi$ implies $\text{per}(\alpha)\sigma \notin \pi$, where $\text{per}(\alpha)$ is any permutation of α. Definitions of the various commutation relations are given in Section 2 and those of systems in Section 3.

(ii) Prefix-Abelian ≡ VAS ≡ Loopless–GPN ≡ Loopless–PN.

(iii) Prefix-Permutable-Bipolar ≡ PVg ≡ PVm⊊PVc.

Henderson-Zalcstein [6] introduced the extended notion of slice and the notion of generating sets and characterized the slices of PV and PV_c; but their characterization of PV_m was incorrect, as pointed out in [3]. After the present article was submitted for publication, we learned of a manuscript of Henderson-Zalcstein [7], in which they obtained most of their characterizations we present here (including a correct one for PV_m), but not including the most general prefix permutable ones.

A commutation relation provides a convenient method of finding that a language of system A in included (or not included) in the language of system B, while generating sets, described below, are convenient for obtaining the complement, i.e., the forbidden sequences of ths systems. For instance, PV two-ways bounded (PVk, see [14]), can be seen to be Prefix Abelian, thus representable by VAS, contrary to the impression obtained from Fig. 6.1 in [14].

An earlier version of the results represented here was presented in August 1977 in the Waterloo Conference of Theoretical Computer Science [2].

2. Slices

In this section we define various types of slices derived by commutation relations and we prove some relations among them. We start by giving the notations we use in this paper. Let Σ be a finite alphabet, we let Σ^+ be the set of all the words which are composed of distinct letters of Σ. In what follows, late lower case Greek letters denote elements of Σ. Early lower case letters denote words of Σ^+. We denote by $\text{per}(\gamma)$ a word obtained by permuting the letters of γ.

We list some conventions used in the sequel. Let $\alpha = \sigma_1 \cdots \sigma_k$ be a word in Σ^+, then we denote the ith element of α by α_i; the length of α by $\#\alpha$; the set of elements of α by $W(\alpha)$; the i-prefix of α (i.e., $\sigma_1 \cdots \sigma_i$) by ${}_i\alpha$, where ${}_0\alpha = \Lambda$ is the empty word. Let Z denote the integers, N denote the non-negative integers and $(-N)$ the non-positive integers. $\mathbf{0}$ is the zero-vector and for $s \in Z^n$ s_i denote the ith coordinate of s. Let s_+ (s_-) be the set of all the coordinates of the vector s with strictly positive (negative) values.

Definition 1. A *slice* π over Σ is a prefix closed subset of Σ^+. If in addition $\Sigma \subseteq \pi$, the slice π is called a Σ-slice.

Our "Σ-slice" coincides with "slice" in [11, 12]. Thus our results here are not directly an extension and an amplification of Lipton's results. But in fact in our earlier report [1] we proved the results for Σ-slices.

We are mainly interested in Prefix-Permutable slices because most previously defined synchronization primitives which were described in [14] (except Lock/Unlock, PVor, Bilogic and Program Schemata) give rise to Prefix-Permutable slices.

Definition 2. A slice π is called a *Prefix-Permutable slice* (P-slice in short) if for every α, σ and every $\text{per}(\alpha)$

$$\alpha \in \pi,\ \alpha\sigma \notin \pi \Rightarrow \text{per}(\alpha)\sigma \notin \pi. \tag{2.1}$$

Note that a PΣ-slice is called a commutative slicé in [14]. The prefix permutability property (2.1) extends to words $\beta \in \Sigma^+$ instead of symbols $\sigma \in \Sigma$,

$$\alpha \in \pi,\ \alpha\beta \notin \pi \Rightarrow \text{per}(\alpha)\beta \notin \pi.$$

Example 1. The following slice can easily be seen to be a P-slice.

$$\pi_1 = \{\Lambda, \sigma, \tau, \rho, \sigma\tau, \tau\sigma, \rho\tau, \sigma\tau\rho, \tau\sigma\rho\}.$$

The next notion will characterize the slices of VAS.

Definition 3. A slice π is called a *Prefix-Abelian slice* (PA-slice in short) if for every $\alpha\sigma \in \Sigma^+$ and every $\text{per}(\alpha\sigma)$,

$$\alpha \in \pi,\ \alpha\sigma \notin \pi \Rightarrow \text{per}(\alpha\sigma) \notin \pi. \tag{2.2}$$

Property (2.2) is the *abelian* property. Intuitively it means that by reordering the operations we do not change the total result. The following lemma is obvious.

