ANNALES SOCIETATIS MATHEMATICAE POLONAE
Series IV: FUNDAMENTA INFORMATICAE IV.2 (1981)

ON PARTIAL LANGUAGES

Jan G r a b o w s k i

Humboldt University

Received October 24, 1979

AMS categories: 68C25

A b s t r a c t. The paper studies the behavior of control structures of algorithms designed for (partly) parallel execution. A generalization of Peterson's computation sequence set, the partial language, is discussed, which reflects the concurrency of events. In particular, the families of partial languages definable by Petri nets and by safe Petri nets are investigated with respect to closedness under certain operations. Trace languages (Mazurkiewicz) and path expressions (Campbell and Habermann) are included in the considerations.

K e y w o r d s: computation sequence set, partial word, partial language, concurrent processes, Petri net, path expression, trace language.

1. Introduction. This paper deals with algorithms designed for (partly) parallel execution. The analysis of such an algorithm (e.g., in order to decide correctness or equivalence) is in general an extremely difficult task.

For some purposes it may be sufficient to analyze merely the control structure underlying the algorithm.

In traditional (serial) algorithms the control structure can be understood as a (possibly infinite) directed graph whose vertices are called states and whose arcs, called transitions, are labelled with instructions. Here an instruction is regarded as being something indivisible and uninterpreted; hence it is fully adequate to assume instructions to be symbols of a finite alphabet. The set of label words corresponding to paths which lead from an initial state to a terminal state of the graph is called the computation sequence set [13] of the control structure, or simply the language defined by it. Which of the computation sequences belonging to the set is (or can be) really executed depends on the interpretation of the instructions and on the actual data to which the algorithm is applied.

If the number of states is finite, then the computation sequence set is a regular language. However, as it is often convenient to regard algorithmic devices as being capable of infinite memory, there is no imperative reason to restrict considerations to control structures with finite state sets.

Peterson [12] and Peterson and Bredt [14] investigate a variety of classes of control structures with possibly infinite state sets, such as petrinets, parallel program schemata, UCLA graphs etc., and compare them via the families of languages definable by them. For example, petrinets define a family of languages which properly contains the family of all regular languages. This family has been studied by Peterson [13], Hack [6], Starke [15], Greibach [5], and others.

However, the concept of petrinets, as well as the other control structure concepts appearing in Peterson's comparative study, are designed for modelling "parallel" (or "concurrent") systems of

algorithmic processes, i.e. systems in which events can occur concurrently, which means that they occur in no definite mutual ordering with respect to time. This property is not sufficiently reflected by the computation sequence set, for in a computation sequence the events (instruction executions) are forced into a linear ordering.

In discussing the next examples, as well as in considerable parts of the subsequent sections, we assume the reader's familiarity with petrinets. (See Peterson [11], Starke [18] for an explanation of the basic notions).

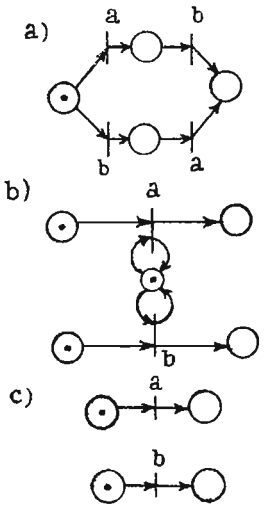


Fig. 1

Example 1. Figure 1 shows three petrinets with their initial markings. Assuming terminal markings suitably chosen, we see that they all define the same computation sequence set, namely {ab, ba}. It is, however, obvious that in the net of Figure 1c the events a and b can occur concurrently, while in the first two nets they cannot. This difference is not reflected in the computation sequence set (the language) of a petrinet.

To overcome this defect we shall introduce the concept of a partial word. For example, "parallel multiplication" of the two symbols a and b yields the partial word a × b. Thus the partial language defined by our three nets are {ab, ba}, {ab, ba}, and {a × b}, respectively.

Partial words seem to give a more adequate description of what a control structure can do. This becomes evident if we are interested in properties like determinacy. In our example the net of Figure 1c is determinate in that it can generate a single partial word only, while the first two nets are indeterminate.

In this article we shall introduce the concepts of partial word and partial language and formulate first results of partial language theory. The concepts are not really new; they appear more or less explicitly in Mazurkiewicz [10], Winkowski [17], Campbell and Habermann [2], Berzins and Kapur [1], but have not been systematically studied by those authors*.

Another notion very similar to that of a partial word is the notion of a process in Genrich [4].

Previous work on partial languages definable by Petri nets has been confined to safe (1-bounded) Petri nets, i.e. Petri's condition/event systems. In this paper we try to extend the concept to what is now usually called a Petri net (cf. Hack [7], Peterson [13]), or sometimes a place/transition system. It turns out that the basic notion ("p is firable in N at M") has no evident "canonical" generalization. Thus the natural ideas of the special case (the case of safe Petri nets) become ambiguous and doubtful when carried over to the general case (In our paper we start with the general case and then specialize to safe Petri nets).

---

*After completing this paper the author received a paper which does report a systematical study. (Winkowski [18]).

Thus, two families of partial languages will be considered in this paper: petrinet partial languages* and regular partial languages (partial languages of safe petrinets). They are investigated with respect to closedness under certain operations. Typical partial-language operations are: parallel product, parallel iteration, Y-section, and restriction. We shall give canonical representations of petrinet partial languages which correspond to known canonical representations of petrinet languages.

Related concepts such as trace languages (Mazurkiewicz [10]) and path expression languages [1,2] are included in our conside rations.

The reader should be aware of a phenomenon occurring, e.g., in Figure 1a . There are more than one transition labelled with the same symbol. We say that the transitions of the petrinet are renamed. The transitions labelled with identical symbols could be understood as calling the same operation. Hence there may occur** partial words in which a symbol is self-concurrent, which means that it occurs at two positions which are incomparable (with respect to time, causality or what-ever you think of). The two occurrences then represent two control events calling the same operation. (In unrenamed and safe petrinets such a situation is impossible). If you decide that such a situation is undesirable, you have to solve a "mutual exclusion problem". (That's why we need partial languages: they enable us to give such problems a precise meaning).

Since partial words are more difficult to handle than words, our considerations will sometimes contain unpleasant technical difficulties. The author hopes that further experience in using the concept will help to clear the underwood.

---

*Constructions like this led the author to write "petrinet", instead of "Petri net", henceforth.

**(Though not in our example).

I am grateful to Peter Starke, who introduced me to Petri net theory, and to the referee, who helped me to revise and, hopefully, to improve the paper.

Some terminology

By $\underline{N}$ we denote the set of all natural numbers (including zero), by $\underline{N}^k$ its k-fold cartesian power.

A partial ordering on a set is an irreflexive and transitive binary relation; if $<$ is such a relation it should be clear what is meant by $a<b$, $a>b$, $a \leqq b$, and $a \geqq b$. $<$ is said to be an ordering iff for any two distinct elements a and b we have $a<b$ or $b>a$. $a <\cdot b$ means that b is an immediate successor of a under $<$.

The transitive closure of a binary relation R is denoted by $\underline{Tr}(R)$. We shall frequently use the fact that if R is contained in a partial ordering then $\underline{Tr}(R)$ is a partial ordering.

Mappings from a set A to a set B will be regarded as subsets of the cartesian product $A \times B$; this explains what is meant by the union of two mappings. There will appear only such mappings f from A to B that for each $a \in A$ there exists a unique $b \in B$ with $(a,b) \in f$. We shall use the abbreviation $f: A \rightarrow B$.

The restriction of a mapping $f: A \rightarrow B$ to a subdomain $A' \subseteq A$ is denoted by $f|_{A'}$. Similarly $R|_{A'}$ is the restriction of the binary relation R to the set A'.

The sign $:=$ $(:\leftrightarrow)$ is used in definitions. It sets the left-hand side equal (equivalent) to the right-hand side.

The end of a proof is denoted by a square ■.

2. Partial words and partial languages

Basic definitions

Definitions. A labelled partially ordered set over the alphabet X is a triple $(A,<,\beta)$ where A is a finite set, $<$ is a partial ordering on A, and β: $A \to X$.

Two labelled partially ordered sets $(A,<,\beta)$ and $(A',<',\beta')$ over X are said to be isomorphic iff there is an order isomorphism f: $(A,<) \to (A',<')$ such that $\beta' f = \beta$.

A partial word over X is an isomorphy class of labelled partially ordered sets over X. The class containing $(A,<,\beta)$ is denoted by $[(A,<,\beta)]$.

The set of all partial words over X is denoted by $\underline{PW}(X)$. A partial language over X is a subset of $\underline{PW}(X)$.

Words are special partial words whose underlying partial ordering is an ordering. As usual, we regard symbols (elements of X) as special words. Thus we have $X \subset \underline{W}(X) \subset \underline{PW}(X)$, where $\underline{W}(X)$ denotes the set of all words over X. A language over X is a subset of $\underline{W}(X)$. The empty word (isomorphy class of the empty set) is denoted by $\underline{e}$.

A partial word $[(A,<,\beta)]$ is said to be pure if β is a one-to-one mapping.

The above definition, as well as many of the definitions to follow, use a representative of an isomorphy class. The necessary proof that it is independent of the choice of the representative is always left to the reader.

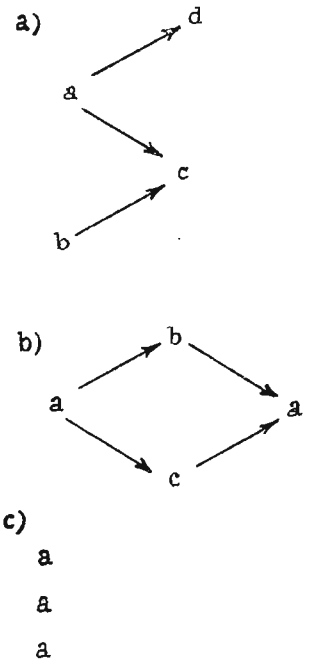


Fig. 2

Example 2. If convenient, we represent a partial word by a graph with labelled vertices. The meaning is obvious. Figure 2 shows three examples of partial words. The first of them is pure (assuming that a,b,c,d be pairwise distinct).

Double monoids

We shall consider a variety of operations on partial words and partial languages. Let us start with two "multiplications".

Definitions. The serial product of two partial words is defined by

$$[(A,<,\beta)] * [(A',<',\beta')] := [(A \cup A', <'', \beta \cup \beta')]$$

where $A \cap A' = \emptyset$, and

$$a <'' b :\leftrightarrow a < b \text{ or } a <' b \text{ or } a \in A \text{ and } b \in A'.$$

The parallel product of two partial words is defined by

$$[(A,<,\beta)] \times [(A',<',\beta')] := [(A \cup A', < \cup <', \beta \cup \beta')]$$

where $A \cap A' = \emptyset$

The star $*$ for serial multiplication will normally be omitted.

The operations and are extended to partial languages:

$L_1 * L_2 := \{p_1 * p_2 : p_1 \in L_1 \text{ and } p_2 \in L_2\}$

$L_1 \times L_2 := \{p_1 \times p_2 : p_1 \in L_1 \text{ and } p_2 \in L_2\}$.

Definition. A double monoid is an algebra $(M, *, \times, \underline{e})$ with two binary operations $*$ and $\times$ and one 0-ary operation $\underline{e}$, which fulfils the axioms

$$p_1 * (p_2 * p_3) = (p_1 * p_2) * p_3$$

$$p * \underline{e} = p$$

$$\underline{e} * p = p$$

$$p_1 \times (p_2 \times p_3) = (p_1 \times p_2) \times p_3$$

$$p_1 \times p_2 = p_2 \times p_1$$

$$p \times \underline{e} = p.$$

Proposition 1. Under the previously defined operations the set $\underline{PW}(X)$ forms a double monoid.

P r o o f. Immediately from the definitions.

While every word can be built up from symbols by application of $*$ and $\underline{e}$, the three operations $*$, $\times$, and $\underline{e}$ are not sufficient to construct every partial word from symbols. In Figure 2 the second and third partial words have representations $a*(b\times c)*a$ abd $a\times a\times a$, respectively, while the first one cannot be built up in this way.

Definition. Let $\underline{M}(X)$ denote the double monoid generated by the set X within the double monoid $(\underline{PW}(X), *, \times, \underline{e})$.

Proposition 2. $\underline{M}(X)$ is the free double monoid generated by X.

P r o o f. We shall merely outline the proof.

It should be clear what is the difference between an element of $\underline{M}(X)$ and its representation by an expression built of symbols of X and functional symbols $*$, $\times$, $\underline{e}$. For example, the two expressions $x\times y$ and $y\times x$ represent the same element of $\underline{M}(X)$. What we have to show is that if two expressions represent the same element of $\underline{M}(X)$, they can be transformed into each other via application of the axioms.

A partial word is said to be $*$-reducible ($\times$-reducible) iff it has a representation $p = p_1 * p_2$ ($p = p_1 \times p_2$) as product of non-empty partial words.

The proof proceeds in the following steps:

(1) It is shown that a partial word cannot be both $*$-reducible and $\times$-reducible.

(2) Let $p \in \underline{M}(X)$ be represented by the expression T. By sole application of the axioms T can be transformed into an expression $T_1 * T_2 * \ldots * T_n$ (if p is $*$-reducible) or $T_1 \times T_2 \times \ldots \times T_n$ (if p is $\times$-reducible) consisting of $*$-irreducible, resp. $\times$-irreducible, factors and representing again p. This is shown by induction on the "height" of T.

(3) If $T_1 * \ldots * T_n$ and $T_1' * \ldots * T_{n'}'$ are two such "irreducible representations" of the same partial word, then $n = n'$, and T_i and T_i' represent the same partial word.

Similarly, if $T_1 \times \ldots \times T_n$ and $T_1' \times \ldots \times T_{n'}'$ are two "irreducible representations" of the same partial word, then $n = n'$, and the expressions $T_1, \ldots, T_n$ represent the same partial words as $T_1', \ldots, T_n'$ up to a permutation.

From the observations (1)-(3) the proposition is easily derived. ∎

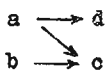
As we have observed above, $\underline{M}(X)$ is properly contained in $\underline{PW}(X)$. How can we decide whether a given partial word $[(A,<,\beta)]$ is in $\underline{M}(X)$? It is sufficient to do this for pure partial words, for if $\varepsilon: A \to A$ is the identical mapping, then $[(A,<,\beta)]$ is in $\underline{M}(X)$ iff $[(A,<,\varepsilon)]$ is in $\underline{M}(A)$. The following proposition answers the question.

Proposition 3. Let $[(A,<,\beta)]$ be a pure partial word. Without loss of generality we may assume that $\beta = \varepsilon: A \to A$, the identical mapping. The following conditions are equivalent:

(i) $[(A,<,\varepsilon)] \in \underline{M}(A)$.

(ii) If $a,b,c,d \in A$, $a < c$, $a < d$, $b < c$, $a \not\leq b$, $c \not\leq d$, then $b < d$.

The situation described by condition (ii) may be visualized by a picture



showing the assumed ordering between a,b,c, and d. If such a picture occurs then (assumed that condition (ii) holds) we know that $b < d$.

P r o o f.

(I) (i) implies (ii).

The empty word trivially fulfils our condition (ii). The same holds for each symbol (element of A). It remains to show that condition (ii) is invariant under the operations $*$ and $\times$.

Assume that $p_i = [(A_i, <_i, \varepsilon_i)]$ fulfil the condition (ii), and that $A_1 \cap A_2 = \emptyset$. Let $p_1\ p_2 = [(A_1 \cup A_2, <, \varepsilon)]$. Assume a situation

$$\begin{array}{ccc} a & \longrightarrow & d \\ & \searrow & \\ b & \longrightarrow & c \end{array}$$

to be given and the conclusion $b < d$ not to hold. Then $b \notin A_1$ or $d \notin A_2$. The cases are symmetric to each other; we consider the case $b \notin A_1$ only. Thus $b \in A_2$. But then $c \in A_2$ and $a \in A_2$. Since p_2 fulfils condition (ii), we have $b \in d$.

The proof for the parallel product is similar.

(II) (ii) implies (i).

We have to show that: If a pure partial word p fulfils condition (ii), then it is empty or a symbol, or it admits a representation $p = p_1 * p_2$ or $p = p_1 \times p_2$, where p_1 and p_2 are non-empty. Since condition (ii) is hereditary to the subwords p_1 and p_2, this is sufficient.

Suppose that p is not empty and not a symbol. Let $p = [(A, <, \varepsilon)]$. Let B be the set of all minimal (under $<$) elements of A. If B is a singleton, we have $p = [(B, <|_B, \varepsilon|_B)] * [(C, <|_C, \varepsilon|_C)]$, $C := A \setminus B$, and we are done. Suppose now that B contain more than one element.

