8272 COPY 3

MAKING NETS ABSTRACT AND STRUCTURED

and

NETS AND THEIR RELATION TO C S P

Ludwik Czaja

Technical Monograph PRG-38

January 1984 and June 1984

Oxford University Computing Laboratory
Programming Research Group
8-11 Keble Road
Oxford OX1 3QD

Institute of Informatics
University of Warsaw
Poland

on leave at

Oxford University Computing Laboratory
Programming Research Group
8 - 11 Keble Road
Oxford OX1 3QD

MAKING NETS ABSTRACT AND STRUCTURED

Ludwik CZAJA

Oxford University Computing Laboratory
Programming Research Group
8-11 Keble Road, OXFORD OX1 3QD, U.K.

(on leave from Institute of Informatics
University of Warsaw, Poland)

January 1984

1. Introduction

By abstract net is meant here a one of the Petri Nets' structure but of quite general interpretation: arbitrary objects may be assigned to places and arbitrary transformations on "markings" – to transitions. Certainly various sorts of Petri Nets may be expressed in this setting, by specifying a particular interpretation. But also such structures as arithmetic or boolean expressions, sequential flowchart schemata, data-flow systems etc. can be represented as abstract nets. The representation however involves pictures – amorphic collections of lines usually, hardly intelligible perhaps apart from simple structures, like trees. But there is a way of structuring large nets from simple, easy to understand parts, by making use of suitably chosen operators on nets. We chose here a concurrency operator "‖", corresponding to that for CSP [Hoa 81] and, in a sense, inverse to it – a subtraction operator "\". In Section 3 there is a simple example of net construction by means of these operators. Section 4 is concerned with decomposition of nets wrt concurrency operator (‖-factorisation). It turns out that some nets, Petri Nets, for example, may be ‖-factorised in any possible way, but usually this is not so with some other interpretations. Theorem 4.3 establishes a necessary and sufficient condition for an abstract net to be ‖-factorisable wrt a given partition of places. Theorem 4.4 states uniqueness of an ultimate ‖-factorisation, the factorisation into *atomic* not decomposable subnets. Section 5 contains an algebraic note, and suggests a linear notation for nets.

This work was supported by a Visiting Fellowship Research Grant from the UK Science and Engineering Research Council

2. Preliminaries

2.1 General denotations

$\{x_1, x_2, \ldots, x_n\}$	- the set of elements $x_1, x_2, \ldots, x_n$
$\langle x_1, x_2, \ldots, x_n \rangle$	- the n-tuple of these elements
$\{x \in X \colon G\}$	- the set of all x from X satisfying a predicate G
$(S_1, S_2, \ldots, S_n)$	- partition of a set S; thus, $S_i \cap S_k = \emptyset$ for $i \neq k$ and $\bigcup_i S_i = S$; $\bigcup_i$ stands for $\bigcup_{i=1}^{n}$
$f\colon A \to B$	- total function from A into B
$f\colon A \twoheadrightarrow B$	- partial function from A into B
$[\alpha_1 \to E_1 \mid \alpha_2 \to E_2 \mid \ldots \mid \alpha_n \to E_n]$	- conditional expression with conditions $\alpha_1, \alpha_2, \ldots, \alpha_n$ and expressions $E_1, E_2, \ldots, E_n$

Relations and their restrictions. If A, B are sets then r is a relation provided that $r \subseteq A \times B$. A restriction of r to a set $A' \subseteq A$ is

$r|A' = r \cap A' \times B$, and to the set $B' \subseteq B$ is

$r|B' = r \cap A \times B'$

The same definition applies to functions $f : A \to B$ or $f : A \twoheadrightarrow B$ considered as single-valued relations. By analogy, if $r \subseteq A \times B \cup B \times A$, i.e. r is a bipartite relation, then