Lemma 1. *Every PA-slice is a P-slice.*

Note the fact that in a PA-slice π,

$$\alpha\sigma \in \pi,\ \alpha\dot{\sigma}\tau \notin \pi \Rightarrow \alpha\tau\sigma \notin \pi. \tag{2.3}$$

The abelity condition is strictly stronger than (2.3). For instance, the slice $\{\Lambda, \rho, \sigma, \sigma\tau, \rho\tau, \rho\tau\sigma\}$ satisfying (2.3), is a P-slice but is not a PA-slice. We prove later that VAS-slice is a PA-slice, therefore (2.3) is not sufficient for the characterization of VAS-slices. Note that a Σ-slice which satisfies (2.3) and is a PΣ-slice but is not a PAΣ-slice, contains at least five letters in the alphabet (an example is given in [1]).

To obtain the PA-slice (abelian) we exhausted the full permutation group on the "history of the process"; to define stronger commutation relations we have to distinguish special properties of the letters (actions) of Σ, using the notion of movers which appears in [13].

Definition 4. Let $\pi \subseteq \Sigma^+$ and let $\sigma \in \Sigma$, we call σ a *right-mover* with respect to π, if for every $\alpha \in \Sigma^+$ and $\tau \in \Sigma$,

$$\alpha\sigma\tau \in \pi \Rightarrow \alpha\tau\sigma \in \pi. \tag{2.4}$$

Similarly, we call σ a *left-mover* with respect to π, if for every α and τ,

$$\alpha\tau\sigma \in \pi \Rightarrow \alpha\sigma\tau \in \pi.$$

We define now the relation "PB" which will characterize PVm-slices.

Definition 5. Let $R\sigma : \alpha \to \alpha\sigma$ be the operation of *right-attachment.* A slice π is called a *Bipolar-slice* if every element $\sigma \in \Sigma$ is either a right-mover or the operation $R\sigma$ preserves the slice, i.e., for every α

$$\alpha \in \pi,\ \alpha\sigma \in \Sigma^+ \Rightarrow \alpha\sigma \in \pi. \tag{2.5}$$

We let Σ_l be the set of all elements in Σ with $R\sigma$-property and denote $\Sigma_r = \Sigma \setminus \Sigma_l$. Note that every element in Σ_r is a right-mover. Moreover, the following lemma is easily derived and is, in fact, equivalent to the above definition.

Lemma 2. *A slice is a Bipolar-slice iff* $\Sigma = \Sigma_l \cup \Sigma_r$, *where*

(i) $\Sigma_l \cap \Sigma_r = \emptyset$,

(ii) *every element in* Σ_r *is a right-mover which does not satisfy (2.5),*

(iii) *every element in* Σ_l *is a left-mover and satisfies* (2.5).

In what follows we are interested only in *Prefix-Permutable-Bipolar slices* (PB-slices, in short), i.e., P-slices which are also Bipolar.

Example 2. The slice $\pi_2 = \{\Lambda, \sigma, \tau, \rho, \delta, \sigma\tau, \tau\sigma, \rho\delta, \delta\rho\}$ is obviously a PB-slice (in which $\Sigma_l = \emptyset$). Lipton [11, 12] showed that it is PVm-definable.

Lemma 3. *Every* PB-*slice is a* PA-*slice.*

Proof. Let π be a PB-slice. Let α, σ be such that

(1) $\qquad \alpha \in \pi$ and $\alpha\sigma \notin \pi$.

We have to prove that for every permutation per($\alpha\sigma$), on the word $\alpha\sigma$, per($\alpha\sigma$) $\notin \pi$. Assume, conversely, that

(2) per($\alpha\sigma$) $\in \pi$.

Distinguish between the two cases:

(i) If $\sigma \in \Sigma_l$, then $\alpha\sigma \notin \pi$ iff $\alpha\sigma \notin \Sigma^+$. Therefore also per($\alpha\sigma$) $\notin \Sigma^+$ which contradicts (2).

(ii) Assume $\sigma \in \Sigma_r$. In this case we can move σ to the right by repeated application of (2.4) on per($\alpha\sigma$) and get per'(α)$\sigma \in \pi$. We know that $\alpha \in \pi$, therefore, by the prefix permutability, we conclude that $\alpha\sigma \in \pi$, which contradicts (1). Thus we proved that in any case, per($\alpha\sigma$) $\notin \pi$, which completes the proof.

Corollary 4. PB $\subsetneq$ PA $\subsetneq$ P.