C a s e A. Let there exist a partition $B = B_1 \cup B_2$ into two non-empty subsets such that no element of B_1 has an upper bound (with respect to $<$) in common with any element of B_2. Then we set

$$A_1 := \{a : \text{there is a } b_1 \in B_1 \text{ with } b_1 \leqq a\},$$

$$A_2 := \{a : \text{there is a } b_2 \in B_2 \text{ with } b_2 \leqq a\},$$

and we have $p = [(A_1, <|_{A_1}, \varepsilon|_{A_2})] \times [(A_2, <|_{A_2}, \varepsilon|_{A_2})]$.

C a s e B. Assume that no such partition exists. For any two elements of B we take their minimal common upper bounds under $<$, if such common upper bounds exist. These upper bounds form a set C_0. Choose the maximal elements of C_0, obtaining a subset C_1. Let D be the subset of all elements of A which are not comparable under $<$ with any element of C_1. Add the minimal elements of D to C_1, obtaining a totally unordered subset C_2 of A. Each element of A is comparable with some element of C_2. We also know that B and C_2 are disjoint.

(1) Let $b \in B$ and $c \in C_2$. Then $b < c$.

By definition of B there exists a $b' \in B$ with $b' < c$. Assume that $b \neq b'$. There exists a sequence $b = b_1, \ldots, b_m = b'$ of elements of B with the property that b_i and b_{i+1} have a common upper bound c_i. c_i can be chosen from C_1. We assume that the sequence is of minimal length. If $m > 2$ then we have a configuration

$$\begin{array}{ccc} b_{m-1} & \longrightarrow & c_{m-2} \\ & \searrow & \\ b_m & \longrightarrow & c_{m-1} \end{array}$$

By condition (ii) this implies $b_m < c_{m-2}$. Thus b_m and b_{m-2} already have a common upper bound, which contradicts the assumed minimality of m. The conclusion is that $m \leqq 2$, and b has an upper bound in common with b', namely c_1. If $c = c_1$ then we are done. If $c \neq c_1$ then c is incomparable with c_1. Applying condition (ii) to the configuration

$$\begin{array}{ccc} b' & \longrightarrow & c \\ & \searrow & \\ b & \longrightarrow & c_1 \end{array}$$

yields $b < c$.

Now we define another set E:

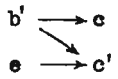
$E := \{e : e \in A$ and e has an immediate successor in $C_2\}$.

(2) Let $e \in E$ and $c \in C_2$. Then $e < c$.

e has an immediate successor c' in C_2, $e < c'$. Assume that $c \neq c'$.

C a s e (A) $c' \in C_1$

By definition of C_1 we may assume that there exist two elements b, b' of B such that c' is a common upper bound to them while e is not. Hence $b \not\leqq e$ or $b' \not\leqq e$. Assume that $b' \not\leqq e$. Applying condition (ii) to the configuration



we obtain $e < c$.

C a s e (B) $c' \in C_2 \setminus C_1$.

c' is a minimal element of D. Hence $e \notin D$, which implies the existence of a $c'' \in C_1$ with $e < c''$. We have reduced Case B to Case A.

Now we define

$A_1 := \{a : a \in A$ and there is a $c \in C_2$ with $a < c\}$

$A_2 := A \setminus A_1$.

A_1 is non-empty since $B \subseteq A_1$. A_2 is non-empty since $C_2 \subseteq A_2$. The set of minimal elements of A_2 is C_2.

(3) If $a_1 \in A_1$ and $a_2 \in A_2$ then $a_1 < a_2$. - If $a_1 \in A_1$ then $a_1 < c$ for a certain $c \in C_2$. Hence $a_1 \leqq e$ for a certain $e \in E$. This implies $a_1 < c$ for all $c \in C_2$. On the other hand we have $c \leqq a_2$ for some $c \in C_2$. This implies $a_1 < a_2$.

We have found a partition $p = [(A_1, <|_{A_1}, \varepsilon|_{A_1})] * [(A_2, <|_{A_2}, \varepsilon|_{A_2})]$ This completes the proof. ∎

Smoothing, weakening and strengthening of partial languages

Definitions. Let p and q be partial words. We say that q is smoother than p if they obey representations $p = [(A, <, \beta)]$ and $q = [(A, <', \beta)]$ such that $<$ is a subset of $<'$.

$\underline{We}(p) := \{q : q \text{ is smoother than } p\}$ is called the weakening of p. It is a finite partial language containing p.

$\underline{Sm}(p) := \underline{We}(p) \cap \underline{W}(X)$ is called the smoothing of p. (We suppose that $p \in \underline{PW}(X)$). This is a finite non-empty language.

Let $L \subseteq \underline{PW}(X)$.

$\underline{We}(L) := \bigcup\{\underline{We}(p) : p \in L\}$ is called the weakening of L.

$\underline{Sm}(L) := \underline{We}(L) \cap \underline{W}(X)$ is called the smoothing of L.

$\underline{St}(L) := \{p : p \in L \text{ and } p \text{ is not smoother than any other } q \in L\}$ is called the strengthening of L.

L is called weak if $\underline{We}(L) = L$, and strong if $\underline{St}(L) = L$.

From these definitions many exercises for the reader result, e.g. the equalities $\underline{We}(\underline{St}(L)) = \underline{We}(L)$, $\underline{Sm}(St(L)) = \underline{Sm}(L)$, $\underline{St}(\underline{We}(L)) = \underline{St}(L)$, or the equivalence of the two conditions "L is a language" and "L is weak and strong".

Application of the operation $\underline{Sm}$ projects notions and facts of partial language theory to those of language theory. It is this kind of connection which will guide us in our following considerations. An example of such a projection is the shuffle product:

Definition. Let $w, w' \in \underline{W}(X)$. The shuffle product of w and w' is $w \parallel w' := \{u_1v_1u_2v_2 \ldots u_mv_m : m \in \underline{N},\ u_1 \ldots u_m = w,\ v_1 \ldots v_m = w'\}$, a finite language.

Let $L, L' \subseteq \underline{W}(X)$. The shuffle product of L and L' is $\bigcup\{w \parallel w' : w \in L,\ w' \in L'\}$, denoted by $L \parallel L'$. (Cf. Eilenberg [3])

Proposition 4. $\underline{Sm}(L_1 * L_2) = \underline{Sm}(L_1) * \underline{Sm}(L_2)$,
$\underline{Sm}(L_1 \times L_2) = \underline{Sm}(L_1) \parallel \underline{Sm}(L_2)$.

P r o o f. Trivial. ■

Serial and parallel iteration

Definitions. Let L be a partial language.

$L^* := \bigcup \{M_n : n \in \underline{N}$, with $M_0 := \{\underline{e}\}$, $M_{n+1} := M_n * L$.

$L^\times := \bigcup \{N_n : n \in \underline{N}$, with $N_0 := \{\underline{e}\}$, $N_{n+1} := N_n \times L$.

If L is a language then we set

$L'' := \bigcup \{P_n : n \in \underline{N}\}$, with $P_0 := \{\underline{e}\}$, $P_{n+1} := P_n \| L$.

L^* ($L^\times$; L'') is valled the serial (parallel; shuffle) iteration of L.

The serial iteration is known from language theory; for languages it coincides with Kleene's star operation.

Proposition 5. $\underline{Sm}(L^*) = (\underline{Sm}(L))^*$,

$\underline{Sm}(L^\times) = (\underline{Sm}(L))''$, for any partial language L.

P r o o f. Immediately from Proposition 4. ■

Example 3. Suppose that $L := \{ab\}$ with $a \neq b$. L^* consists of all words of the form $\underbrace{abab...ab}_{n \text{ times}}$, with $n \in \underline{N}$.

$L^\times$ consists of all partial words of the form $\underbrace{ab \times ab \times ... \times ab}_{n \text{ times}}$, with $n \in \underline{N}$.

L'' is the so-called Dyck set over $\{a,b\}$. It consists of all words $p \in \{a,b\}^*$ such that $\#_a(p) = \#_b(p)$ and $\#_a(r) \geq \#_b(r)$ for any initial segment r of p. ($\#_x(w)$ means "number of occurrences of x in w"). From this example we learn that shuffle iteration does not preserve regularity.

The Y-section of partial languages

Let us now turn to the most typical operation with partial languages. Look at the two partial words depicted in Figure 3. Although they are not identical, they have something in common. They do not describe the same execution of a control structure

but, in a sense, consistent executions, in that their smoothings are not disjoint: $\underline{Sm}(u) \cap \underline{Sm}(v) = \{abcabd\}$.

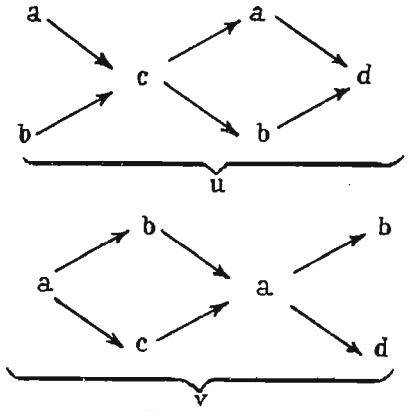


Fig. 3

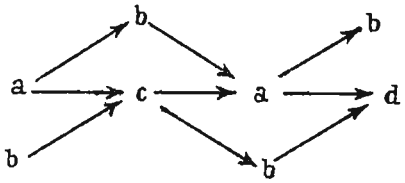


Fig. 4

An adequate operation for describing the "degree" of consistency between partial words or partial languages is the Y-section.

Definition. Let $(A,<,\beta)$ and $(A_i,<_i,\beta_i)$ $(i=1,2)$ be labelled partially ordered sets over X and $Y \subseteq X$. We say that $(A,<,\beta)$ is the Y-composition of $(A_1,<_1,\beta_1)$ and $(A_2,<_2,\beta_2)$ iff

(i) $A = A_1 \cup A_2$,

(ii) $A_1 \cap A_2 = \beta^{-1}(Y)$,

(iii) $\beta_i = \beta|_{A_i}$ $(i=1,2)$, and

(iv) $< = \underline{Tr}(<_1 \cup <_2)$.

Definition. Let p_1, p_2, and p be partial words. We define

$p \in p_1 \overset{\frown}{Y} p_2$

: $\longleftrightarrow$ there exist representatives

$(A_1, <_1, \beta_1)$ of p_1,

$(A_2, <_2, \beta_2)$ of p_2,

$(A, <, \beta)$ of p

such that $(A, <, \beta)$ is the Y-composition of $(A_1, <_1, \beta_1)$ and $(A_2, <_2, \beta_2)$.

$p_1 \overset{\frown}{Y} p_2$ is called the Y-section of p_1 and p_2.

The Y-section of partial languages is defined by

$$L_1 \overset{\frown}{Y} L_2 := \bigcup \{ p_1 \overset{\frown}{Y} p_2 : p_1 \in L_1, \ p_2 \in L_2 \}.$$

The Y-section of two partial words is always a finite partial language. The $\{a,b,c,d\}$-section of the two partial words shown in Figure 3 is $\{abcabd\}$, while their $\{a,c,d\}$-section consists of one partial word shown in Figure 4. Observe how the partial words belonging to $p_1 \overset{\frown}{Y} p_2$ are formed: p_1 and p_2 are stuck together at points carrying symbols from Y.

It is easily established that for languages over X the X-section coincides with the intersection. This is generally not true with partial languages:

Example 4. Let $u := b \times ab$. Then $u \underset{a,b}{\frown} u = \{b \times ab, \ a(b \times b)\}$.

Note that the $\emptyset$-section always coincides with the parallel product.

Proposition 6. For any two partial languages L_1 and L_2 over X:

$\underline{We}(L_1 \underset{X}{\frown} L_2) = \underline{We}(L_1) \cap \underline{We}(L_2)$,

$\underline{Sm}(L_1 \underset{X}{\frown} L_2) = \underline{Sm}(L_1) \cap \underline{Sm}(L_2)$.

Consequently the three conditions

$\underline{We}(L_1) \cap \underline{We}(L_2) \neq \emptyset$,

$\underline{Sm}(L_1) \cap \underline{Sm}(L_2) \neq \emptyset$, and

$L_1 \underset{X}{\cap} L_2 \neq \emptyset$

are equivalent.

P r o o f. We have to prove the first equality only, and it is sufficient to do this for partial words.

(I) $\underline{We}(p_1 \underset{X}{\cap} p_2) \subseteq \underline{We}(p_1) \cap \underline{We}(p_2)$.

Let p be in the X-section of p_1 and p_2 and let q be smoother than p. By the definition of the Y-section, relativized to the case Y = X, this means that $p = [(A,<,\beta)]$, $p_i = [(A,<_i,\beta)]$ (i=1,2), and $q = [(A,<',\beta)]$, with $\underline{Tr}(<_1 \cup <_2) = < \subseteq <'$. Hence $<_i \subseteq <'$, and p_i is smoother than q (i=1,2).

(II) $\underline{We}(p_1) \cap \underline{We}(p_2) \subseteq \underline{We}(p_1 \underset{X}{\cap} p_2)$.

Let q be weaker than p_1 and p_2. We can choose A and β such that, together with partial orderings $<'$, $<_1$, and $<_2$ on A, they specify representatives for q, p_1, and p_2, and such that $<_1$ and $<_2$ are subsets of $<'$. But then $\underline{Tr}\ (<_1 \cup <_2) \subseteq <'$. The partial word $p = [(A, \underline{Tr}(<_1 \cup <_2), \beta)]$ belongs to $p_1 \underset{X}{\cap} p_2$, and q is smoother than p. ■

Proposition 7. The operation of Y-section is commutative and associative, for any fixed Y.

P r o o f. The commutativity follows from the obvious symmetry in the definition.

To prove associativity, it is sufficient to show that $p \in p_1 \underset{Y}{\cap} (p_2 \underset{Y}{\cap} p_3)$ is equivalent to the existence of representations $p = [(A,<,\beta)]$ and $p_i = [(A_i,<_i,\beta_i)]$ such that

(i') $A_1 \cup A_2 \cup A_3 = A$,

(ii') $A_1 \cap A_2 = A_2 \cap A_3 = A_1 \cap A_3 = \beta^{-1}(Y)$,

(iii') $\beta|_{A_i} = \beta_i$ (i=1,2,3),

(iv') $< = \underline{Tr}(<_1 \cup <_2 \cup <_3)$.

The proof of this statement is a straightforward application of conditions (i)-(iv) above and is left to the reader. ■

The operation of Y-section reflects what is meant by the synthesis of a complex control structure from simpler components. Consider, for example, Hoare's concept of "Communicating Sequential Processes" (Hoare [8]). There we have a variety of algorithms $A_1,\ldots,A_n$ operating on individual memories and communicating with each other via distinguished communication statements. A_i can pass a message to A_j if and only if both algorithms are ready to execute the appropriate communication statements. If we forget about the concrete operations executed by the algorithms and consider merely their control structures, we may assume that A_i's control structure defines a partial language L_i over an alphabet X_i. X_i contains names of the communication statements belonging to A_i. Let c_{ij} be the name for "communication from A_i to A_j", and let C be the set of all these symbols. Then it seems reasonable to require that the partial language of the whole variety be $L_1 \underset{C}{\cap} L_2 \underset{C}{\cap} \ldots \underset{C}{\cap} L_n$.

Conditions of unambiguity

Example 4 shows a Y-section of partial words which consists of more than one partial word. In this example the way of sticking the two components together is "ambiguous" in that it is not uniquely determined by the components. This ambiguity, in the general case, can give rise to nondeterminacy of a composite control structure even if its components are determinate. (We still leave the concept of determinacy to the reader's intuition). Let us look for conditions under which such an ambiguity cannot occur.

Definition. We associate with each partial word $p = [(A,<,\beta)]$ its automorphisms. An automorphism is a permutation of A which is an isomorphism of labelled partially ordered sets representing p. An automorphism is said to be trivial on Y if it has all elements of $\beta^{-1}(Y)$ as fixed points. The whole automorphism group of p is said to be trivial on Y if every automorphism is trivial on Y.

Proposition 8. If the automorphism groups of p_1 and p_2 are trivial on Y then $p_1 \underset{Y}{\frown} p_2$ consists at most of one partial word p.

P r o o f. Assume that $p = [(A,<,\beta)]$, $p' = [(A,<',\beta)]$, $p_i = [(A_i,<_i,\beta_i)] = [(A_i,<_i',\beta_i)]$, and that $(A,<,\beta)$ is the Y-composition of $(A_1,<_1,\beta_1)$ and $(A_2,<_2,\beta_2)$, while $(A,<',\beta)$ is the Y-composition of $(A_1,<_1',\beta_1)$ and $(A_2,<_2',\beta_2)$. We have to show that $p=p'$.

There exists an automorphism f_i of p_i which maps $(A_i,<_i,\beta_i)$ on $(A_i,<_i',\beta_i)$. By our assumption this automorphism is trivial on Y. We may form the union $f = f_1 \cup f_2$. This is an isomorphism between $(A,<,\beta)$ and $(A,<',\beta)$. Hence $p = p'$. ∎

Call two symbols x,y concurrent in $p = [(A,<,\beta)]$ iff there exist two incomparable (under $<$) elements a,b of A such that $\beta(a)=x$ and $\beta(b)=y$. If x is concurrent with x then x is said to be self-concurrent.