$r|A' = r \cap (A' \times B \cup B \times A')$

$r|B' = r \cap (A \times B' \cup B' \times A)$

We write $r|a$, $r|b$ instead of $r|\{a\}$, $r|\{b\}$ $(a \in A,\ b \in B)$

2.2 Abstract nets

An abstract net is a system $P = \langle\langle S,T,F\rangle,A,I\rangle$ where S is a finite set of places, drawn as circles, T is a finite set of transitions, drawn as bars, $F \subseteq S\times T \cup T\times S$, a bipartite relation, is a set of arrows going from places to transitions or from transitions to places. If multiplicities of arrows are required, then F may be regarded as a function $F: S\times T \cup T\times S \to \{0,1,2,..\}$ For nets considered here, we assume $S \neq \emptyset$. The structure $\langle S,T,F\rangle$ is called a net-schema and this is exactly as in Petri nets. The interpretation however, is quite abstract: A is an arbitrary set (in Petri nets $A = \{0,1,2,...\}$). $M: S \to A$ is an arbitrary total function called a marking of the net P. $\mathbf{M} = S \to A$ is the set of all markings. I is a mapping which with every transition $t\in T$ associates a binary relation in $\mathbf{M}$ i.e. $I(t) \subseteq \mathbf{M}\times\mathbf{M}$. $I(t)$ will be written t^I and will be called interpretation of t. Transition t is firable at a marking M if there exists M′ such that Mt^IM' and in this case M′ is a next marking following M, resulted from firing t. A sequence of markings:

$M_0,M_1,M_2,...$ and transitions:

$t_0,t_1,t_2,...$

are said to be a computation sequence and a firing sequence respectively, if for $i=0,1,2,...$, transition t_i is arbitrarily selected, firable in M_i and $M_it_i^IM_{i+1}$. For $j>i$, M_j is reachable from M_i through a firing sequence $t_i, t_{i+1},...,t_{j-1}$ and, by convention, M_i is reachable from M_i through the empty sequence. M′ is reachable from M iff M′ is reachable from M through a certain firing sequence $v=t_0...t_n$; we write then $M\xrightarrow{v}M'$. A language generated by a net P from a marking M_0 is:

$$L(M_0,P) = \{v\in T^*: \exists M.M_0\xrightarrow{v}M\}$$

We assume here t^I to be a partial function $t^I: \mathbf{M}\to\mathbf{M}$ and thus write $M' = t^I(M)$ whenever Mt^IM'. If t^I is undefined for $M \in \mathbf{M}$ i.e. t^I is not firable at M, it is written $t^I(M) = \perp$ and we assume $\perp \notin \mathbf{M}$. For reasons made clear further, two conventions are admitted: $M \cup \perp = \perp \cup M = \perp$ and $\perp|S_0 = \perp$ for any $M\in\mathbf{M}$ and $S_0\subseteq S$. Example: for Petri's place/transition nets with "weak firing rule", the t^I relation is defined by

$$Mt^IM' \iff \forall s\in S.M(s)\geq F(s,t) \wedge M'(s)-M(s)=F(t,s)-F(s,t)$$

and this indeed is a partial function.

2.3 Parallel combination

Let $P = \langle\langle S_P,T_P,F_P\rangle,A_P,I_P\rangle$ and $Q = \langle\langle S_Q,T_Q,F_Q\rangle,A_Q,I_Q\rangle$.

Assuming $S_P \cap S_Q = \emptyset$, a parallel combination $R = P\|Q$ of P and Q is a net

$R = \langle\langle S_R,T_R,F_R\rangle,A_R,I_R\rangle$, where:

$S_R = S_P \cup S_Q$, $T_R = T_P \cup T_Q$, $F_R = F_P \cup F_Q$, $A_R = A_P \cup A_Q$

and the interpretation I_R is defined by:

$$t^{I_R}(M_R) = \begin{cases} t^{I_P}(M_P) \cup M_Q & \text{if } t \in T_P - T_Q \\ t^{I_Q}(M_Q) \cup M_P & \text{if } t \in T_Q - T_P \\ t^{I_P}(M_P) \cup t^{I_Q}(M_Q) & \text{if } t \in T_P \cap T_Q \end{cases}$$

where marking of $R = P \| Q$ is $M_R = M_P \cup M_Q$. Operation $\|$ will also be referred to as a concurrency operation.

Notes

(a) $M_R = M_P \cup M_Q$, the union of functions, is understood as the union of relations. Due to $S_P \cap S_Q = \emptyset$, M_R again is a function - this motivates the assumption! The reason for convention $M \cup \bot = \bot \cup M = \bot$ is also evident: transition t should be firable in the net R provided that t is firable in this constituent P and/or Q of R to which t belongs.