Proof. The relations PB $\subseteq$ PA $\subseteq$ P are derived from Lemma 1 and Lemma 3. The slice π_1, which appears in Example 1 is a P-slice which is not a PA-slice. Therefore PA $\subsetneq$ P. The slice $\pi_3 = \{\Lambda, \sigma, \sigma\tau, \sigma\tau\rho\}$ is a PA-slice which is not a PB-slice. Thus we have proved the desired relations.

Note that the above relations remain valid for Σ-slices.

3. The vector-systems

Here we define the systems of integer vectors which we work with, and the slices derived from them.

Definition 6. (1) An *Unrestricted Vector Replacement System* (U–VRS) is a system $\Phi = (\Sigma, t, r, s^0)$ where

(i) Σ is a set of symbols, t and r are maps from Σ to Z^n ($t(\sigma)$ is called a *test-vector* and $r(\sigma)$ is called a *replacement-vector*). For $s \in Z^n$ we define

$$(s)\sigma = s + r(\sigma) \quad \text{and} \quad (s)x\sigma = (s)\alpha + r(\sigma).$$

The coordinates of the vector $s \in Z^n$ are called *semaphores*.

(ii) $s^0 \in Z^n$ is a given initial value of the *semaphores'-vector*.

(2) An U–VRS system Φ induces a slice $\pi = \Pi(\Phi)$, which is the set of all words $\alpha = \tau_1 \cdots \tau_m \in \Sigma^+$ such that

$$s^0 + \sum_{i=1}^{k-1} r(\tau_i) + t(\tau_k) \geq 0 \quad \text{for every } 1 \leq k \leq m. \tag{3.1}$$

It is easy to verify that $\Pi(\Phi)$ is indeed a slice (i.e., prefix closed).

(3) A slice π is called *U–VRS definable* is there is an U–VRS system Φ such that $\pi = \Pi(\Phi)$.

Definition 7. (1) An U–VRS system Φ is called a *Vector Replacement System* (VRS) if Φ satisfies the following restriction:

$$r(\sigma)-t(\sigma)\geq \mathbf{0}, \quad \text{for every } \sigma\in\Sigma. \tag{3.2}$$

(2) A VRS system Φ is called a *Generalized Petri Net* (GPN) if for every $\sigma\in\Sigma$,

$$t(\sigma)\in(-N)^n \quad \text{and} \quad t_-(\sigma)\cap r_+(\sigma)=\emptyset. \tag{3.3}$$

Recall that t_- (t_+) is the set of all the strictly negative (positive) coordinates of t. Note that (3.2) and (3.3) imply that $r_-(\sigma)\subseteq t_-(\sigma)$.

(3) A GPN is called a *Petri-Net* (PN) if for every $\sigma\in\Sigma$,

$$t(\sigma)\in\{0,-1\}^n \quad \text{and} \quad r(\sigma)\in\{0,1,-1\}^n. \tag{3.4}$$

Example 3. Let π_1 be the slice $\pi_1=\{\Lambda,\sigma,\tau,\rho,\sigma\tau,\tau\sigma,\rho\tau,\sigma\tau\rho,\tau\sigma\rho\}$. One can show, using the following PN-system, that π_1 is a PN-definable. Take $\Phi=(\Sigma,t,r,s^0)$ where,

$$\begin{aligned}
\Sigma &= \{\sigma,\tau,\rho\}, & s^0 &= (1,1,1,2),\\
t(\sigma) &= (-1,0,-1,-1), & r(\sigma) &= (-1,1,0,0),\\
t(\tau) &= (0,-1,0,-1) & r(\tau) &= (1,-1,1,-1),\\
t(\rho) &= (-1,-1,-1,-1), & r(\rho) &= (0,0,-1,-1).
\end{aligned}$$

Note that π_1 is also a P-slice, as claimed in Example 1.

From Definition 6 and Definition 7, we conclude at once that

$$\text{U–VRS}\supseteq\text{VRS}\supseteq\text{GPN}\supseteq\text{PN}. \tag{3.5}$$

We shall prove later that these systems are equivalent in the sense that all generate the same family of slices.

Definition 8. (1) A VRS system Φ is called a *Vector Addition System* (VAS) if for each $\sigma\in\Sigma$, $t(\sigma)=r(\sigma)$.

(2) A GPN (PN) is called a *Loopless–GPN* (*Loopless–PN*) if for every $\sigma\in\Sigma$, $t_-(\sigma)=r_-(\sigma)$. The term Loopless is connected to the network representation of Petri Nets.