The following is easily verified by the reader: If p has no self-concurrent symbol belonging to the alphabet Y then the automorphism group of p is trivial on Y. The absence of self-concurrency in the components is thus a sufficient condition for the Y-section to be unambiguous. In the partial word u of Example 4 the symbol b is self-concurrent.

Restriction of partial languages

The following operation is a generalization of Hack's and Starke's "restriction" of languages, the latter being referred to

as smooth restriction in this paper. We prefer the use the authorized word "restriction" although it not very suggestive).Restriction expresses that a control structure, when joined with other control structures, is "indifferent" to exactly those symbols which do not occur in any partial word generated by it.The notion is mainly of technical importance.

Definition. The alphabet of a partial language $L \subseteq \underline{PW}(X)$ is the set of symbols which occur in at least one number of L. It is a subset of X.

Definition. Let L_1 and L_2 be partial languages whose alphabets are X_1 and X_2, respectively.

$$L_1 ! L_2 := L_1 \overset{\frown}{\underset{X_1 \cap X_2}{}} L_2$$

is called the restriction of L_1 and L_2.

In contrast to the operations introduced heretofore, restriction cannot be defined elementwise.

Example 5. Set

$$u := a \begin{array}{l} \nearrow b \\ \searrow c \end{array}, \quad v := \begin{array}{c} a \\ c \rightarrow d \end{array}.$$

Then

$$\{u\} ! \{v\} = \left\{ a \begin{array}{l} \nearrow b \\ \searrow c \rightarrow d \end{array} \right\}, \text{ while } \{u,d\} ! \{v\} = \emptyset.$$

Proposition 9. Let L_1 and L_2 be two partial languages whose alphabets are X_1 and X_2, respectively. Let $X_1 \cup X_2 \subseteq X$. Then

(1) $L_1 ! L_2 = (L_1 \times (X_2 \setminus X_1)^X) \underset{X}{\cap} (L_2 \times (X_1 \setminus X_2)^X)$, and

(2) $L_1 ! L_2 = (L_1 \times (X \setminus X_1)^X) \underset{X}{\cap} (L_2 \times (X \setminus X_2)^X) \underset{X}{\cap} (X_1 \cup X_2)^X$.

Proof. Set $Y := X_1 \cap X_2$.

(1) The inclusion from left to right is shown as follows:

Assume that $p \in p_1 \underset{Y}{\cap} p_2$, $p_i \in L_i$. Choose representations $p = [(A,<,\beta)]$, $p_i = [(A_i,<_i,\beta_i)]$ such that $(A,<,\beta)$ is the Y-composition of $(A_1,<_1,\beta_1)$ and $(A_2,<_2,\beta_2)$. Set $r_i := [(A\setminus A_i, \emptyset, \beta|_{A\setminus A_i})]$. Then $r_i \in (X_{3-i} \setminus X_i)^X$, and $p_i \times r_i \in L_i \times (X_{3-i} \setminus X_i)^X$. From $p_i \times r_i = [(A,<_i,\beta)]$ and $\underline{Tr}(<_1 \cup <_2) = <$ we conclude $p \in (p_1 \times r_1) \underset{X}{\cap} (p_2 \times r_2)$.

The inclusion from right to left is shown in a completely analogous way.

(2) Let L' (L'') be the right-hand side of (1), (2). Then $L'' = (L' \times (X \setminus (X_1 \cup X_2))^X) \underset{X}{\cap} (X_1 \cup X_2)^X$. Since $X_1 \cup X_2$ is the alphabet of L', from this equality one easily concludes that $L' = L''$. ■

The above proposition allows us to give a formula for the smooth version of restriction.

Definition. Given languages L_1 and L_2 whose alphabets are X_1 and X_2, we define

$$L_1 \circledR L_2 := (L_1 \parallel (X_2 \setminus X_1)^*) \cap (L_2 \parallel (X_1 \setminus X_2)^*),$$

and call this operation smooth restriction.

Smooth restriction appeared first in Hack's and Starke's theorems on canonical forms of free petrinet languages, cf. Starke [1].

Proposition 10. $\underline{Sm}(L_1 \,!\, L_2) = \underline{Sm}(L_1) \circledR \underline{Sm}(L_2)$.

P r o o f. Immediately from Propositions 4, 5, 6, and 8. ■

Weak and strong restriction

Definitions. Suppose $L_1, L_2 = \underline{PW}(X)$.

$L_1 \,?\, L_2 := \underline{We}(L_1 \,!\, L_2)$ is called weak restriction,

$L_1 \,!!\, L_2 := \underline{St}(L_1 \,!\, L_2)$ is called strong restriction of L_1 and L_2.

The reason why we introduce these operations is that the operation of restriction does in general preserve neither weakness nor strength.

Proposition 11. Restriction, weak restriction, and strong restriction are commutative and associative operations.

P r o o f. The commutativity follows from that of the Y-section.

Let us prove the associative law.

(I) Restriction is associative.

From formula (2) of Proposition 9 one derives by induction the formula

$$L_1 \,!\, (L_2 \,!\, (\ldots !\, L_n)..)) = \bigcap_X \{(L_i \times (X \setminus X_i)^X) \underset{i=1,\ldots,n}{\cap_X} (X_1 \cup \ldots \cup X_n)^X.$$

This formula implies the associative law.

(II) Weak restriction is associative.

We prove an auxiliary equality

$$\underline{We}(L_1 \,!\, L_2) = \underline{We}(L_1 \,!\, \underline{We}(L_2)).$$

The inclusion from left to right is trivial since $\underline{We}$ is a closure operation preserving the alphabet of a partial language. For the converse inclusion let $q = [(A,<',\beta)]$ be in $p_1 \underset{Y}{\cap} q_2$, where $p_1 \in L_1$, $q_2 \in \underline{We}(L_2)$, and Y is the intersection of the alphabets of L_1 and L_2. p_1 and q_2 have representatives $(A_1,<_1,\beta_1)$ and $(A_2,<'_2,\beta_2)$ whose Y-composition is $(A,<',\beta)$. Especially we have $<' = \underline{Tr}(<_1 \cup <'_2)$. q_2 is smoother than a certain $p_2 = [(A_2, <_2, \beta_2)] \in L_2$, with $<_2 \subseteq <'_2$. Then $< := \underline{Tr}(<_1 \cup <_2) \subseteq <'$, and q is smoother than $[(A,<,\beta)] \in p_1 \underset{Y}{\cap} p_2 \subseteq L_1 \,!\, L_2$. Hence $q \in \underline{We}(L_1 ! L_2)$. From our auxiliary equality the associative law directly follows.

(III) Strong restriction is associative.

Again the proof proceeds via an auxiliary equality

$$\underline{St}(L_1 \,!\, L_2) = \underline{St}(L_1 \,!\, \underline{St}(L_2)).$$

The proof of this equality is a simple application of the definitions; we leave it to the reader. The associative law is an evident consequence. ■

Renamings

In the subsequent sections it will be useful to study the effect of renamings on partial languages.

Definition. A renaming is a mapping

$h: X \to Y \cup \{\underline{e}\}$.

By

$$h([(A,<,\beta)]) := [(B,<|_B,\ h\beta|_B)]$$

with

$$B := \{a : a \in A,\ h\beta(a) \neq \underline{e}\}$$

the renaming h is extended to a mapping $h: \underline{PW}(X) \to \underline{PW}(Y)$. Words are mapped to words; this is the usual extension of a renaming to a length-decreasing monoid homomorphism $\underline{W}(X) \to \underline{W}(Y)$. A renaming is said to be $\underline{e}$-free if $h(X) \subseteq Y$. If $L \subseteq \underline{PW}(X)$, we define $h(L) := \{h(p) : p \in L\}$. If $L' \subseteq \underline{PW}(Y)$ then $h^{-1}(L') := \{p : p \in \underline{PW}(X) \text{ and } h(p) \in L'\}$. The latter operation on partial languages is called inverse renaming.

Constructing partial words from symbols

Let us return to the question how partial words can be constructed of symbols. We saw that parallel and serial product are not powerful enough for this purpose.

Proposition 12. Any (non-empty) partial word can be constructed of symbols by use of the following operations: serial and parallel product, restriction, and ($\underline{e}$-free) renaming*).

*As the referee pointed out, the list of operations is redundant. Parallel product can be omitted since it can be generated from $\underline{e}$-free renaming and restriction. Serial product can be replaced by some extension of renaming.

P r o o f. Any (non-empty) partial word is obtained from a non-empty pure partial word by an($\underline{e}$-free) renaming. Without loss of generality the pure partial word is assumed to be $p = [(A,<,\varepsilon)]$, where ε is the identical mapping on A. Set $q := [(A,\emptyset,\varepsilon)]$. q is the parallel product of the symbols occurring in p. p is the restriction of q and all words aa', where $a,a' \in A$ and $a < a'$. ■

Unlike the characterization of $\underline{W}(X)$ as the free monoid generated by X, this method of building partial words in $\underline{PW}(X)$ generally uses symbols not belonging to X.

Weakness and renamings

The following lemma will be used in the next section.

Proposition 13. Application of a renaming to a partial language preserves weakness.

P r o o f. Assume that $L = \underline{We}(L) \subseteq \underline{PW}(T)$ and $h : T \to X \cup \{\underline{e}\}$ is a renaming. We have to show that $\underline{We}(h(L)) = h(L)$.

Suppose that $p = [(A,<,\beta)] \in L$ and $p' = h(p)$ and q' smoother than p'. We have to show that $q' \in h(L)$. Assume that $p' = [(A',<',\beta')]$, where $A' = \{a : a \in A \text{ and } h\beta(a) \neq \underline{e}\}$, $<' = <|_{A'}$, $\beta' = h\beta|_{A'}$. Let $q' = [(A',\precsim',\beta')]$, and let $\precsim'$ contain $<'$. We define $\precsim := \underline{Tr}(< \cup \precsim')$, a binary relation on A.

We show that $\precsim|_{A'} = \precsim'$. Assume that $c \precsim d$ for two elements of A'. By definition of $\precsim$ this implies

$$c = c_0 \mathrel{\tilde{\leqq}}' d_0 < c_1 \precsim' d_1 \dots d_{n-1} < c_n \mathrel{\tilde{\leqq}}' d_n = d.$$

Since $c_i \precsim' d_i$, we have $c_i, d_i \in A'$ $(i=1,\dots,n-1)$. Consequently $d_i <' c_{i+1}$ $(i=0,\dots,n-1)$, and $d_i \precsim' c_{i+1}$. Thus we have $c \precsim' d$, and this means $\precsim|_{A'} \subseteq \precsim'$. The converse inclusion is trivial.

Now we show that $\precsim$ is irreflexive. Assume that $a \precsim a$ for a certain $a \in A$. Since $a \not< a$, there must be $b \in A'$ such that $a \mathrel{\tilde{\leqq}} b \precsim a$. Hence $b \precsim b$, and $b \precsim' b$. This contradicts the irreflexivity of $\precsim'$.

Now we set $q := [(A, \precsim, \beta)]$. q is smoother than p since $\precsim$ contains $\prec$. Since L is weak we have $q \in L$. Finally we see that $h(q) = [(A', \precsim|_{A'}, \beta')] = q'$. ■

3. Weak partial languages of petrinets

In this section we shall study families of partial languages which are definable by petrinets. The petrinet concept is one of the widely-used models of control structures involving concurrency. We assume the reader to be familiar with this concept. Nevertheless we repeat some basic definitions, thereby introducing the notations to be used in this paper.

Definition. A petrinet of dimension k is a 5-tuple

$$N = (T, (t^-)_{t \in T}, (t^+)_{t \in T}, M_0, M_1)$$

where T is a nonempty finite set, t^+ and t^- are vectors in $\underline{N}^k$, and M_0, M_1 are vectors in $\underline{N}^k$.

The triple $(T, (t^-)_{t \in T}, (t^+)_{t \in T})$ constitutes what is sometimes called a generalized Petri net. For simplicity we have omitted the word "generalized" and added the initial and terminal vectors M_0 and M_1 to the specification of a petrinet. The above triple may (and occasionally in this paper it will) be visualized by a finite directed graph with vertex set $T \cup \{1,\ldots,k\}$ and edges leading from T to $\{1,\ldots,k\}$ and vice versa, namely $t^-(i)$ edges from i to t and $t^+(i)$ edges from t to i. The elements of $\{1,\ldots,k\}$ are the places, and the elements of T are the transitions of the petrinet. Any vector $M \in \underline{N}^k$ is called a marking of the petrinet; M(i) is the number of tokens in place i at marking M. M_0 (M_1) is called the initial (terminal) marking.

Example 6. Consider the 2-dimensional petrinet

$$N = (T, (t^-)_{t \in T}, (t^+)_{t \in T}, M_0, M_1)$$

with $T = \{t_1, t_2\}$, $t_1^-(1) = t_1^+(2) = t_2^-(2) = t_2^+(1) = 1$, $t_1^-(2) = t_1^+(1) = t_2^-(1) = t_2^+(2) = 0$, $M_0(1) = M_1(1) = 1$, $M_0(2) = M_1(2) = 0$. The graph representation of N with the marking M_0 ($=M_1$) is shown in Figure 5.

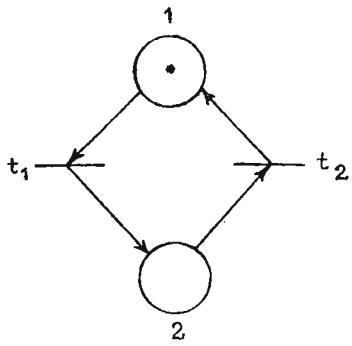


Fig. 5

Definition. (The next definitions refer to a fixed petrinet $N = (T, (t^-)_{t \in T}, (t^+)_{t \in T}, M_0, M_1))$. If M is a marking, then we say that a transition t is firable at M iff $M \geqq t^-$. The result of firing t at M is the marking $M + \Delta(t)$, where $\Delta(t) := -t^- + t^+$. (Comparison and addition of vectors are defined componentwise).

So far we have repeated well-known basic concept of petrinets. We hope that this will help the reader to get accustomed to our terminology. Normally at this point the notion of firing is extended to words, embedding the concept of a petrinet into that of a (possibly infinite) automaton. But we intend to define firability of partial words. For example, in the petrinet of Figure 1c) we wish not only the words e, a, b, ab, and ba, but also the partial word $a \times b$ to be firable at the shown initial marking M_0. This is natural, since $M_0 \geqq a^- + b^-$. (In other words, at the marking M_0 the transitions a and b are simultaneously enabled to fire. They form a "firable bag").

Definition. Let X be a finite alphabet. A bag of elements of X is a mapping of X into $\underline{N}$. If U is a bag and $x \in X$ then $U(x)$ is called the multiplicity of occurrence of x in U.

A bag is nothing but an $\underline{N}$-set [3] or, equivalently, an element of the free commutative semigroup (with unity) generated by X.

By the way we remark that a bag U could be identified with a partial word, namely a parallel product of symbols which contains the symbol x U(x) times. This identification is coherent with our notions of firability to be defined below.

In the following the petrinet N with transition set T is assumed fixed. Let Δ be the above defined mapping of T into k-dimensional vectors of integers.

Definition. Let U be a bag of transitions. We set

$$U^- := \Sigma\{U(t)\cdot t^- : t \in T\},$$

$$U^+ := \Sigma\{U(t)\cdot t^+ : t \in T\},$$

$$\Delta(U) := -U^- + U^+ = \Sigma\{U(t)\cdot\Delta(t) : t \in T\}.$$

We say that U is firable at the marking M iff $M \geqq U^-$.

Definition. Assume that $p = [(A,>,\beta)]$ is a partial word over T. If B is any subset of A then we set $(\beta B)(t) := \text{card}\{a : a \in B, \beta(a) = t\}$. βB is a bag of transitions (elements of T).

Definition. Let A be a set and $<$ a partial ordering on A. A subset $B \subseteq A$ is said to be left-monotonic in A under $<$ iff the following implication holds: If $b \in B$ and $a < b$, then $a \in B$.

Definition. Let $p = [(A,<,\beta)]$ be a partial word over T and let M be a marking of N. p is said to be weakly firable in N at M iff for any two disjoint subsets B and C of A the following implication holds:

If B and each $B \cup \{c\}$ $(c \in C)$ are left-monotonic in A under $<$, then βC is firable at $M + \Delta(\beta B)$.

Weak firability of partial words is a direct generalization of firability of words in the usual sense. The left-monotonic

sets are generalizations of the initial sehments of words; the reader may think of a left-monotonic set as a possible intermediate situation during the firing of the partial word.

Definition. Let M and M' be markings of N. M' is said to be reachable from M in N iff there exists a partial word $p = [(A,<,\beta)]$ such that p is weakly firable at M in N and $M + \Delta(\beta A) = M'$. $\Delta(\beta A)$ will be abbreviated by $\Delta(p)$.

The set of all partial words weakly firable at a given marking is a weak partial language. In the net of Figure 1c) the firability of ab and ba now appears as a consequence of the weak firability of $a \times b$. As we remarked in our introductory discussion, the mutual ordering of the occurrences of a and b in the weakly firable words ab and ba is somewhat artificial, not inherent in the petrinet. A natural notion of firability should exclude ab and ba. - Further discussion of this point is postponed to a later section. For the time being we are satisfied with weak firability.