(b) If $T_P \cap T_Q = \emptyset$ then $R = P \| Q$ works as P and Q in parallel and independently of each other: P and Q are entirely loosely coupled nets. If $T_P \cap T_Q \neq \emptyset$ then P and Q synchronise mutually on transitions from $T_P \cap T_Q$. The opposite extreme of coupling is when $T_P = T_Q$: P and Q are then entirely tightly coupled nets.

(c) Operation $\|$ is associative and commutative, so we use

$\|_{i=1}^{n} P_i$ to denote $P_1 \| P_2 \| \ldots \| P_n$, provided

that $S_i \cap S_j = \emptyset$, for $i \neq j$. Similarly, if $\{P_z\}$ $(z \in Z)$ is an indexed family of nets with disjoint sets of places, then by

$\|_{z \in Z} P_z$ is denoted the parallel combination of all P_z.

(d) If $R = P \| Q$ then $L(M_R, R) = L(M_P, P) \| L(M_Q, Q)$,

where the operation "$\|$" on languages is the parallel combination of processes from the model for CSP [Hoa 84].

2.4 Neighbourhood

For a given net $P = \langle\langle S,T,F\rangle,A,I\rangle$, we use the notation:
$^\bullet s^\bullet = {}^\bullet s \cup s^\bullet$, $^\bullet t^\bullet = {}^\bullet t \cup t^\bullet$, where

$^\bullet s = \{t \in T: F(t,s)\}$, $s^\bullet = \{t \in T: F(s,t)\}$,

$^\bullet t = \{s \in S: F(s,t)\}$, $t^\bullet = \{s \in S: F(t,s)\}$

So, $^\bullet s^\bullet$ and $^\bullet t^\bullet$ denote the neighbourhood of place s and transition t respectively, with no mention to which net it is related. This is satisfactory as long as one net was fixed for consideration, but is no longer. If a transition t belongs to several nets combined by $\|$ operation. It is then necessary to indicate in which net the neighbourhood of t is considered. We introduce notation:

$$nbh(t,P) = \{s \in S_P: F_P(s,t) \vee F_P(t,s)\}$$

$$nbh(s,P) = \{t \in T_P: F_P(s,t) \vee F_P(t,s)\}$$

$$nbh(S_0,P) = \bigcup_{s \in S_0} nbh(s,P) \qquad \text{for} \quad S_0 \subseteq S$$

However, we will retain the "dot notation" if there is no ambiguity.

2.5 Subtraction – an inverse to concurrency operation

In Section 3 we will make a modest use of an operation inverse, in a sense, to "$\|$". Let

$P = \langle\langle S_P,T_P,F_P\rangle,A_P,I_P\rangle$, $Q = \langle\langle S_Q,T_Q,F_Q\rangle,A_Q,I_Q\rangle$. $R = P\backslash Q$ is a net

$R = \langle\langle S_R,T_R,F_R\rangle,A_R,I_R\rangle$, with:

$S_R = S_P - S_Q$, $T_R = nbh(S_R,P)$, $F_R = F_P - F_Q$,

$A_R = A_P - A_Q$, $M_R = M_P - M_Q$,

$$t^{I_R}(M_R) = t^{I_P}(M_P) - t^{I_Q}(M_Q)$$

and by convention

$$\perp - M = M - \perp = \perp$$

The subtraction allows to remove unnecessary subnets from nets constructed by parallel combination.

3. Net construction - a familiar example

Concurrency operation ‖ suggests constructing large, mentally unmanagable nets from small, easy to understand components, each of which models a meaningful object. As an example, consider Five Dining Philosophers. Although simple, it clearly displays the idea of structuring. Three versions of the problem are shown and we assume the ordinary Petri net interpretation

Version 1.

The behaviour of i-th fork is modelled by the net in Fig.3.1 and of i-th philosopher by the net in Fig.3.2. The philosophers are numbered 0,1,2,3,4 clockwise. $\oplus$, $\ominus$ mean addition and subtraction modulo 5, fork i is on the left of i-th philosopher, fork i⊕1 on his right, so transition *i pick i* causes picking by i-th philosopher his left fork etc. In Fig.3.3 the whole net called TABLE' is shown:

$$\mathrm{TABLE'} = \|_{i=0}^{4}(\mathrm{FORK}_i \| \mathrm{PH}_i)$$

from which shaded places, as superfluous can be removed as follows. Let nets DL_i and DR_i be as in Fig 3.4, then