Example 4. Let π_3 be the slice $\pi_3=\{\Lambda,\sigma,\sigma\tau,\sigma\tau\rho\}$. One can show, using the following Loopless–PN system Φ, that π_3 is a L–PN definable. Take $\Phi=(\Sigma,t,r,s^0)$ where

$$\begin{aligned}
\Sigma &= \{\sigma,\tau,\rho\}, & s^0 &= (0,0,1),\\
t(\sigma) &= (0,0,-1), & r(\sigma) &= (1,1,-1),
\end{aligned}$$

$$t(\tau) = (-1, 0, 0), \qquad r(\tau) = (-1, 1, 1),$$
$$t(\rho) = (0, -1, -1), \qquad r(\rho) = (1, -1, -1).$$

Note that π_3 is already a PA-slice.

We conclude easily, from Definition 6 to 8 that

$$\text{L–GPN} \supseteq \text{L–PN}. \tag{3.6}$$

Note that we cannot determine easily how VAS relates to (3.6). We introduce now some special types of VAS.

Definition 9. Let Φ be a VAS-system. If for every $\sigma \in \Sigma$,
(1) $r(\sigma) \in N^n \cup (-N)^n$, then Φ is called a *PVg-system*,
(2) $r(\sigma) \in \{0, 1\}^n \cup \{0, -1\}^n$, then Φ is called a *PVm-system*,
(3) $r(\sigma) \in N^n \cup (-N)^n$ and contains at most one non-zero coordinate, then Φ is called a *PVc-system*.

Note that PVg $\supseteq$ PVm and PVg $\supseteq$ PVc. The following lemma and corollary are easily derived.

Lemma 5.
(i) *Every U–VRS-definable slice is a P-slice.*
(ii) *Every VAS-definable slice is a PA-slice.*
(iii) *Every L–GPN-definable slice is a PA-slice.*
(iv) *Every PVg-definable slice is a PB-slice.*

Corollary 6.
(i) PB $\subseteq$ GPN $\subseteq$ VRS $\subseteq$ U–VRS $\subseteq$ P.
(ii) L–PN $\subseteq$ P–GPN $\subseteq$ PA and L–PN $\subseteq$ VAS $\subseteq$ PA.
(iii) PVm $\subseteq$ PVg $\subseteq$ PB and PVc $\subseteq$ PVg $\subseteq$ PB.

4. Generating sets

The notion of a generating set for a slice is a useful tool introduced in [5]. We use several variants of this notion to characterize the slices of various synchronization primitives.

Definition 10. (1) A collection of couples,

$$G = \{(A_i, \sigma_i) \mid A_i \subseteq \Sigma, \sigma_i \in \Sigma, i = 1, \ldots, n\}$$

is called a *generating set* (GS, in short) for a language $\pi \subseteq \Sigma^+$ (notation $\pi = \Pi(G)$) if

$$\alpha \notin \pi \quad \text{iff} \quad (W({}_i\alpha), \alpha_{i+1}) \in G \quad \text{for some } 0 \leq i \leq \#\alpha - 1.$$

(2) A collection of sets, G, is called a *generating set of type A* (GS–A) for a language $\pi \subseteq \Sigma^+$ if

$$\alpha \notin \pi \quad \text{iff} \quad W({}_i\alpha) \in G \quad \text{for some } 1 \leq i \leq \#\alpha.$$

(3) For a GS–A we define

$$\Sigma_s = \{\sigma \mid \exists A \text{ s.t. } A \cup \{\sigma\} \in G \text{ and } A \notin G\}$$

A GS–A is called a *generating set of types B* (GS–B) if it is closed under union of "Stoppers", i.e.,

$$A \in G, \qquad B \subseteq \Sigma_s \Rightarrow A \cup B \in G. \tag{4.1}$$

Note that GS–A is the notion of generating set defined in [5]. A language π defined by a generating set is obviously prefix closed, hence a slice. By closing a GS–A under (4.1) we obtain a GS–B, the set of stoppers of the obtained GS–B is a subset of that of GS–A.