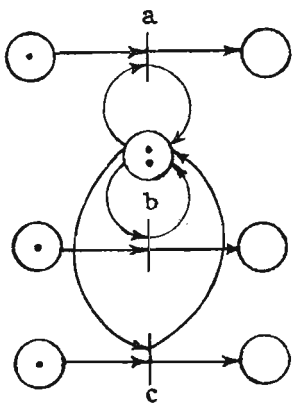


Fig. 6

Example 7. Figure 6 shows a 7-dimensional petrinet N with initial marking $M_0 = (1,1,1,0,0,0,2)$. In this net any two of the transitions a, b, and c are simultaneously firable at M_0,

while a, b, and c together are not. Examples of weakly firable partial words are abc, (a × b)c, and a × bc. a × b × c is not weakly firable.

Definition. The weak partial language defined by the petrinet $N = (T, (t^-)_{t \in T}, (t^+)_{t \in T}, M_0, M_1)$ is

$$\underline{WePL}(N) := \{p : p \in \underline{PW}(T),\ p \text{ weakly firable at } M_0 \text{ in } N,\ M_0 + \Delta(p) = M_1\}.$$

The language defined by the petrinet N is

$$\underline{L}(N) := \underline{WePL}(N) \cap \underline{W}(T) = \underline{Sm}(\underline{WePL}(N)).$$

Definition. Let PN denote the class of all petrinets.

$$\underline{WePL}(PN) := \{\underline{WePL}(N) : N \in PN\}.$$

$$\underline{RWePL}(PN) := \{h(L) : L \in \underline{WePL}(PN),\ h \text{ an } \underline{e}\text{-free renaming}\}.$$

$$\underline{R^eWePL}(PN) := \{h(L) : L \in \underline{WePL}(PN),\ h \text{ a renaming}\}.$$

The elements of $\underline{WePL}(PN)$ are called free, or unrenamed, weak petrinet partial languages. The elements of $\underline{RWePL}(PN)$ are called $\underline{e}$-free (-renamed) weak petrinet partial languages. The elements of $\underline{R^eWePL}(PN)$ are called (renamed) weak petrinet partial languages. Due to Proposition 13 all the partial languages belonging to these families are indeed weak ones. Smoothing the above families yields language families $\underline{L}(PN)$, $\underline{RL}(PN)$, and $\underline{R^eL}(PN)$, which are well-known from the literature. (In Hack [1] these families are denoted by L_0^f, L_0, and L_0^λ, respectively).

Proposition 14. $\underline{L}(PN) \subset \underline{WePL}(PN)$, $\underline{RL}(PN) \subset \underline{RWePL}(PN)$, $\underline{R^eL}(PN) \subset \underline{R^eWePL}(PN)$.

P r o o f. The inclusion is shown by equipping a given petrinet with a "run place", which carries one token and is connected with every transition via a loop:  This run place serializes the firing of transitions. For the new net N' we have $\underline{WePL}(N') = \underline{L}(N)$. ∎

Closure properties

As Hack [6], Starke [15], Greibach [5], and others did with petrinet languages, we want to characterize the families of partial languages just defined in terms of closure properties. We confine ourselves to WePL(PN) and R^{e}WePL(PN).

First we observe that WePL(PN) is not closed under union, Y-section, parallel and serial product, and parallel and serial iteration. (Some of these operations fail to preserve weakness, but even if we reestablish the weakness by application of We, the resulting operations do not preserve membership in WePL(N)). This follows from corresponding properties of L(PN) [6,15]. However, we have

Proposition 15. WePL(PN) is closed under weak restriction.

P r o o f. Assume that two petrinets

$$N' = (T', (t^{-'})_{t \in T'}, (t^{+'})_{t \in T'}, M_0', M_1')$$

and

$$N'' = (T'', (t^{-''})_{t \in T''}, (t^{+''})_{t \in T''}, M_0'', M_1'')$$

be given, with dimensions k' and k'' and possibly overlapping transition sets T' and T''. We set $T := T' \cup T''$ and $U := T' \cap T''$. We define a petrinet

$$N := (T, (t^{-})_{t \in T}, (t^{+})_{t \in T}, M_0, M_1)$$

of dimension k'+k''. It is constructed by "sticking together" N' and N'' at transitions belonging to U. More precisely, we set

$$M_0 := \begin{pmatrix} M_0' \\ M_0'' \end{pmatrix}, \qquad M_1 := \begin{pmatrix} M_1' \\ M_1'' \end{pmatrix},$$

$$t^- := \begin{cases} \begin{pmatrix} t^{-'} \\ 0 \end{pmatrix} & \text{if } t \in T' \setminus T'', \\ \begin{pmatrix} 0 \\ t^{-''} \end{pmatrix} & \text{if } t \in T'' \setminus T', \\ \begin{pmatrix} t^{-'} \\ t^{-''} \end{pmatrix} & \text{if } t \in U; \end{cases}$$

analogously we define t^+.

Suppose that $L' := \underline{WePL}(N')$, $L'' := \underline{WePL}(N'')$. We assume that T' (T'') be the alphabet of L' (L''). (Transitions of T' (T'') not occuring in any partial word of L' (L'') could be omitted). We show that $\underline{WePL}(N) = L' \,?\, L''$.

(I) $\underline{WePL}(N) \subseteq L' ? L''$.

Assume that $p \in \underline{WePL}(N)$. Define two renamings h' and h'' setting

$$h'(t) := \begin{cases} t & \text{if } t \in T' \\ \underline{e} & \text{if } t \in T \setminus T'; \end{cases}$$

$h''(t)$ is defined analogously.

(i) $p \in \underline{We}(h'(p) \underset{U}{\frown} h''(p))$.

Assume $p = [(A,<,\beta)]$. Then $h'(p) = [(A', <|_{A'}, h'\beta|_{A'})]$, with $A' = \{a : a \in A, h'\beta(a) \in T'\}$. Similarly $h''(p) = [(A'', <|_{A''}, h''\beta|_{A''})]$, with $A'' = \{a : a \in A,\ h''\beta(a) \in T''\}$. We set $\prec := \underline{Tr}(<|_{A'} \cup <|_{A''})$. This is a partial ordering since $<|_{A'} \cup <|_{A''} \subseteq <$. The latter inclusion also implies that $\prec \subseteq <$. Hence p is smoother than $q := [(A,\prec,\beta)]$, and $q \in h'(p) \underset{U}{\frown} h''(p)$.

(ii) $h'(p) \in L'$ and $h''(p) \in L''$.

The two statements are analogous to each other, and we prove the first one only.

Let B' and C' be subsets of A', and let B' and each $B' \cup \{c\}$, $c \in C'$, be left-monotonic subsets of A' under $<|_{A'}$. Form the

smallest subset of A which is left-monotonic under $<$ and contains $B' \cup \{c\}$, and denote it by Q_c.

We show that $B := \bigcup \{Q_c : c \in C'\} \setminus C'$ is a left-monotonic subset of A under $<$. Take $a \in A$ and $b \in B$ such that $a < b$. Then $b \in Q_c$ for a certain $c \in C'$, and $b \notin C'$. Hence $a \in Q_c$. Assume $a \in C'$. Since $b \in Q_c$, we have $b \leqq b'$ for a certain $b' \in B' \cup \{c\}$. Hence $a <|_{A'} b'$, which contradicts the assumption that B' as well as $B' \cup \{c\}$ and $B' \cup \{a\}$ are left-monotonic under $<|_{A'}$. Consequently $a \in Q_c \setminus C'$, and $a \in B$.

Now we show that each $B \cup \{c\}$, $c \in C'$, is left-monotonic under $<$: Assume that $a \in A$ and $a < c$. Then $a \in Q_c$. If $a \in C'$ then a and c would be comparable under $<|_{A'}$. But this is excluded by our assumption.

From the facts derived above we conclude that $\beta C'$ is weakly firable at $M_0 + \Delta(\beta B)$, i.e.

$$M_0 + \Delta(\beta B) \geqq (\beta C')^-.$$

Projection on the coordinates belonging to N' yields the inequality

$$M_0' + \Delta'(\beta B') \geqq (\beta C')^{-'}.$$

(Note that $\beta(B \setminus B')$ consists only of transitions not belonging to the set T').

We have shown that $\beta C'$ is firable in N' at $M_0' + \Delta'(\beta B')$. Consequently $h'(p)$ is firable in N' at M_0'. To show that $h'(p) \in L'$ we have to prove the equality $M_0' + \Delta'(\beta A') = M_1'$. But this follows from $M_0 + \Delta(\beta A) = M_1$ by projection on the coordinates belonging to N'. Hence $h'(p) \in L'$.

From (i) and (ii) our statement (I) follows immediately.

(II) $L' ? L'' \subseteq \underline{\mathrm{WePL}}(N)$.

It is sufficient to show that $L' ! L'' \subseteq \underline{\mathrm{WePL}}(N)$. Let $p' \in L'$ and $p'' \in L''$. This means that p' (p'') is weakly firabe at M_0' (M_0'')

in N' (N'') and that $M_o' + \Delta'(p') = M_1'$ and $M_o'' + \Delta''(p'') = M_1''$. We have to show that for each $p \in p' \underset{U}{\cap} p''$ p is weakly firable at M_o in N and $M_o + \Delta(p) = M_1$.

Let $p = [(A,<,\beta)]$. p' and p'' have representatives $(A',<',\beta')$ and $(A'',<'',\beta'')$ whose U-composition ϕs $(A,<,\beta)$. Let B and C be disjoint subsets of A such that B and $B \cup \{c\}$ are left-monotonic subsets of A under $<$, for each $c \in C$. $B \cap A'$ ($B \cap A''$) is left-monotonic in A' (A'') under $<'$ ($<''$). We part C into there subsets: $C' := C \cap (A' \setminus A'')$, $C'' := C \cap (A'' \setminus A')$, $D := C \cap A' \cap A''$. Then for all $c \in C' \cup D$ the set $(B \cap A') \cup \{c\}$ is a left-monotonic subset of A' under $<'$, and so for all $c \in C'' \cup D$ the set $(B \cap A'') \cup \{c\}$ is a left-monotonic subset of A'' under $<''$.

Hence

$$M_o' + \Delta'(\beta'(B \cap A')) \geqq (\beta'(C' \cup D))^{-'}$$

and

$$M_o'' + \Delta''(\beta''(B \cap A'')) \geqq (\beta''(C'' \cup D))^{-''}.$$

Combination of the two vector inequalities yields

$$M_o + \Delta(\beta B) \geqq (\beta C)^{-}.$$

Hence p is weakly firable at M_o in N. From $M_o' + \Delta'(p') = M_1'$ and $M_o'' + \Delta''(p'') = M_1''$ follows $M_o + \Delta(p) = M_1$. Thus we have shown that $p \in \underline{\text{WePL}}(N)$. ■

The closure property shown above suggests the question whether the partial languages belonging to $\underline{\text{WePL}}(PN)$ admit a canonical representation involving weak restriction and, possibly, some other operations. We shall give such a representation. For simplicity (not for essential reasons) we confine ourselves to pure petrinets. The reader should try to eliminate this technical restriction.

Definition. The k-dimensional petrinet $N = (T, (t^-)_{t \in T}, (t^+)_{t \in T}, M_0, M_1)$ is called a pure petrinet if for each coordinate $i \in \{1,\dots,k\}$ and each $t \in T$ the inequality $t^-(i) + t^+(i) \leqq 1$ holds.

This property means that the graph representation of the net contains no multiple arcs and no circuits of length 2.

Besides weak restriction our canonical representation will involve derivatives.

Definition. Let L be a partial language and let p and q be partial words. The weak derivative $\partial_q^p(L)$ is defined by

$$\partial_q^p(L) := \{r : prq \in \underline{We}(L)\}.$$

The weak derivative is always a weak partial language. If L is a language and p and q are words then $\partial_q^p(L)$ coincides with the well-known derivative of languages.

Definition. $\Sigma := \{\sigma,\tau\}$, $\Sigma_k := \{\sigma_1,\tau_1,\dots,\sigma_k,\tau_k\}$.

$$W := \{\sigma\tau\}^x, \quad W_k := \{\sigma_1\tau_1\}^x \times \dots \times \{\sigma_k\tau_k\}^x.$$

Let $\mathcal{D}_j^h$ denote the weak derivative $\partial_{\tau^j}^{\sigma^h}(W)$, and let $\mathcal{D}_{j_1\dots j_k}^{h_1\dots h_k}$ denote the weak derivative

$$\partial_{\tau_1^{j_1}\dots\tau_k^{j_k}}^{\sigma_1^{h_1}\dots\sigma_k^{h_k}}$$

$\mathcal{D}_j^h$ is the unrenamed weak partial language of the net of Figure 7 with initial marking h and terminal marking j. $\mathcal{D}_{j_1\dots j_k}^{h_1\dots h_k}$ is the unrenamed weak partial language of the net of Figure 8 with initial marking $M_0(i) = h_i$ and terminal marking $M_1(i) = j_i$.

Besides weak derivatives we shall need strong derivatives.

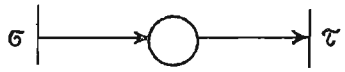


Fig. 7

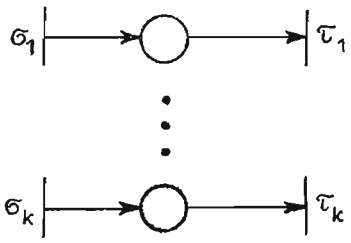


Fig. 8

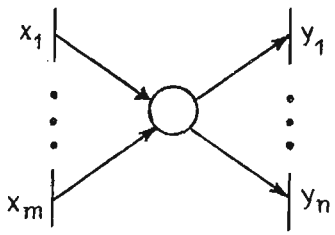


Fig. 9

Definition

$$\underline{\partial}_q^p(L) := \underline{St}(\partial_q^p(L))$$

is called the strong derivative of L with respect to p and q. The stron g derivatives of W and W_k will be denoted by $\underline{\mathcal{D}}_j^h$ and $\underline{\mathcal{D}}_{j_1,\ldots,j_k}^{h_1,\ldots,h_k}$, respectively.

Proposition 16. Assume that $L \subseteq \underline{PW}(T)$. There exists a pure petrinet N with $L = \underline{WePL}(N)$ iff L admits a representation $L = L_1 ? \ldots ? L_k$, each L_i being of the form $g^{-1}(V)$, where g is an e-free renaming and V is a weak derivative of W.

(Note that inverse e-free renamings trivially preserve weakness).

P r o o f. (I) Any partial language admitting the representation is defined by a pure petrinet.

Let V be a weak derivative of W, $V = \mathcal{D}_j^h$, and let $g: T \to \Sigma$ be an e-free renaming. Then $g^{-1}(V) = \underline{WePL}(N)$, where N is the net shown in Figure 9 with initial marking h and terminal marking j.

Thus $g^{-1}(V)$ is defined by a pure petrinet. It remains to show that if $L_1,\ldots,L_k$ are defined by pure petrinets then $L_1 ? \ldots ? L_k$ is defined by a pure petrinet. This is shown by applying to $L_1,\ldots L_k$ the construction of Proposition 15. That construction obviously yields a pure petrinet.

(II) Any partial language $L = \underline{WePL}(N)$, where N is a pure petrinet, admits the required representation.

Suppose $L = \underline{WePL}(N)$, where N is a k-dimensional pure petrinet. Let M_0 (M_1) be the initial (terminal) marking of N. Cut N into k one-dimensional nets N_i (i=1,...,k), N_i being of the form shown in Figure 9 with initial (terminal) marking $M_0(i)$ ($M_1(i)$). Let $L_i = \underline{WePL}(N_i)$. Then $L = L_1 ? \ldots ? L_k$. It remains to show that each L_i is of the form $g^{-1}(V)$ where V is a weak derivative of W. This is easily verified by looking at the net of Figure 9 and defining $g(x_1)=\ldots=g(x_m)=\sigma$ and $g(y_1)=\ldots=g(y_n)=\tau$. ■

Proposition 16 has its "smooth counterpart" in a theorem of Starke [1] on the canonical representation of petrinet languages.

Now let us turn to the family $\underline{R^{e}WePL}(PN)$.

Proposition 17. Assume that $L_1, L_2 \in \underline{R^{e}WePL}(PN)$. Then the following partial languages belong to $\underline{R^{e}WePL}(PN)$:

(1) $L_1 \cup L_2$,

(2) $L_1 * L_2$,

(3) $h(L_1)$, where h is an arbitrary renaming,

(4) $g^{-1}(L_1)$, where g is an e-free renaming,

(5) $\underline{We}(L_1 \underset{Y}{\frown} L_2)$, where Y is any alphabet.

P r o o f. Suppose $L_i = h_i(\underline{WePL}(N_i))$, $h_i: T_i \to X_i \cup \{e\}$ (i=1,2).

(1) We have to construct a net N and a renaming h such that $h(\underline{WePL}(N)) = L_1 \cup L_2$.

This is done in the same way as in the corresponding proof for languages [6]: The two nets are combined in the manner shown in Figure 10 via start transitions s_1, s_2 and end transitions

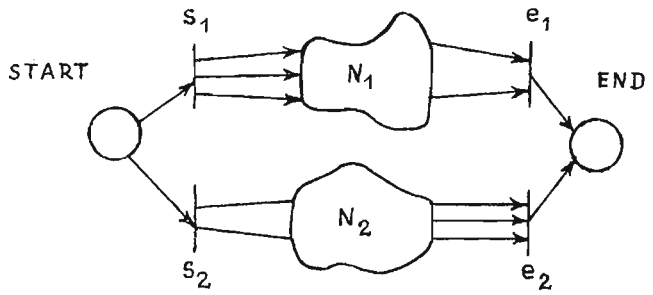


Fig. 10

e_1, e_2. s_i generates the initial marking M_{oi} in N_i, and e_i erases the terminal marking M_{1i} in N_i. The initial marking (terminal marking) of N is 1 on the place START (the place END)and 0 on all other places. The renaming h is defined by

$$h := h_1 \cup h_2 \cup \{(s_1, \underline{e}), (s_2, \underline{e}), (e_1, \underline{e}), (e_2, \underline{e})\}.$$

(2) Closure under serial product

Assumptions are as in (1). The construction of N is suggested by Figure 11.

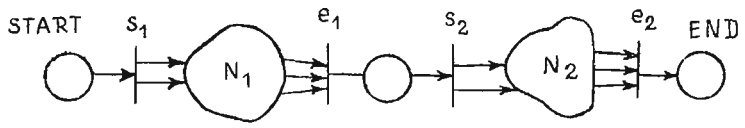


Fig. 11

(3) Closure under renaming

Trivial from the definition.

(4) Closure under inverse $\underline{e}$-free renaming

An easy application of the definitions: If $g: Y \longrightarrow X_i$ and $x \in X_1$ and $h(t) = x$ then we replace t in the net N by a collection $\{t_y : g(y)=x\}$ of transitions, with $t_y^- := t^-$ and $t_y^+ := t^+$. We set $f(t_y) := y$. If, however, $h(t) = \underline{e}$ then we don't

replace t, and we set $f(t) := \underline{e}$. For the resulting net N the equality $f(\underline{WePL}(N')) = g^{-1}(L)$ holds.

(5) We may assume that T_1 and T_2, the transition sets of the two petrinets, are disjoint, and that $Y \subseteq X_1 \cup X_2$. Set $F_i := \underline{WePL}(N_i)$.

Consider the partial language obtained from F_i by replacing each symbol $t \in T_i$ with $h(t_i) = y \in Y$ by the word $t'\ y\ t''$. (t' and t'' are fresh symbols). Denote this partial language by F_i'. We show that F_i' is an unrenamed weak petrinet partial language: Split in N_i each transition t with $h_i(t) = y \in Y$ into two transitions t' and t'' connected with each other via an additional place and also connected with additional places p_y and q_y in the way shown in Figure 12. For each $y \in Y$ we have one place p_y and one place q_y. We connect them via a transition named y, as shown in Figure 13.

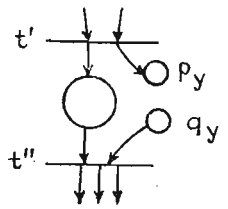


Fig. 12

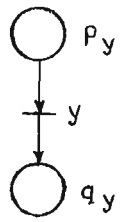


Fig. 13

The net obtained in this way, call it N_i', with the new places marked initially and terminally zero, generates F_i'.

Now define a renaming h' as follows:

$h'(t') := h'(t'') := \underline{e}$ if $t \in T_i$ and $h_i(t) \in Y$,

$h'(t) := h_i(t)$ if $t \in T_i$ and $h_i(t) \in Y$,

$h'(y) := y$ if $y \in Y$.

Then $L_i = h'(F_i')$, and $\underline{We}(L_1 \underset{Y}{\cap} L_2) = \underline{We}(h'(F_1' \underset{Y}{\cap} F_2')) = h'(F_1' \,?F_2')$, Due to Proposition 15 this is in $\underline{R^e PWePL}(PN)$. ■

Note that our list of closure properties does not contain serial and parallel iteration. There are good reasons for this on

the language level: If there exists an algorithm to solve the reachability problem for petrinets, then $(d\ L_0\ c)^*$ is not in $\underline{R^eL}(PN)$, where L_0 is defined as $L_0 = \{a^n b^n : n \in \underline{N}\}$ and a,b,c,d are distinct symbols. (Cf. Greibach [1]) But $(d\ L_0\ c)^* = (d\ L_0\ c)^{\parallel} \cap (d\ \{a,b\}^*\ c)^*$. The latter set is regular and hence in $\underline{R^eL}(PN)$. The result is that closure of $\underline{R^eL}(PN)$ under either serial or shuffle iteration would imply the unsolvability of the reachability problem.

On the other hand, the parallel iteration yields petrinet partial languages if applied to regular languages.

Proposition 18. If R is a regular language then $\underline{We}(R^{\mathbf{X}}) \in \underline{R^eWePL}(PN)$.

P r o o f. It is easy to see that every regular language R is defined by some petrinet of special type, satisfying the following equations:

$$t^+(1)+\ldots+t^+(k) = t^-(1)+\ldots+t^-(k) = 1 \text{ for each } t \in T,$$

$$M_0(1)+\ldots+M_0(k) = M_1(1)+\ldots+M_1(k) = 1.$$

(In common terms of petrinet theory, such petrinets are called "safe state machines"). We have $R = \underline{L}(N) = \underline{WePL}(N)$. There is exactly one place i with $M_0(i) = 1$ ($M_1(i) = 1$); we call this place the start place (the end place). Now we augment N with

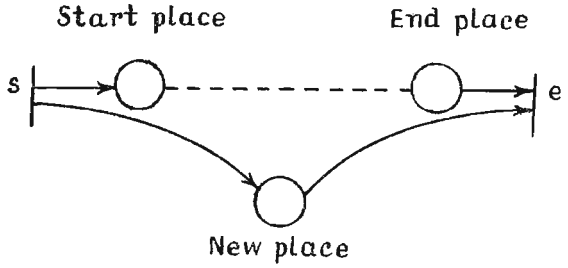


Fig. 14

two new transitions s and e and one new place, connected with the original net in the way shown in Figure 14. The new initial and terminal marking shall be zero on all places. Let N' denote the net obtained after these modifications. Let h be the renaming which erases s and e and leaves all other symbols invariant. The reader easily checks that $h(\underline{WePL}(N')) = \underline{We}(R^X)$. ■

The closure properties of $\underline{R^e WePL}(PN)$ shown in Proposition 17 suggest the question for an algebraic characterization of this family in terms of the operations mentioned in that proposition. Such a characterization may be based upon Proposition 16:

Proposition 19. $\underline{R^e WePL}(PN)$ is the smallest family of partial languages containing all weak derivatives of W and closed under weak restriction, inverse e-free renamings, and renamings.

The proof follows from Propositions 16 and 17 observing the fact that every partial language in $\underline{R^e WePL}(PN)$ can be defined by a pure petrinet. ■

Let us try to give another characterization.

Proposition 20. Suppose $L \in \underline{WePL}(PN)$. Then there exist a natural number k, a strong derivative V of W_k, a finite partial language F and a renaming g such that

$$L = \underline{We}(g(V_{\Sigma_k} \cap F^X)).$$

P r o o f. Suppose that $L = \underline{WePL}(N)$ where N is a petrinet with initial (terminal) marking M_0 (M_1). Without loss of generality assume that its transition set T is disjoint with Σ_k.

With each $t \in T$ we associate a partial word u_t, which contains τ_i exactly $t^-(i)$ times, for i=1,...,k, and no other symbols, and whose underlying partial ordering is empty (this means that u_t is the parallel product of its symbols). Similarly we assign to t a partial word v_t, which contains σ_i exactly

$t^+(i)$ times and is the parallel product of these symbols. Then we set $w_t := u_t\ t\ v_t$. This is a partial word over $\Sigma_k \cup T$. We define

$$F := \{w_t : t \in T\},$$

$$g(\sigma_i) := g(\tau_i) := \underline{e},$$

$$g(t) := t$$

Assume $M_0(i) = h_i$ and $M_1(i) = j_i$ $(i=1,\ldots,k)$.

We set $V := \mathfrak{D}^{h_1,\ldots,h_k}_{j_1,\ldots,j_k}$. This is a strong derivative of W_k We have $\underline{We}(V) = \underline{WePL}(N_k)$, where N_k is the petrinet of Figure 8 with initial (terminal) marking M_0 (M_1).

We will show that

$$L = \underline{We}(g(V \cap_{\Sigma_k} F^X)).$$

In the subsequent formulae we shall use the symbols Δ_N and Δ_{N_k} for better discrimination of the Δ-mappings of N and N_k.

(I) $L \subseteq \underline{We}(g(V \cap_{\Sigma_k} F^X))$.

Assume that $p = [(A, \prec, \beta)] \in L$. In a first step we construct a partial word u over $\Sigma_k \cup T$ by replacing in p the symbol t by the partial word w_t, for each $t \in T$.u is defined as follows: Assume that $w_{\beta(a)} = [(B_a, \sqsubset_a, \gamma_a)]$, and that the B_a's are pairwise disjoint. It is convenient to assume that a itself belongs to B_a. By name, a is that element of B_a whose γ_a-image is in T. Thus $\gamma_a(a) = \beta(a)$. - Set $B := \bigcup\{B_a : a \in A\}$, $\gamma := \bigcup\{\gamma_a : a \in A\}$. Define a partial ordering $\ll$ on B setting

$b \ll b' :\leftrightarrow$ there exist $a, a' \in A$ with $a \prec a'$, $b \in B_a$, $b' \in B_{a'}$ or there exists an $a \in A$ with $b \sqsubset_a b'$.

We set $u := [(B, \ll, \gamma)]$.

Now we claim that the restriction of u to $B \setminus A$ is in $\underline{WePL}(N_k)$ Δ_N and Δ_{N_k} enjoy the following relation:

$\Delta_N(\beta a) = \Delta_{N_k}(\gamma(B_a \setminus \{a\}))$ for all $a \in A$. (*) From this we conclude that

$$M_o + \Delta_{N_k}(\gamma(B \setminus A)) = M_o + \Delta_N(\beta A) = M_1 .$$

Hence it remains to show that the restriction of u to $B \setminus A$ is weakly firable at M_o in N_k.

Let C be a subset of $B \setminus A$ which is left-monotonic under $\ll|_{B\setminus A}$ the same shall be true of $C \cup \{d\}$ for each $d \in D$, D another subset of $B \setminus A$ disjoint with C. We have to show that at $M_o + \Delta_{N_k}(\gamma C)$ the bag γD is firable in N_k. We can think of the transitions belonging to this bag as belonging to $\{\tau_1, \ldots, \tau_k\}$, since the other transitions of N_k are anyway always firable. To C we add as many elements of A as necessary to obtain a set $C' \supseteq B$ which is left-monotonic under $\ll$ in B and for which each $C' \cup \{d\}$, $d \in D$, is left-monotonic under $\ll$ in B as well. As a consequence $C' \cap A$ is left-monotonic in A under $\prec$.

We set $E := \{e : e \in A, B_e \cap D \neq \emptyset\}$. Then each of the sets $(C' \cap A) \cup \{e\}$, $e \in E$, is left-monotonic in A under $\prec$. For if $a \prec e$ and $e \in E$ then $a \ll d$ for a certain $d \in D$, hence $a \in C'$. From this it follows that βE is firable at $M_o + \Delta_N(\beta(C' \cap A))$ in N. In other words:

$$M_o + \Delta_N(\beta(C' \cap A)) \geqq (\beta E)^- .$$

From equality (*) above follows

$$\Delta_N(\beta(C' \cap A)) = \Delta_{N_k}(\gamma(\cup\{B_a : a \in C' \cap A\} \setminus A)).$$

The set $\cup\{B_a : a \in C' \cap A\}$ is nearly equal to C' since C' is left-monotonic. Closer inspection shows that it may differ from C' only in that it contains some more elements from $\gamma^{-1}(\{\sigma_1, \ldots, \sigma_k\})$. Hence

$$\Delta_N(\beta(C' \cap A)) \geqq \Delta_{N_k}(\gamma(C' \setminus A)) = \Delta_{N_k}(\gamma C).$$

On the other hand

$$(\beta(e))^{-} \geqq \sum \{(\gamma(b))^{-} : b \in B_e \setminus \{e\}\}.$$

Hence

$$(\beta E)^{-} \geqq (\gamma D)^{-}.$$

Summarizing the inequalities derived above we obtain

$$M_o + \Delta_{N_k}(\gamma C) \geqq (\gamma D)^{-},$$

which was to be shown.

We have proved that $u|_{B \setminus A} \in \underline{\mathrm{WePL}}(N_k) = \underline{\mathrm{We}}(V)$. Consequently $u|_{B \setminus A}$ is smoother than a certain $v \in V$. Let $v = [(B \setminus A, <, \gamma|_{B \setminus A})]$.

In the next step we define another partial word w over $T \cup \Sigma_k$. w is the parallel product of all $w_{\beta(a)}$, a running through A. Let Γ be the union of the Γ_a's. Then $w = [(B, \Gamma, \gamma)]$.

u is smoother than w. u must be contained in the intersection $\underline{\mathrm{We}}(v \times T^X) \cap \underline{\mathrm{We}}(w)$. Hence $(v \times T^X) \overset{\frown}{\Sigma_k \cup T} w \neq \emptyset$ due to Proposition 10. But this implies $v \overset{\frown}{\Sigma_k} w \neq \emptyset$.

Lemma 20.1. If $v_1 \in v \overset{\frown}{\Sigma_k} w$ then p is smoother than $g(v_1)$.

P r o o f. Let $[(B, <_1, \gamma)] = v_1$. This means that $<_1$ is the transitive closure of $< \cup \Gamma$. We have to show that $<_1|_A$ is contained in $\Im$.

Assume that $a <_1 a'$, where a and a' are elements of A. Without loss of generality we may assume that a' is an immediate successor of a under $<_1|_A$.

Let $a = b_o <_1 b_1 <_1 \cdots <_1 b_m = a'$ be a chain of immediate successor under $<_1$. We show that all the elements of the chain are in $B_a \cup B_{a'}$. Assume that b_i is the first and b_j the last member of the chain which belongs to $B_{a''}$ (a'' distinct from a and a'). Then $b_i \Gamma a''$ and $a'' \Gamma b_j$, hence $a <_1 a'' <_1 a'$, in contradiction to our above assumption on immediate successorship.

Thus the chain reduces to

$$a \sqsubset b < b' \sqsubset a',$$

whence $b \ll b'$, $a \ll a'$, and finally $a \prec a'$. The lemma is proved.

From the above lemma we conclude that

$$p \in \underline{We}(g(v_1)) \subseteq \underline{We}(g(v \textstyle\bigcirc_{\Sigma_k} w)) \subseteq \underline{We}(g(V \textstyle\bigcirc_{\Sigma_k} F^X)).$$

(II) $\underline{We}(g(V \bigcirc_{\Sigma_k} F^X)) \subseteq L$

In this part of the proof we use strictly the same denotations as in the first part. Assume that $p = g(v_1)$, where $v_1 \in v \bigcirc_{\Sigma_k} w$, $v \in V$, and $w \in F^X$. We have to show that $p \in L$. Let $w = [(B, \sqsubset, \gamma)]$ be the parallel product of a family of partial words from F, which we denote by $\{w_{\beta(a)} : a \in A\}$. The partial word $w_{\beta(a)}$ is $[(B_a, \sqsubset_a, \gamma_a)]$, and B, $\sqsubset$, and γ are the union of the B_a's, the $\sqsubset_a$'s, and the γ_a's, respectively. β is a mapping of A into T. Since t occurs in $w_{\beta(a)}$, $\beta(a)$ occurs in $w_{\beta(a)}$. Assume that the index a is contained in B_a and that it denotes the place of the symbol $\beta(a)$. In other words, $\gamma_a(a) = \beta(a)$. We have $a \in A$.

We use appropriate representations for v_1 and v: $v_1 = [(B, <_1, \gamma)]$ $v = [(B \setminus A, <, \gamma|_{B \setminus A})]$. Hence p is equal to $[(A, <_1|_A, \beta)]$.

$M_0 \to \Delta(p) = M_1$ follows immediately from the equality (*).

It remains to show that p is weakly firable at M_0 in N.

Assume that C and E are disjoint subsets of A such that C and each $C \cup \{e\}$, $e \in E$, are left-monotonic in A under $<_1|_A$. We set $\tilde{C} := \cup\{B_a : a \in C\}$. It is easy to show that $\tilde{C}$ is left-monotonic in B unde $<_1$: If $b' \in \tilde{C}$ and $b \in B$ and $b <_1 b'$, then there exist some a and $a' \in A$ such that $b \in B_a$ and $b' \in B_{a'}$. We have $a' \in C$, and we want to show that $a \in C$. From the structure of the partially ordered sets $(B_a, \sqsubset_a)$ and $(B_{a'}, \sqsubset_{a'})$ it is easy to derive that $a \leqq_1 a'$. Hence $a \in C$.