Example 5. (1) Let π_1 be the P-slice appearing in Example 1 ($\pi_1 = \{\Lambda, \sigma, \tau, \rho, \sigma\tau, \tau\sigma, \rho\tau, \sigma\tau\rho, \tau\sigma\rho\}$). Then the following GS is a generating set for π_1:

$$G_1 = \{(\{\sigma\}, \rho), (\{\tau\}, \rho), (\{\rho\}; \sigma), (\{\tau, \rho\}, \sigma)\}.$$

(2) Let π_3 be the PA-slice appearing in Example 4 ($\pi_3 = \{\Lambda, \sigma, \sigma\tau, \sigma\tau\rho\}$). The following GS–A generates π_3:

$$G_2 = \{\{\tau\}, \{\rho\}, \{\sigma, \rho\}\}.$$

(3) Let $\pi_4 = \{\Lambda, \sigma, \sigma\tau\}$ be a slice, over the alphabet $\Sigma = \{\sigma, \tau, \rho\}$, which can easily be seen to be PB-slice. One can check and see that the following generating sets generate π_4:

$$G_4' = \{\{\tau\}, \{\rho\}, \{\sigma, \rho\}, \{\sigma, \tau, \rho\}\} \qquad \text{(a GS–A slice)},$$

$$G_4 = \{\{\tau\}, \{\rho\}, \{\sigma, \rho\}, \{\sigma, \tau, \rho\}, \{\tau, \rho\}\} \qquad \text{(a GS–B slice)}$$

and moreover also that, $\Sigma_s(G_4') \supseteq \Sigma_s(G_4)$.

The connection between the gnerating sets and the slices is given in the following theorem.

Theorem 7. (1) *A slice is a* P-*slice iff it has a* GS.

(2) *A slice is a* PA-*slice iff it has a* GS–A.

(3) *A slice is a* PB-*slice iff it has a* GS–B.

Proof. (1) Let π be a P-slice. We define a GS, G, and prove that $\pi = \Pi(G)$. Define

$$G = \{(W(\alpha), \sigma) \mid \alpha \in \pi, \alpha\sigma \in \Sigma^+ \text{ and } \alpha\sigma \notin \pi\}.$$

Let $\bar{\pi} = \Pi(G)$ (the slice generated by G). We prove that for every $\alpha \in \Sigma^+$ $\alpha \notin \pi$ iff $\alpha \notin \bar{\pi}$.

Assume, first, that $\alpha \notin \pi$. Let i be the first index such that ${}_i\alpha \in \pi$ and ${}_{(i+1)}\alpha \notin \pi$. By definition $(W({}_i\alpha), \alpha_{i+1}) \in G$, which implies that $\alpha \notin \bar{\pi}$.

Conversely, assume that $\alpha \notin \bar{\pi}$, let i be the first index such that $(W({}_i\alpha), \alpha_{i+1}) \in G$. By definition, there is a per$({}_i\alpha)$, such that

$$\text{per}({}_i\alpha) \in \pi \quad \text{and} \quad \text{per}({}_i\alpha)\alpha_{i+1} \notin \pi.$$

Therefore, by the prefix permutability property we conclude ${}_i\alpha\ \alpha_{i+1} \notin \pi$, which implies that $\alpha \notin \pi$.

We now prove that every GS generates a P-slice. Let G be a GS and let $\pi = \Pi(G)$. Then π is a slice. Let α and σ be such that $\alpha \in \pi$ and $\alpha\sigma \notin \pi$. Then by definition $(W(\alpha), \sigma) \in G$. For any per$(\alpha)$, per $(\alpha)\sigma \notin \pi$ since $(W(\text{per}(\alpha)), \sigma) = (W(\alpha), \sigma) \in G$.

(2) The proof is similar to that of (1) but with the following GS–A (instead of GS):

$$G = \{A \cup \{\sigma\} \mid (A, \sigma) \in \text{GS}\}.$$

(3) Let π be a PB-slice. We prove that it has a GS–B which generates it. By (2) it has a GS–A, G_1. Extend G_1 to a set G by closing G_1 under condition (4.1), G is obviously a GS–A and already a GS–B. From Definitions 5, 10 and Lemma 2 one can prove that $\Sigma_s = \Sigma_r$. We have to prove that G generates π. Note that

$$G_1 \subseteq G \Rightarrow \Pi(G_1) \supseteq \Pi(G). \tag{4.2}$$

After proving (2) it is enough to prove that $\alpha \notin \Pi(G_1)$ iff $\alpha \notin \Pi(G)$. By (4.2) it suffices to prove $\alpha \notin \Pi(G) \Rightarrow \alpha \notin \Pi(G_1)$.

Let $\alpha \notin \Pi(G)$ and let i be an index such that $W({}_i\alpha) \in G$. Assume that $\alpha \in \Pi(G_1)$ and that $W({}_i\alpha) \notin G_1$. Let B be the extension of A, such that $W({}_i\alpha) = A \cup B$, where $A \in \Pi(G_1)$ and $B \subseteq \Sigma_s \backslash A$. $\Pi(G_1)$ is a PB-slice and we assume that $\alpha \in \Pi(G_1)$. Therefore, by the right-moving of the elements of B (since $\Sigma_s = \Sigma_r$), we conclude $\alpha = \bar{\alpha}\beta \in \Pi(G_1)$, where $W(\bar{\alpha}) = A$ and $W(\beta) = B$.