We set $D := (\cup\{B_e : e \in E\}) \cap \gamma^{-1}(\tau_1, \ldots, \tau_k\})$. Then

$$(\gamma D)^{-} = (\beta E)^{-}.$$

Every $(\tilde{C} \setminus A) \cup \{d\}$, $d \in E$, is left-monotonic in $B \setminus A$ under $<_1 | B \setminus A$ and so is $\tilde{C} \setminus A$. Hence they are left-monotonic under the rougher partial ordering $<$. Consequently γD is firable in N_k at $M_0 + \Delta_{N_k}(\tilde{C} \setminus A)$. From

$$M_0 + \Delta_N(C) = M_0 + \Delta_{N_k}(\tilde{C} \setminus A) \geq (\gamma D)^{-} = (\beta E)^{-}$$

it follows that βE is firable at $M_0 + \Delta_N(C)$ in N. ■

Proposition 20 allows us to establish another characterization of $\underline{R^e WePL}(PN)$.

Proposition 21. $\underline{R^e WePL}(PN)$ is the smallest family of languages which contains the weak derivatives of all W_k's, containing each set of the form $\underline{We}(F^X)$, where F is a finite partial language, and which is closed under renamings and under the operation $(L_1, L_2) \mapsto \underline{We}(L_1 \underset{Y}{\frown} L_2)$ for every alphabet Y.

P r o o f. This follows from Propositions 17 and 20 in virtue of the facts that

$$\underline{We}(g(V \circledcirc_{\Sigma_k} F^X)) = g(\underline{We}(\underline{We}(V) \circledcirc_{\Sigma_k} \underline{We}(F^X))$$

and that $\underline{We}(F^X)$ belongs to $\underline{R^e WePL}(PN)$. The latter statement is proved similarly to Proposition 18. ■

3. Safe petrinets and regular partial languages

So far we have not given a satisfactory answer to the question "What is firability in petrinets". We have seen that weak firability does not match our intuition of that concept.

In this section the question will be answered for safe petrinets.

Definition. The k-dimensional petrinet $N = (T,(t^-)_{t \in T},(t^+)_{t \in T},M_0,M_1)$ is said to be safe if it obeys the following two properties:

(i) For each $t \in T$: $t^- + t^+ \neq 0$.

(ii) For each M: If M is reachable from M_0, then $M(i) \leqq 1$ for each $i \in \{1,\ldots,k\}$.

Condition (ii) says that every marking reachable from the initial marking is a zero-one marking(has only 0 and 1 as values). Condition (i) means that N contains no "isolated transitions". It has been added for convenience. Note that the safeness does not depend on the terminal marking. We say that N is safe at M iff the petrinet obtained from N by taking M as a new initial marking is a safe petrinet.

Safe petrinets are equivalent to Petri's condition/event nets (cf.Genrich et al. [4]). Their (unrenamed) partial languages were studied by Mazurkiewicz [10].

Definition. Let t_1 and t_2 be two transitions in a safe petrinet. t_1 and t_2 are said to be independent (t_1 In t_2) iff $t_1^-(i) + t_2^-(i) \leqq 1$ for each coordinate i.

Obviously a transition t cannot be independent of itself. Otherwise we would have $t^- = 0$, and this would mean that t either is isolated or can generate non-0-1 markings.

If two independent transitions are both firable at a marking at which N is safe, then the two transitions can fire simultaneously. Moreover, we have the following criterion of firability of a bag:

Proposition 22. Let M be a marking at which the petrinet N is safe. Then a bag U of transitions is firable at M if and only if the following conditions hold:

(i) $U(t) \leqq 1$ for each transition t, i.e. U can be regarded as a set of transitions.

(ii) For each t in U: $M \geqq t^-$.

(iii) For each pair t_1, t_2 of distinct transitions in U: t_1 In t_2.

P r o o f. (I) Necessity of the condition:

(i) Since M is zero-one, $M \geqq t^- + t^-$ would imply $t^- = 0$. Hence t would be isolated, or $M + \Delta(t) + \Delta(t)$ would be non-zero-one.

(ii) Trivial.

(iii) $1 \geqq M(i) \geqq t_1^-(i) + t_2^-(i)$.

(II) Sufficiency of the condition:

Let $M \geqq t_j^-$ $(j=1,\ldots,n)$ and t_j, t_k be pairwise distinct and independent. Then $M(i) \geqq t_1^-(i) + \ldots + t_n^-(i)$, since at most one of the summands is different from zero. ■

Proposition 22 states that in safe petrinets the simultaneous firability of several transitions is equivalent to individual firability plus pairwise independence. Hence it is equal to pairwise simultaneous firability. This enables us to give the following notion of firability of a partial word.

Definition. Let N be a safe petrinet at the marking M. Let $p = [(A, <, \beta)] \in \underline{PW}(T)$, where T is the transition set of N.

We say that p is firable at M in N iff p is weakly firable at M in N and for any two elements a and b of A the following implication holds: If $a < b$ then not $\beta(a)$ In $\beta(b)$.

(Remember that $<_\cdot$ denotes the relation of immediate successorship under $<$.)

Definition. Let N be a petrinet which is safe at its initial marking. Let M_0 (M_1) be its initial (terminal) marking.

$$\underline{PL}(N) := \{p : p \text{ firable in } N \text{ at } M_0 \text{ and } M_0 + \Delta(p) = M_1\}$$

is called the partial language defined by N.

The next proposition relates firability to weak firability.

Proposition 23. If N is safe then

$\{p : p \text{ firable at } M \text{ in } N\} = \underline{St}\,\{p : p \text{ weakly firable at } M \text{ in } N\}$, and

$$\underline{PL}(N) = \underline{St}(\underline{WePL}(N)).$$

P r o o f. Assume that p is firable at M in N. Clearly p is weakly firable. Assume that $p = [(A,<,\beta)]$ is smoother than a certain $q = [(A,<',\beta)]$, which is also weakly firable at M. Let $<$ contain $<'$. We have to show that $p = q$. If $p \neq q$ then there are elements a and b in A with $a <\cdot b$ and not $a <' b$. There exists a subset B of A such that B as well as $B \cup \{a\}$ and $B \cup \{b\}$ are left-monotonic in A under $<'$. Then $\beta\{a,b\}$ is firable at $M + \Delta(\beta B)$. By Proposition 22 this implies $\beta(a)$ $\underline{In}$ $\beta(b)$. Since p is firable and $a <\cdot b$, this is impossible. We have shown that $\underline{PL}(N) \subseteq \underline{St}(\underline{WePL}(N))$. Conversely, if $p = [(A,<,\beta)] \in \underline{St}(\underline{WePL}(N))$ and p is not firable, then there exists a pair (a,b) such that $a <\cdot b$ and a $\underline{In}$ b.

Deleting this pair from the relation $<$ yields a partial word q which, by Proposition 22, is weakly firable. But p is properly smoother than q. This is a contradiction to $p \in \underline{St}(\underline{WePL}(N))$. ∎

Definition.

$\underline{PL}(SPN) := \{\underline{PL}(N) : N \text{ is a safe petrinet}\}$,

$\underline{RPL}(SPN) := \{h(L) : L \in \underline{PL}(SPN) \text{ and } h \text{ is an } \underline{e}\text{-free renaming}\}$,

$\underline{R^{e}PL}(SPN) := \{h(L) : L \in \underline{PL}(SPN) \text{ and } h \text{ is a renaming}\}$.

Partial languages belonging to the family $\underline{R^{e}PL}(SPN)$ are called regular partial languages.

If this notion of regularity is restricted to languages, we obtain Kleene's well-known regular languages. (This will follow from Propositions 29 and 30). The generalization seems to be a natural one. It is an interesting question whether regular partial languages as defined above play the same role in partial language theory as the regular language do in language theory.

The class $\underline{PL}(SPN)$ has the following nice property;

Proposition 24. If N is a safe petrinet and p_1, p_2 are two different elements of $\underline{PL}(N)$, then $\underline{We}(p_1) \cap \underline{We}(p_2) = \emptyset$. The proof appears in Appendix A. ■

Proposition 25. $\underline{PL}(SPN)$ is closed under strong restriction.

P r o o f. Let L_1 and L_2 be in $\underline{PL}(SPN)$. We show that $L_1 \,!!\, L_2 \in \underline{PL}(SPN)$. Assume that $L_i = \underline{PL}(N_i)$ and that N_i safe. We set $K_i := \underline{We}(L_i)$. Then $K_i = \underline{WePL}(N_i)$. By Proposition 15: $K_1 \,?\, K_2 = \underline{WePL}(N)$ for some petrinet N. Moreover, inspecting the construction used in the proof of Proposition 15, we check that N is a safe petrinet. Hence $\underline{PL}(N) = \underline{St}(K_1 \,?\, K_2) = K_1 \,!!\, K_2 = L_1 \,!!\, L_2$. Here we have utilized the equality $\underline{St}(K_1 ! K_2) = \underline{St}(\underline{St}(K_1) ! \underline{St}(K_2))$, cf. the proof of Proposition 11 (III). ■

Proposition 26. $\underline{PL}(SPN)$ is closed under restriction.

Proof. We show that the restriction of two partial languages L_1 and L_2 in $\underline{PL}(SPN)$ always coincides with their strong restriction. Assume that p and p' are in $L_1 \,!\, L_2$ and that p is smoother than p'. We have to show that $p = p'$. Let T_i be the alphabet of L_i and $S := T_1 \cap T_2$. Let

$$p = [(A, <, \beta)], \quad p_i = [(A_i, <_i, \beta_i)] \in L_i,$$

$$p' = [(A, <', \beta)], \quad p_i' = [(A_i, <_i', \beta_i)] \in L_i$$

and assume that conditions (i)-(iv) of our definition of S-composition hold for p, p_1, p_2, and p', p_1', p_2', respectively. Further assume that $<$ contains $<'$.

$\underline{Tr}(<_i \cup <_i')$ is irreflexive since it is contained in $\underline{Tr}(< \cup <')$. Consequently $p_i \,\widehat{T_i}\, p_i' \neq \emptyset$, or $\underline{We}(p_i) \cap \underline{We}(p_i') \neq \emptyset$. By Proposition 24 this implies $p_i = p_i'$. Hence $p = p'$. ■

Proposition 27. Assume that L_1 and L_2 are regular partial languages, i.e. members of $R^{\oplus}\underline{PL}(SPN)$.

Then the following partial languages are regular:

(1) $L_1 \cup L_2$,

(2) $L_1 * L_2$,

(3) $h(L_1)$, where h is any renaming,

(4) $g^{-1}(L_1)$, where g is any e-free renaming,

(5) $L_1 \underset{Y}{\cap} L_2$, where Y is any alphabet (in particular: $L_1 \times L_2$),

(6) L_1^*.

P r o o f. The proofs of (1)-(4) use the same constructions as the corresponding proofs in Proposition 17. The proof of (5) also uses the construction mentioned in Proposition 17 plus the closure of PL(SPN) under restriction (Proposition 26). ■

Proposition 28. $R^{\oplus}PL(SPN)$ contains all finite partial languages.

P r o o f. Follows from Propositions 12 and 27. ■

Proposition 29. $R^{\oplus}PL(SPN)$ contains all regular languages.

P r o o f. Any regular language is defined by a "safe state machine" as described in the proof of Proposition 18. This is a safe petrinet. ■

Proposition 30. A language R is regular iff there exists a regular partial language L such that R = Sm(L).

P r o o f. If R is a regular language, then R itself is in $R^{\oplus}PL(SPN)$. (Cf. Proposition 29). Assume that $L \in R^{\oplus}PL(SPN)$. We have to show that Sm(L) is a regular language. Assume that L = = h(PL(N)), where N is a safe petrinet and h is a renaming. Apply to F := PL(N) the construction of the proof of Proposition 16: We(F) = F_1 ? ... ? F_k, where F_i is the free weak partial language defined by the 1-dimensional petrinet corresponding to the i^{th} coordinate of N. However, since N is safe, during firing any partial word of We(F) the i^{th} coordinate cannot assume any value greater than 1. This means that the transitions which increase this coordinate must fire alternatingly with those which decrease it. Hence We(F) = R_1 ? ... ? R_k, where R_i is some regular language contained in F_i. From this we conclude that:

$$\underline{Sm}(L) = h(\underline{Sm}(\underline{We}(F))) = h(\underline{Sm}(R_1 \,?\, \ldots \,?\, R_k)))$$
$$= h(\underline{Sm}(R_1)\circledR\ldots\circledR\,\underline{Sm}(R_k))$$
$$= h(R_1\circledR\ldots\circledR R_k),$$

which is a regular language. ■

Proposition 31. A partial language L is regular iff it admits a representation

$$L = g(V \underset{\Sigma_k}{\cap} F^{\times}),$$

where k is a natural number, V is a strong derivative of $(\sigma_1\tau_1)^*\times\ldots\times(\sigma_k\tau_k)^*$, F is a finite partial language over $\Sigma_k \cup T$ where T is a finite alphabet, $F\cap\underline{PW}(T) = \emptyset$, and g is a renaming.

P r o o f. (I) Necessity of the condition.

Suppose $L = h(\underline{PL}(N))$, where N is a safe petrinet. Apply to $\underline{WePL}(N)$ the construction of Proposition 20. We obtain a representation

$$\underline{WePL}(N) = \underline{We}(g(\tilde{V} \underset{\Sigma_k}{\cap} F^{\times})),$$

where $\tilde{V}$ is a strong derivative of W_k. Since N is safe, it does not contain isolated transitions, whence $F\cap\underline{PW}(T) = \emptyset$. The above equality implies

$$\underline{PL}(N) = \underline{St}\,(g(\tilde{V} \underset{\Sigma_k}{\cap} F^{\times})).$$

Assume that p and p' be two elements of $g(V \underset{\Sigma_k}{\cap} F^{\times})$ and that p' is properly smoother than p. Both p and p' belong to $\underline{WePL}(N)$. Let $p = g(v_1), p' = g(v_1'), v_1 \in v \underset{\Sigma_k}{\cap} w$, $v_1' \in v' \underset{\Sigma_k}{\cap} w'$, with $v, v' \in \tilde{V}$ and $w, w' \in F^{\times}$. Since p and p' consist of the same symbols, the same holds for w and w' (observe the definition of F and g!) and consequently for v and v'. But this implies w = w' (due to commutativity of the parallel product) and v = v' (since $\tilde{V}$ is a strong derivative of W_k). By Lemma 20.1 p is smoother than $g(v_1') = p'$. Hence p is properly smoothe than p. This contradiction shows that

$$\underline{St}(g(\tilde{V} \overset{\frown}{\Sigma_k} F^X)) = g(\tilde{V} \overset{\frown}{\Sigma_k} F^X).$$

By a similar argument to that applied in the proof of Proposition 30 we can show that $\tilde{V}$ can be replaced by a derivative V of $(\sigma_1\tau_1)^* x \ldots x (\sigma_k\tau_k)^*$. From

$$\underline{PL}(N) = g(V \overset{\frown}{\Sigma_k} F^X)$$

follows

$$L = hg(V \overset{\frown}{\Sigma_k} F^X).$$

(II) Sufficiency of the condition.

F is a regular partial language. The same is easily proved of V. However, Proposition 27 does not apply automatically, since our representation involves parallel iteration, an operation that does not preserve regularity.

However, since every partial word in F contains at least one symbol from Σ_k, it is evident that there must be a natural number n such that $V \overset{\frown}{\Sigma_k} F^X = V \overset{\frown}{\Sigma_k} (\underbrace{F^* x \ldots x F^*}_{n \text{ factors}})$. Now Proposition 27 applies. ■

Proposition 32. $\underline{R^e PL}(SPN)$ is the smallest family of partial languages containing each set $\{x\}$, where x is a symbol, and being closed under union, serial product, renaming, Y-section and serial iteration.

P r o o f. Follows from Propositions 12, 27, and 31, in virtue of the facts that F^X in Proposition 31 can be replaced by some finite parallel product $F^* x \ldots x F^*$, that parallel product coincides with $\emptyset$-section, and that a strong derivative of $(\sigma_1\tau_1)^* x \ldots x (\sigma_k\tau_k)^*$ is the parallel product of (languagetheoretic) derivatives of $(\sigma_i\tau_i)^*$. These derivatives can clearly be constructed from symbols by serial product and serial iteration. ■

4. On firability in arbitrary petrinets

In Section 3 we found an adequate notion of firability of partial words in safe petrinets. If N is a petrinet and M is a marking of N, let us define strong firability as follows:

Definition: p is strongly firable at M in N iff p belongs to

$$\underline{St}(\{q : q \text{ is weakly firable at } M \text{ in } N\}).$$

From Proposition 23 we know that firability in safe petrinets means nothing else than strong firability. We might take strong firability as the notion of firability in the general case, too, but it is doubtful wether this is the most adequate setting. It turns out that the fine properties derived for safe petrinets do not carry over to strong firability in the general case. In Example 7 the partial words $a \times bc$, $b \times ac$, and $c \times ab$ (and others) are strongly firable. This example shows that Proposition 22 and 24 are no longer true. Example 4 falsifies Proposition 26, and hence our proof of Proposition 27 (5) fails to carry over to the general case.

Let $\underline{StPL}(N)$ be the partial language defined by N with respect to strong firability, and $\underline{StPL}(PN) := \{\underline{StPL}(N) : N \text{ a petrinet}\}$, $\underline{R^eStPL}(PN) := \{h(\underline{StPL}(N)) : N \text{ a petrinet, } h \text{ a renaming}\}$. As in Proposition 25, we can show that $\underline{StPL}(PN)$ is closed under strong restriction. (Compare with Proposition 15!)An analogon of Proposition 16 can be shown for $\underline{StPL}(PN)$ by using strong instead of weak restriction. Finally, a "strong" version of Proposition 20 can be proved: Assume that $L \in \underline{StPL}(PN)$, or, more generally, $L \in \underline{R^eStPL}(PN)$. Then there exist k, T, a strong derivative V of W_k and a finite partial language F over $\Sigma_k \cup T$ such that $L = g(V \cap \Sigma_k^* F^\times)$, for some renaming h. The idea is the same as in the proof of Proposition 31.

The next example shows that our notion of strong firability involves curious consequences.

Example 8. Figure 15a shows a 4-dimensional petrinet with initial marking (1,0,0,1). The terminal marking is to be the same.

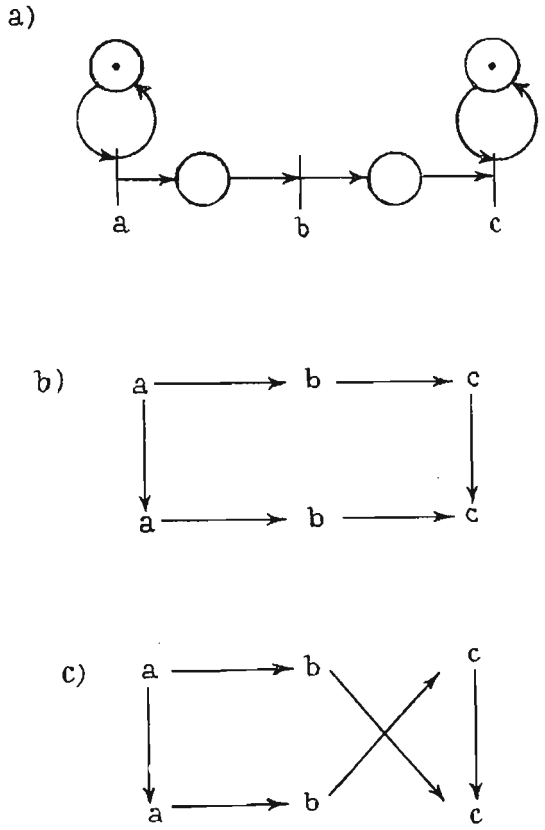


Fig. 15

L(N) is the language $(abc)^{\parallel}$. Two partial words belonging to StPL(N) are depicted in Figures 15b and 15c. Since they are incomparable under the smoother-relation, we are suggested to think of them as representing two essentially different ways of "executing" the petrinet. But this would mean that it makes an essential difference whether the first occurrence of c consumes a token produced by the first or the second occurrence of a, etc. Transition c seems to be able to discriminate between two tokens lying on

the same place. The reader should reflect on the question whether this is reasonable.

Thus, the question "Which is the adequate notion of firability in arbitrary petrinets?" is left open.

At this point we end our discussion of petrinet partial languages. In the Appendices we will relate petrinet partial languages to other partial language concepts, and give some concluding remarks.

APPENDIX A. Traces and trace languages

Definition. Let $w = [(A,\ll,\beta)]$ be a word in X^*, and let U be a symmetric and irreflexive binary relation on X. A permutation π of A is called a U-permutation for $(A,\ll,\beta)$ if for any pair (a,a') of elements of A the following implication holds: if $a \ll a'$ and $\pi(a) \gg \pi(a')$ then $\beta(a)\ U\ \beta(a')$.

Definition. Two words w and w' over X are said to be U-equivalent iff they obey representations $w = [(A,\ll,\beta)]$ and $w' = [(A,<<, \beta\pi^{-1})]$, where π is a U-permutation for $(A,\ll,\beta)$.

Here and in the sequel, X and the symmetric and irreflexive binary relation U are assumed fixed.

Proposition A1. U-equivalence is an equivalence relation on X^*.

P r o o f. Reflexivity is trivial.

Symmetry: If π is a U-permutation for $(A,\ll,\beta)$, then π^{-1} is a U-permutation for $(A,\ll,\beta\pi^{-1})$.

Transitivity: If π is a U-permutation for $(A,\ll,\beta)$ and ρ is a U-permutation for $(A,\ll,\beta\pi^{-1})$ then $\rho\pi$ ("apply first π, then ρ") is a U-permutation for $(A,\ll,\beta)$. ∎

By the way we remark that U-equivalence is the reflexive and transitive closure of the following binary relation: "w' is obtained from w by interchanging two symbols which are neighbors in w and which stand in the relation U". The proof of this fact is an easy application of the definition.

Definition. $[w]_U$ denotes the equivalence class of the word w under U-equivalence. (This is a finite language).

Mazurkiewicz [10] defined a trace over X (with respect to U) as a U-equivalence class $[w]_U$, and a trace language as a set of traces. He introduced operations on trace languages and studied some concept of defining trace languages by petrinets.

It was also Mazurkiewicz who pointed out the connection between U-equivalence classes and partial words. This connection will now be established in a precise manner.

Definition. Suppose $p = [(A,<,\beta)] \in \underline{PW}(X)$. p is called a weak U-trace iff for any two elements a,a' of A the following implication holds: If a and a' are incomparable under $\leqq$, then $\beta(a)\ U\beta(a')$.

p is called a U-trace iff $p \in \underline{St}(\{p : p \text{ is a weak U-trace}\})$.

Proposition A2

(i) If $w \in X^*$, then $[w]_U = \underline{Sm}(p)$ for exactly one U-trace p.

(ii) If p is a U-trace, then $\underline{Sm}(p) = [w]_U$ for every $w \in \underline{Sm}(p)$.

P r o o f.

(I) If p is a weak U-trace and $w \in \underline{Sm}(p)$, then $\underline{Sm}(p) \subseteqq [w]_U$.

Suppose that $p = [(A,<,\beta)]$, $w = [(A,\ll,\beta)]$, and $w' = [(A,\ll',\beta)]$, where $w,w' \in \underline{Sm}(p)$. We have to show that w and w' are U-equivalent. Take the permutation π of A which obeys the following condition: $a_1 \ll' a_2$ iff $\pi(a_1) \ll \pi(a_2)$. π is an isomorphism of ordered sets which maps $(A, \ll')$ to $(A,\ll)$. This implies that the labelled ordered sets $(A,\ll',\beta)$ and $(A,\ll,\beta\pi^{-1})$ are isomorphic. Hence $w = [(A,\ll,\beta\pi^{-1})]$.

It remains to show that π is a U-permutation. Assume that $a_1 \ll a_2$ and $\pi(a_1) \gg \pi(a_2)$. Then $a_1 \gg' a_2$. Since w and w' are in $\underline{Sm}(p)$, this implies that neither $a_1 \leqq a_2$ nor $a_1 \geqq a_2$. Hence $\beta(a_1)\ U\ \beta(a_2)$. Assertion (I) is proved.

Given $w = [(A,\ll,\beta)] \in X^*$, we define a partial word $[w]_U := [(A,<,\beta)]$ setting

$a < a' :\leftrightarrow (a) \ll \pi(a')$ for every U-permutation π of $(A,\ll,\beta)$.

It is evident that the relation $<$ is a partial ordering.

(II) If $w \in X^*$ then $\langle w\rangle_U$ is a weak U-trace, and $[w]_U \subseteqq \underline{Sm}(\langle w\rangle_U)$. Suppose that a and a' be incomparable with respect to the partial ordering $\leqq$ of $\langle w\rangle_U$. By symmetry we may assume that $a \ll a'$. By the definition of $<$ there must exist a U-permutation π for $(A,\ll,\beta)$

such that $\pi(a) \gg \pi(a')$. But this implies $\beta(a) \cup \beta(a')$. Hence $\langle w \rangle_U$ is a weak U-trace.

Now assume that $w' \in [w]_U$. Let $w' = [(A, \ll, \beta\pi^{-1})]$ for some U-permutation π. We have $w' = [(A, \ll', \beta)]$, where $a_1 \ll' a_2 :\leftrightarrow \pi(a_1) \ll \pi(a_2)$. If $a_1 < a_2$, then $\pi(a_1) \ll \pi(a_2)$ by the definition of $\langle w \rangle_U$. Hence $a_1 \ll' a_2$. This means that w' is smoother than $\langle w \rangle_U$. Hence $[w]_U \subseteq \underline{Sm}(\langle w \rangle_U)$.

Now we prove (i). From (I) and (II) follows that $[w]_U \subseteq \underline{Sm}(\langle w \rangle_U)$. $\langle w \rangle_U$ is a U-trace, for otherwise it would be properly smoother than some weak U-trace q, $\underline{Sm}(\langle w \rangle_U) \subsetneq \underline{Sm}(q)$. On the other hand, (I) implies $\underline{Sm}(q) \subseteq [w]_U$, which yields a contradiction. The uniqueness statement in (i) is evident, since there cannot be two different partial words with identical smoothings.

Finally we prove (ii). If p is a U-trace and $w \in \underline{Sm}(p)$ then $\underline{Sm}(p) \subseteq [w]_U$ by (I). On the other hand $[w]_U \subseteq \underline{Sm}(\langle w \rangle_U)$, and $\langle w \rangle_U$ is a U-trace, too. Hence $\underline{Sm}(p) = \underline{Sm}(\langle w \rangle_U)$. This implies the desired equality $\underline{Sm}(p) = [w]_U$. ■

Henceforth we shall use the notation $\langle w \rangle_U$ introduced in our proof for the U-trace corresponding to the word w. Note that from our proposition it follows that U-traces behave like words in some sense: The smoothings of two different U-traces are disjoint.

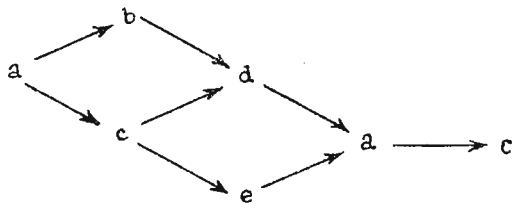


Fig. 16

Example 9 (taken from Mazurkiewicz [1]). Suppose that $X := \{a,b,c,d,e\}$, $U := \{(b,c),(c,b),(b,e),(e,b),(d,e),(e,d)\}$, $w := abcdeac$. Then $[w]_U = \{abcdeac, abcedac, acbdeac, acbedac, acebdad\}$. The U-trace $\langle w \rangle_U$ is shown in Figure 16.

Proposition A3. The set of all U-traces over X is the regular partial language

$$L := Y_1^* \,!\, \ldots \,!\, Y_n^*$$

where $\{Y_1,\ldots,Y_n\} := \{\{x,y\} : x \in X,\ y \in X,\ (x,y) \notin U\}$.
(Note that $\{Y_1,\ldots,Y_n\}$ contains all singletons $\{x\}$, $x \in X$).

P r o o f. By Proposition 9

$$L = \bigcap_X \{Y_i^* \times (X \setminus Y_i)^X : i=1,\ldots,n\} \bigcap_X (Y_1 \cup \ldots \cup Y_n)^X.$$

The last term is equal to X^X and may therefore be omitted:

$$L = \bigcap_X \{Y_i^* \times (X \setminus Y_i)^X : i=1,\ldots,n\}.$$

(I) Every element of L is a weak U-trace.

Assume that $p = [(A,<,\beta)] \in L$. Let a and a' be two elements of A which are incomparable under $<$. We have to show that $\beta(a) \cup \beta(a')$. p belongs to a set

$$\bigcap_X \{p_i \times q_i : i=1,\ldots,n\}$$

with $p_i \in Y_i^*$, $q_i \in (X \setminus Y_i)^X$. Let $p_i = [(A_i, <_i, \beta|_{A_i})]$, $q_i = [(A \setminus A_i, \emptyset, \beta|_{A \setminus A_i})]$ and $< = \underline{Tr}(<_1 \cup \ldots \cup <_n)$.

If not $\beta(a) \cup \beta(a')$ then $\{\beta(a), \beta(a')\} = Y_i$ for a certain. Hence $\{a,a'\} \subseteq A_i$. But $<_i$ is an ordering on A_i. a and a' must therefore be comparable under $<_i$ and a fortiori under $<$.

(II) Every weak U-trace over X is in $\underline{We}(L)$.

Assume that $p = [(A,<,\beta)]$ is a weak U-trace. Set $A_i := \beta^{-1}(Y_i)$, $<_i := <|_{A_i}$ $(i=1,\ldots,n)$. By definition of Y_i the relation $<_i$ is an ordering. Set $p_i := [(A_i,<_i,\beta|_{A_i})] \in Y_i^*$, $q_i := [(A \setminus A_i, \emptyset, \beta|_{A \setminus A_i})] \in (X \setminus Y_i)^X$. Set $<' := \underline{Tr}(<_1 \cup \ldots \cup <_n)$. Then $<'$ is contained in $<$, and hence p issmoother than $p' := [(A,<',\beta)] \in L$.

(III) Any two different elements of L have dosjoint smoothings.

Assume that $p, p' \in L$ and $\underline{Sm}(p) \cap \underline{Sm}(p') \neq \emptyset$. By Proposition 6 this means that $p \underset{X}{\cap} p' \neq \emptyset$. Hence there exist representations $p = [(A,<,\beta)]$ and $p' = [(A,<',\beta)]$ such that $\underline{Tr}(< \cup <')$ is a partial ordering. We show that $<$ is contained in $<'$. Suppose that $a < b$. Since $p \in L$, $<$ is the transitive closure of $<_1 \cup \ldots \cup <_n$, where $<_i$ is an ordering on $A_i := \beta^{-1}(Y_i)$. Hence there exists a chain $a = a_0 <\cdot \ldots <\cdot a_m = b$ such that $\{a_j, a_{j+1}\} \subseteq Y_{i_j}$ $(j=0,\ldots,m-1)$. But then $(\beta(a_j), \beta(a_{j+1})) \notin U$. Hence a_j and a_{j+1} must be comparable under $<'$. $a_{j+1} <' a_j$ would contradict the assumption that $\underline{Tr}(< \cup <')$ is irreflexive. Hence $a_j <' a_{j+1}$, and $a <' b$.

In the same way it is shown that $<'$ is contained in $<$. Hence $p = p'$.

Now let us combine the results (I)-(III) to prove our proposition.

If p is a U-trace then $p \in \underline{We}(L)$ by (II). Let $p \in \underline{We}(q)$ and $q \in L$. Then q is a weak U-trace due to (I). Hence $p = q$, and $p \in L$.

Conversely, if $p \in L$, then p is a weak U-trace due to (I). Let $p \in \underline{We}(q)$, where q is a U-trace. Then $q \in L$. Due to (III) this implies that $p = q$, and p is a U-trace. ■

In Mazurkiewicz's work appears the notion of a regular trace language. Let us discuss this notion.

Definitions. If L is a language over X then

$$\langle L \rangle_U := \{\langle w \rangle_U : w \in L\}.$$

A regular U-trace language is a partial language $\langle L \rangle_U$ where L is a regular language.

The serial U-trace product of two U-traces is defined by $\langle w_1 \rangle_U \circ \langle w_2 \rangle_U := \langle w_1 w_2 \rangle_U$. (Justification of this definition is left to the reader. The serial U-trace product of two U-trace languages is defined by $\langle L_1 \rangle_U \circ \langle L_2 \rangle_U := \langle L_1 L_2 \rangle_U$. The serial U-trace iteration of $\langle L \rangle_U$ is

$(\langle L\rangle_U)^\circ := \cup\{L_i : i \in \underline{N}\}$, where $L_0 := \{\underline{e}\}$, $L_{i+1} := L_i \circ \langle L\rangle_U$.

Example 10. Set $X := \{x,y,z\}$, $U := \{(x,y),(y,x)\}$. Then $\langle zx\rangle_U = zx$, $\langle yz\rangle_U = yz$, and $\langle zx\rangle_U \circ \langle yz\rangle_U = z(x \times y)z$.

The example shows that the serial product is not identical with the serial product of partial words, as it might appear at first sight. Generally the serial U-trace multiplication is dependent on the relation U, although this is not expressed in our notation.

Proposition A4. (Mazurkiewicz [1]). A U-trace language is regular iff it can be built from the U-trace languages $\{\underline{e}\}$ and $\{x\}$ $(x \in X)$ by means of union, serial U-trace product, and serial U-trace iteration.