But this is a contradiction, since if $W(\bar{\alpha}) \in G_1$, then by definition $\bar{\alpha}\beta \notin \Pi(G_1)$. Thus we conclude that $\alpha \notin \Pi(G_1)$.

It remains to prove that every GS–B generates a PB-slice. Let G be a GS–B. By (2) and Lemma 1, $\Pi(G)$ is a PA-slice and a P-slice. Therefore it is enough to prove the right-moving for the elements of Σ_s $(=\Sigma_r)$ and the right-attachment $(R\sigma)$ for the elements in $\Sigma \backslash \Sigma_s$ $(=\Sigma_l)$.

Let $\sigma \in \Sigma_s$ and let $\alpha\sigma\tau \in \pi = \Pi(G)$. If $\alpha\tau \in \pi$ then by the abelity we conclude that $\alpha\tau\sigma \in \pi$. If $\alpha\tau \notin \pi$ then $W(\alpha\tau) \in G$ (since $\alpha \in \pi$). By property (4.1), $W(\alpha\tau) \cup \{\sigma\} \in G$, since $\{\sigma\} \subseteq \Sigma_s$. But this contradicts the assumption that $\alpha\sigma\tau \in \pi$. Therefore we proved the right-moving of the elements of Σ_s.

Let $\tau \in \Sigma \backslash \Sigma_s$, $\alpha \in \pi$ such that $\alpha\tau \in \Sigma^+$. If $\alpha\tau \notin \pi$ then by definition $W(\alpha\tau) \in G$ by which we conclude that $\tau \in \Sigma_s$, a contradiction (since Σ is not empty, otherwise the theorem is trivial).

5. Concrete realizations of systems defined by generating sets

We start by constructing a Petri Net (Loopless Petri Net) realization for a P-slice (PA-slice), thus proving that the chains of containments in Corollary 6 are actually equivalences. Similarly we construct a PVmultiple realization for PB-slices.

Theorem 8 (Characterization of PN-slices). PN ≡ P.

Proof. In view of Corollary 6 it is enough to prove that PN ⊇ P. Let π be a P-slice and let G be a GS such that $\pi = \Pi(G)$. We first define the equivalent PN-system (Definition 7) $\Phi = (\Sigma, t, r, s^0)$. Take Σ as the set of symbols. Let n be the number of elements of G. To every pair (A_i, σ_i) in G we associate a semaphore s_i and define $s_1^0 = |A_i|$ (the number of elements in A_i). For every $\sigma \in \Sigma$ define

$$[t(\sigma)]_i = \begin{cases} -1 & \text{if } \sigma \in A_i \text{ or } \sigma = \sigma_i \\ 0 & \text{otherwise,} \end{cases}$$

$$[r(\sigma)]_i = \begin{cases} -1 & \text{if } \sigma \in A_i \\ 0 & \text{if } \sigma = \sigma_i \\ +1 & \text{otherwise.} \end{cases}$$

Let $\bar{\pi} = \Pi(\Phi)$. One can easily prove that $\bar{\pi}$ is already a PN-definable slice.

Claim. (i) $(s^0)\alpha \geq \mathbf{0}$ for all α in Σ^+,
(ii) $[(s^0)\alpha]_i = 0$ iff $[W(\alpha) = A_i$ or $W(\alpha) = A_i \cup \{\sigma_i\}]$,
(iii) let $\sigma \notin W(\alpha)$ then $[(s^0)\alpha + t(\sigma)]_i = -1$ iff $\sigma = \sigma_i$ and $W(\alpha) = A_i$.

Proof. Indeed, the initial value of semaphore i is $|A_i|$. The action "add $r(\sigma)$" decreases it iff $\sigma \in A_i$. Thus (i) is valid.

(ii) is valid since semaphore i gets its minimal value iff it is not increased in any symbol, and this can happen iff $W(\alpha) = A_i$ or $W(\alpha) = A_i \cup \{\sigma_i\}$.

The value of semaphore i can be less than zero iff $[(s^0)\alpha]_i = 0$ and $[t(\sigma)]_i = -1$. Therefore, by (ii) and the definition of $t(\sigma)$ we conclude that $\sigma = \sigma_i$, which proves (iii).

To prove that $\pi = \bar{\pi}$ we have to prove that $\alpha \in \pi$ iff $\alpha \in \bar{\pi}$. One can see immediately that it suffices to prove that $\alpha \in \pi \Rightarrow [\alpha\sigma \notin \pi$ iff $\alpha\sigma \notin \bar{\pi}]$. Let $\alpha \in \pi$ and assume that $\alpha\sigma \notin \pi$. By the definition of a generating set it implies that $(W(\alpha), \sigma) = (A_i, \sigma_i) \in G$, for some $1 \leq i \leq n$. By the above claim we know that $[(s^0)\alpha + t(\sigma)]_i = -1$, thus $\alpha\sigma \notin \bar{\pi}$.

Assume now that $\alpha\sigma \notin \bar{\pi}$. Let s_k be a semaphore that becomes negative after adding $t(\sigma)$ to $(s^0)\alpha$. By the above claim $W(\alpha) = A_i$ and $\sigma = \sigma_i$. From the construction of a generating set we conclude that for some per(α), per(α)$\sigma \notin \pi$. But using the prefix permutability we conclude that $\alpha\sigma \notin \pi$.

From Corollary 6 and Theorem 8 we conclude:

Corollary 9. P ≡ U–VRS ≡ VRS ≡ GPN ≡ PN.

Example 6. One can use the method of the proof of Theorem 8 to get from the generating set of π_1 (Example 5) the PN-system which appears in Example 3.

Theorem 10 (Characterization of Loopless–PN). PA ≡ L–PN.

Proof. The method of proof is very similar to that of Theorem 8. It is enough to prove that every PA-slice is a L–PN-definable slice. Let π be a PA-slice and let G be a GS–A such that $\pi = \Pi(G)$. We define the equivalent L–PN-system $\Phi = (\Sigma, t, r, s^0)$. Take Σ as the set of symbols. To every $A_i \in G$ we associate a semaphore s_i and define $s_i^0 = |A_i| - 1$. For every $\sigma \in \Sigma$ define

$$[t(\sigma)]_i = \begin{cases} -1 & \text{if } \sigma \in A_i \\ 0 & \text{if } \sigma \notin A_i \end{cases}$$

$$[r(\sigma)]_i = \begin{cases} -1 & \text{if } \sigma \in A_i \\ +1 & \text{if } \sigma \notin A_i \end{cases}$$

Note that PN are loopless in case that the test vector is equal to the negative part of the replacement vector.

Claim. Let $\alpha \in \Sigma^+$, then for every $\tau \in \Sigma \setminus W(\alpha)$, $(s^0)\alpha + t(\tau) \geq -1$. Moreover, $[(s^0)\alpha + t(\tau)]_i = -1$ iff $\alpha\tau \notin \pi$, where $A_i = W(\alpha) \cup \{\tau\}$.

Proof. The proof of the claim is straightforward from the definitions. The rest of the proof is analogous to the proof of Theorem 8.

Using the same method it is proved in [2] that PA ≡ VAS and therefore from Corollary 6 and Theorem 10 we conclude

Corollary 11. PA ≡ VAS ≡ L–GPN ≡ L–PN.

Example 7. One can use the above method to get, from the GS–A of π_3 (Example 5), the L–PN system which appears in Example 4.

Theorem 12 (Characterization of PVm). PB ≡ PVm.

Proof. From Corollary 6 we conclude that it is enough to prove that PB ⊆ PVm. Let π be a PB-slice. By Theorem 7 we know that there is a GS–A G such that $\pi = \Pi(G)$ and that G is closed under the property $A_i \in G$ and $B \subseteq \Sigma_s \Rightarrow A_i \cup B \in G$, where $\Sigma_s = \Sigma_r = \{\sigma \mid \exists A \text{ s.t. } A \notin G \text{ and } A \cup \{\sigma\} \in G\}$.

Define the equivalent PVm-system $\Phi = (\Sigma, t, s^0)$. Take Σ as the set of symbols, let n be the number of elements in G. To every $A_i \in G$ associate a semaphore s_i with initial value $s_i^0 = |\Sigma_s \cap A_i| - 1$. To every $\sigma \in \Sigma$ define

$$\text{if } \sigma \in \Sigma_s : [t(\sigma)]_i = \text{if } \sigma \in A_i \text{ then } -1 \text{ else } 0$$

$$\text{if } \sigma \in \Sigma \backslash \Sigma_s : [t(\sigma)]_i = \text{if } \sigma \in A_i \text{ then } 0 \text{ else } +1.$$

Note that $\Sigma_s = \Sigma_r$ and $\Sigma \backslash \Sigma_s = \Sigma_l$.