P r o o f: Obvious. ■

Is every regular trace language a regular partial language? The answer is no: Let $X := \{a,b,c\}$, $U := \{(a,b), (b,a)\}$. Let $L = (\{ab\}^*\{c\})^*$. Then $\langle L\rangle_U = (L_0\ \{c\})^*$, where $L_0 := \{a^n \times b^n : n \in \underline{N}\}$. $(L_0\ \{c\})^*$ is a non-regular partial language (its smoothing is a non-regular language).

Is every regular partial language a regular trace language? The answer is no: Example 7 gives a finite partial language which cannot be a trace language since the smoothings of its elements are not pairwise disjoint.

Nevertheless there is a close connection between petrinets and regular trace languages. It has been pointed out by Mazurkiewicz.

Definition. Let t_1 and t_2 be two transitions in a safe petrinet. t_1 and t_2 are said to be fully independent (t_1 IN t_2) iff for each coordinate i the sums

$$t_1^-(i)+t_2^-(i),\ t_1^+(i)+t_2^-(i),\ t_1^-(i)+t_2^+(i)$$

are less than or equal to 1.

Although full independence is a stronger condition than independence as defined in Section 3, Proposition 22 remains valid if we replace In by IN. For if t_1 and t_2 are simultaneously firable at a marking M at which the petrinet is safe, and if $t_1^-(i) = 1$, then $t_2^-(i) = 0$ and $M(i) = 1$. Since t_2 is firable, the marking $M + \Delta(t_2)$ is reachable. This must be a zero-one marking. Hence $t_2^+(i) = 0$.

Similarly we may replace In by IN in the definition of firability, without changing the meaning.

Proposition A5. If N is a safe petrinet, then PL(N) is a regular IN-trace language.

P r o o f. Let M_0 (M_1) be the initial (terminal) marking of N.

(I) If p is weakly firable at M_0, then p is a weak IN-trace.

Suppose $p = [(A,<,\beta)]$. Let a and a' be incomparable under $\leqq$. There exist three left-monotonic subsets B, $B \cup \{a\}$, and $B \cup \{a'\}$ of A. Hence $\beta(a)$ and $\beta(a')$ are simultaneously firable at $M_0 + \Delta(\beta B)$. Proposition 22 says that they must be fully independent: $\beta(a)$ IN $\beta(a')$.

(II) Let $p = [(A,<,\beta)]$ be a weak IN-trace. Assume that for each subset B of A the following implication holds: If B and $B \cup \{a\}$ are left-monotonic subsets of A under $<$, $a \notin B$, then βa is firable at $M_0 + \Delta(\beta B)$. Then p is weakly firable at M_0 in N.

P r o o f: If B and C are disjoint subsets of A, and B and $B \cup \{c\}$ are left-monotonic under $<$ for each $c \in C$, then the elements of C are pairwise incomparable under $<$. Since p is a weak IN-trace we have $\beta(c)$ IN $\beta(c')$ for any two distinct elements c and c' of C. Proposition 22 says that βC is firable at $M_0 + \Delta(\beta B)$.

(III) If $p \in$ PL(N), then p is an IN-trace.

By (I) p is a weak IN-trace. Assume that $p = [(A,<',\beta)]$ and that p is smoother than some $q = [(A,<,\beta)]$, $<$ contained in $<'$, and that q is a weak IN-trace as well. We show that $q \in$ WePL(N).

Assume that B and $B \cup \{a\}$ are left-monotonic subsets of A under $<$. Due to (II) it is sufficient to show that this implies firability of βa at $M_o + \Delta(\beta B)$.

Form the smallest subset of A which contains $B \cup \{a\}$ and is left-monotonic with respect to $<'$. We denote this set by $B' \cup \{a\}$ and assume that $a \notin B'$. We show that B' is left-monotonic under $<'$. If this were not true, we would have $a <' b'$ for some $b' \in B'$. By definition of B' this implies $a <' b$ for some $b \in B \cup \{a\}$. Obviously $b \in B$. We consider a chain of immediate successors:

$$a = b_o <'' \ldots <'' b_n = b.$$

Since $p \in \underline{PL}(N)$, we have not $\beta(b_{i-1})\ \underline{IN}\ \beta(b_i)$ $(i=1,\ldots,n)$. Since q is a weak $\underline{IN}$-trace, b_{i-1} must be comparable with b_i under $<$. $b_{i-1} > b_i$ is impossible, since $< \subseteq <'$. Hence $b_{i-1} < b_i$, for $i=1,\ldots,n$. This implies $a < b$, which contradicts the left-monotonicity of B under $<$.

Weak firability of p means $M_o + \Delta(\beta B') \geqq (\beta(a))^-$. To show that q is weakly firable, we must prove the inequality $M_o + \Delta(\beta B) \geqq (\beta(a))^-$. Assume that this inequality is false. Then there must be a $d \in B' \setminus B$ and a coordinate i such that $(\beta d)^+(i) = 1$ and $(\beta a)^-(i) = 1$. But d is incomparable with a under $<$, and q is a weak $\underline{IN}$-trace. Hence $\beta d\ \underline{IN}\ \beta a$, and $(\beta d)^+(i) + (\beta a)^-(i) \leqq 1$. This is a contradiction.

We have shown that $q \in \underline{WePL}(N)$. Since $\underline{PL}(N) = St(PL(N))\ St(\underline{WePL}(N))$, q must equal p.

p is not smoother than any weak $\underline{IN}$-trace different from p. Hence p is an $\underline{IN}$-trace.

From (III) follows $\underline{PL}(N) = \langle \underline{Sm}(\underline{PL}(N)) \rangle_{\underline{IN}}$. Since $\underline{Sm}(\underline{PL}(N))$ is a regular language, $\underline{PL}(N)$ is a regular $\underline{IN}$-trace language. ■

Proposition 24, the proof of which was postponed until this Appendix, follows from Proposition 45.

APPENDIX B. Path expressions

The calculus of path expressions was invented by Campbell and Habermann [2]. A path expression defines a partial language whose elements are called legal histories. In fact a path expression expresses the constraints which a legal history must fulfil. Such a constraint may be, for example, that two instructions be mutually exclusive (non-concurrent), or that they must occur alternatingly, etc.

Our explanation of the syntax and semantics of path expressions follows Berzins and Kapur [1]. This involves a slight modification of the original definitions given by Campbell and Habermann.

Definition. Let X be a finite alphabet. We define the notion of a path over X.

(i) If $x \in X$, then x is a path over X.

(ii) If P, P_1, and P_2 are paths over X, then P^*, $P^\times$, $(P_1 \cup P_2)$, and $(P_1 * P_2)$ are paths over X.

Definition. Let X be a finite alphabet. We define the notion of a path expression over X.

(i) If P is a path over X, then P is a path expression over X.

(ii) If PE_1 and PE_2 are path expressions over X, then $(PE_1 \,!\, PE_2)$ is a path expression over X.

Definition. $\underline{K}$ is a mapping of path expressions into partial languages defined as follows:

$\underline{K}(x) := \{x\}$,

$\underline{K}(P^*) := (\underline{K}(P))^*$,

$\underline{K}(P^\times) := (\underline{K}(P))^\times$,

$\underline{K}((P_1 \cup P_2)) := \underline{K}(P_1) \cup \underline{K}(P_2)$,

$\underline{K}((P_1 * P_2)) := \underline{K}(P_1) * \underline{K}(P_2)$,

$\underline{K}((PE_1 ! PE_2)) := \underline{K}(PE_1) \,!\, \underline{K}(PE_2)$.

Definition. Let PE be a path expression over X. The set

$$\underline{H}(PE) := \underline{We}(\underline{K}(PE)!Y^X),$$

where Y is the set of all symbols of X not occurring in PE, is called the set of all legal histories of PE.

($\underline{H}(PE)$ in fact depends on X. This dependence is not expressed in our notation).

A set of legal histories is always a partial language.

Our definition given above is simpler than the one given by Berzins and Kapur [1]. There the symbols (regarded as instruction names) are distinguished by their reference to data objects. Let X_i be the set of symbols that refer to data object N^o i. X is the union of all X_i's. Let g_i be the renaming which erases all symbols not belonging to X_i. A partial word p is a legal history in the sense of Berzins and Kapur iff for each i the partial word $g_i(p)$ is a legal history in our sense. However, this definition is not more general than our definition. One can easily construct, for each PE, a path expression PE_i such that $\underline{K}(PE_i) = g_i(\underline{K}(PE))$. Let i run from 1 to n, then the set of legal histories in the sense of Berzins and Kapur is

$$\underline{We}(\underline{K}(PE_1!\ldots!PE_n)!Y^X),$$

which is a special case of a set of legal histories in our sense.

Proposition B1. Let L be a partial language over X. L is the set of all legal histories of some path expression iff L admits a representation

$$L = \underline{We}(L_1 \; ! \; \ldots \; ! \; L_m),$$

where $L_1,\ldots,L_m$ are partial languages obtained from the singletons $\{x\}$, $x \in X$, by application of union, serial product, serial iteration, and parallel iteration.

P r o o f. (I) Necessity of the condition.

Let $L = \underline{We}(\underline{K}(PE)!Y^X)$. Then $\underline{K}(PE) = \underline{K}(P_1)!\ldots!\underline{K}(P_{m-1})$ for some paths $P_1,\ldots,P_{m-1}$. We set $L_i := \underline{K}(P_i)$ and $L_m := Y^X$.

Sufficiency of the condition: Let $L = \underline{We}(L_1!\ldots!L_m)$. There exists a path expression such that $\underline{K}(PE) = L_1!\ldots!L_m$. Moreover we can achieve that every symbol of X occurs in the expression PE. Then $\underline{H}(PE) = \underline{We}(\underline{K}(PE)) = L$. ∎

The last proposition allows us to compare sets of legal histories with petrinet partial languages.

$\underline{We}\ (\{ab\}^{\times}\{c\})^{*}$ is obviously a set of legal histories. If this partial language is in $\underline{WePL}(PN)$ then the reachability problem for petrinets is unsolvable. (Cf. our discussion following Proposition 17).

If parallel iteration is excluded from the operations available for constructing paths, we obtain the regular path expressions. Any set of legal histories of a regular path expression is in $\underline{WePL}(PN)$, moreover it has the form $\underline{We}(L)$, where L is a regular partial language.

The example $\underline{We}\{a \times a\}$ shows that there are even finite partial languages which are not definable by path expressions. This is due to the defect consisting in parallel product not being available for constructing a path. (Note that $\underline{We}\{a \times b\}$ is definable by a path expression if $a \neq b$, because $\{a \times b\} = \{a\} ! \{b\}$).

APPENDIX C. Concluding Remarks

Partial languages allow us to describe the behaviour of algorithms that in volve concurrent events, at a certain level of abstraction. We have learnt to apply several operations to partial languages. We have found characterizations of several families of partial languages in terms of these operations (in a weak sense for arbitrary petrinets, and in a strong sense for safe petrinets). We have also seen how Mazurkiewicz s trace languages and Campbell and Habermann's path expression languages fit into our concept.

Incidentally we have opened a wide field of open problems. Let us summarize at this point the most important open problems and areas of future research.

(1) As we stated in Section 4, the question "Which is the adequate meaning of firability of a partial word in an arbitrary petrinet?" is open. Adequacy means that the notion of firability should reflect our intuition of the capability of a petrinet (unlike strong firability, cf. Example 8), and that it constitutes a partial language family with reasonable properties. An important criterion is the closedness under Y-section.

(2) Short of a more adequate notion one should study the notion of strong firability (cf. Section 4) more thoroughly. An interesting question is the following: Is $\{p_n : n \in \underline{N}\}$ in $\underline{StPL}(PN)$ or in $\underline{R^eStPL}(PN)$? Here p_n is the partial word over $\{a,b,c\}$ depicted in Figure 17. (Cf. Example 8). Note that this partial language can be obtained from three recursively enumerable languages by parallel iteration, (strong) restriction, and renaming. Since it is hard to prove that $\underline{R^eL}(PN)$ does not contain all recursively enumerable languages (see Greibach [5]), it seems hard to prove that our question posed above has a negative answer.

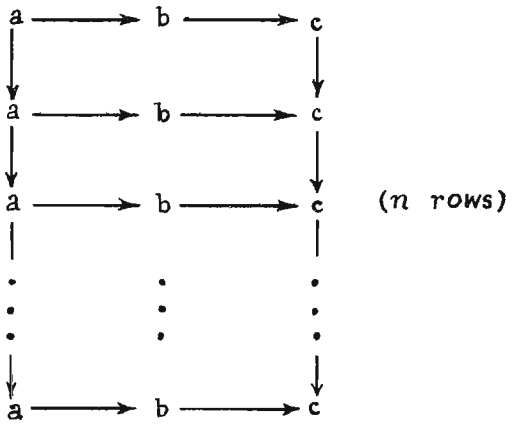


Fig. 17

(3) In our introductory remarks we stated that partial languages reflect the property of a control structure to be determinate. A precise notion of determinacy in terms of partial languages remains to be formulated. It should formalize the idea that a partial word which is "legal" (i.e. which can be generated by the determinate control structure at a fixed initial state) has, in some sense, at most one legal continuation.

Proposition 8 then provides a method of checking whether a control structure composed of simpler components is determinate.

(4) Besides determinacy there are a variety of other properties of control structures which can be formulated in terms of partial languages. Examples are liveness and termination. It is desirable to know the conditions under which these properties are invariant under several operations. This would provide necessary knowledge for the correct synthesis of control structures. It is also desirable to have more operations available than those mentioned in Section 1.

(5) When studying interconnections of control structures one would think of them as capable of an input-output behavior.

Thus we need a generalization of the well-known sequential machine. This is a machine that transforms partial words to partial words. Such a machine could be based on a petrinet. Hence we have a new class of operations on partial languages: transductions definable by various types of machines.

(6) The notion of a partial language is connected with that of the degree of parallelism of a computation. Let the length of a partial word be the length of its longest ordered subset, and let its parallelism degree be the cardinality of its largest totally unordered subset. An important task is to minimize the length of a partial word by maximizing its parallelism degree. A precise formulation of this task should be given. Algorithmic solutions of the task and their effect on partial languages should be studied.

References

[1] B e r z i n s, V., and K a p u r, D., Denotational and axiomatic definitions for path expressions. Computation Structures Group Memo 153-1, Laboratory for Computer Science, M.I.T., November 1977.

[2] C a m p b e l l, R. H., and H a b e r m a n n, A. N., The specification of process synchronization by path expressions. Operating Systems. Lecture Notes in Computer Science vol.16. Springer Verlag, Berlin (West) 1974.

[3] E i l e n b e r g, S., Automata, Languages, and Machines. Vol. A. Academic Press, New York 1974.

[4] G e n r i c h, H., L a u t e n b a c h, K., and T h i a g a r a j a n, P. S., An overview of net theory. Advanced Course on General Net Theory of Processes and Systems, Hamburg 1979.

[5] G r e i b a c h, S. A., Remarks on blind and partially blind one-way multicounter machines. Theoretical Computer Science, 7, 311-324, 1978.

[6] H a c k, M. H. T., Petri net languages. Computation Structures Group Memo 124, Project MAC, M.I.T., 1975.

[7] H a c k, M. H. T., The equality problem for vector addition systems is undecidable. Theoretical Computer Science, 2, 77--95, 1976.

[8] H o a r e, C. A. R., Communicating sequential processes, Communications of the ACM, 21, 8, 1978.

[9] K n u t h, E., Petri nets and trace languages. First European Conference on Parallel and Distributed Processing, Toulouse, 51-56, 1979.

[10] M a z u r k i e w i c z, A., Concurrent program schemes and their verification. DAIMI PB 78, Dept. of Computer Science, Univ. of Aarhus, July 1977.

[11] P e t e r s o n, Petri Nets, ACM computing Surveys.

[12] P e t e r s o n, J. L., Modelling of parallel systems. Ph. D. Thesis, Computer Science, Stanford University, 1974.

[13] P e t e r s o n, J. L., Computation sequence sets. Journal of Computer Aand System Science, 13, 1-24, 1976.

[14] P e t e r s o n, J. L., and B r e d t, T. H., A comparison of models of parallel computation, Information Processing '74. North-Holland Publishing Co., Amsterdam 1974.

[15] S t a r k e, P. H., Free Petri net languages. Seminarbericht Nr. 7, Sektion Mathematik, Humboldt-Universität Berlin, Januar 1978.

[16] S t a r k e, P. H., Petri-Netze. ZKI-Information 3/1978, Zentralinstitut für Kybernetik und Informationsprozesse, Akademie der Wissenschaften der DDR, Juni 1978.

[17] W i n k o w s k i, J. An algebraic characterization of the behaviour of non-sequential systems. Information Processing Letters, 6, 4, 1977.

[18] W i n k o w s k i, J., An algebraic approach to concurrence. Mathematical Foundations of Computer Science 1979. Lecture Notes in Computer Science, vol. 74. Springer Verlag, Berlin (West) 1979, 523-532.