Claim. The semaphore s_i becomes negative (in fact assumes the value -1) after α iff $W(\alpha) = A_i \cup B$, where $B \subseteq \Sigma_s \backslash A_i$ and $A_i \in G$.

Proof. The proof of the claim is straightforward from the definitions. As noted in the above theorems, it is enough to prove that $\alpha \in \pi \Rightarrow [\alpha\sigma \notin \pi \text{ iff } \alpha\sigma \notin \bar{\pi}]$, where $\bar{\pi} = \Pi(\Phi)$. By definitions, the case $\alpha\sigma \notin \pi$ implies $\alpha\sigma \notin \bar{\pi}$ is immediate. Therefore assume that $\alpha\sigma \notin \bar{\pi}$. Let s_i be a semaphore which becomes negative after $\alpha\sigma$. By the above claim $W(\alpha) = A_i \cup B$, where $B \subseteq \Sigma_r \backslash A_i$ and $A_i \in G$. By property (4.1), $A_i \in G$ implies that $A_i \cup B \in G$. hence $\alpha \notin \pi$.

Corollary 13. $\text{PB} \equiv \text{PVm} \equiv \text{PVg} \supsetneq \text{PVc}$.

Proof. By Corollary 6 and Theorem 12 we conclude that $\text{PVm} \equiv \text{PVg} \equiv \text{PB}$. PVc-definable slice is a PB-slice, but as was proved in [11, 12] there is a PVm-definable slice which is not a PVc-definable slice. Therefore $\text{PVc} \supsetneq \text{PVm}$.

In the sense of slice inclusion we have thus settled the open alternative in Fig. 6.1 (in [14]), showing that Case (a) is valid (also for Σ-slices as shown in [1]). Henderson and Zalcstein [6] have claimed to have settled the same point but their characterization of PVm (by GS–A sets) was mistaken.

Acknowledgements

It is my pleasure to thank Professor Eliahu Shamir, from the Hebrew University, Jerusalem, for his fruitful advice during the preparation of this work.

References

[1] D. Dolev, Abstract characterization of slices of various synchronization primitives, Technical Report 7609/1, Weizmann Institute of Science, Israel, Appl. Math. Dept. (September 1976).

[2] D. Dolev, Local Characterization of models of synchronization primitives, Proc. the Waterloo Conf. on Theoretical Computer Science (August 1977).

[3] D. Dolev and E. Shamir, Commutation relations of slices characterize some synchronization primitives, Information Processing Lett. **7** (1) (1978) 7–9.
[4] V.G. Cerf, Multiprocessors, semaphores and a graph model of computation, Ph.D. Thesis, UCLA (1972).
[5] M. Hack, Decision problems for Petri nets and vector additional systems, MAC Technical Memo 59 (1975).
[6] P.B. Henderson and Y. Zalcstein, Characterization of the synchronization languages for PV systems, Conf. Rec. of Seventeenth FOCS Ann. Symp. (1976).
[7] P.B. Henderson and Y. Zalcstein, Synchronization problems solvable by generalized PV systems, JACM. (To appear.)
[8] A.W. Holt and F. Commoner, Events and conditions, Rec. of Project MAC Conf. on Concurrent Systems and Parallel Computation (1970).
[9] R.M. Karp and R.E. Miller, Parallel program schemata, JCSS **3** (1969).
[10] R.M. Miller, Vector replacement systems: A formalism for modeling asynchronous systems, Technical Report 117, Princeton University Computer Science Laboratory (1972).
[11] R.J. Lipton, On synchronization primitive systems, Yale Computer Science Research Report **22** (1973).
[12] R.J. Lipton, Limitations of synchronization primitives with conditional branching and global variables, Proc. Sixth ACM SIGACT Symp. on Theory of Computing (1974).
[13] R.J. Lipton, Reduction: A method of proving properties of parallel programs, CACM **18** (12) (1975).
[14] R.J. Lipton, L. Snyder and Y. Zalcstein, A comparative study of models of parallel computation, Conf. Rec. of Fifteenth Ann. IEEE SWAP Symp. (1974).
[15] S.S. Patil, Limitation and capabilities of Dijkstra's semaphore primitives for coordination among processes, Project MAC Computational Structure Group Memo **57** (1971).
[16] H. Vantiborgh and A. van Lamsweerde, On an extension of Dijkstra's semaphore primitives, Information Processing Lett. **1** (1972) 181–186.