$$\mathrm{TABLE} = \mathrm{TABLE'} \backslash \|_{i=0}^{4}(\mathrm{DL}_i \| \mathrm{DR}_i)$$

which is shown in Fig 3.5 This net is obviously deadlock-prone: the deadlock occurs if all philosophers hold their left or right forks To avoid deadlock, a butler may be called for help:

$$\mathrm{BUTLER} = \mathrm{LEFT} \| \mathrm{RIGHT}$$

where nets LEFT (RIGHT), shown in Fig.3.6, prevent the state in which every philosopher holds its left (right) fork. So, the deadlock-free net is:

$$\mathrm{DEADLOCKFREETABLE} = \mathrm{TABLE} \| \mathrm{BUTLER}$$

Version 2.

Here, $FORK_i$ is as in Version 1. The net for i-th philosopher is in Fig.3.7. The net TABLE, obtained in the same way as in Version 1 (after removing superfluous places), is in Fig.3.8. Although this is the deadlock-free net, so no butler is needed, the livelock may occur: a philosopher can sit down, then pick up the one fork, then put it down, then pick up the other fork, then put it down, then get up without having eaten, then again sit down, etc. to infinity.

Version 3.

Here again, $FORK_i$ is as in Version 1. The net for i-th philosopher is in Fig 3.9. The net TABLE, obtained in the same way as in Version 1, is in Fig.3.10. This is a deadlock-prone net, so the butler should be applied.

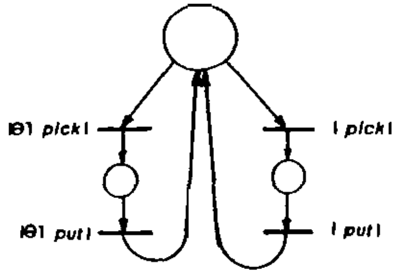


Fig.3.1 $FORK_1$

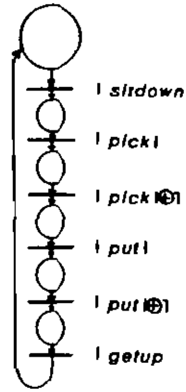


Fig.3.2 PH_1

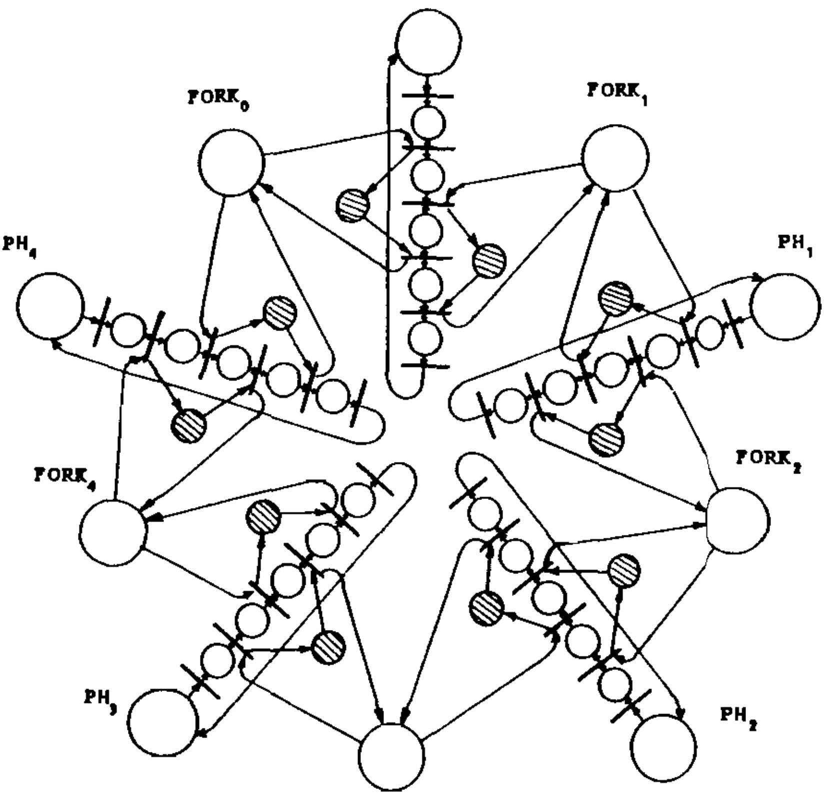


Fig.3.3 TABLE'

Fig.3.4

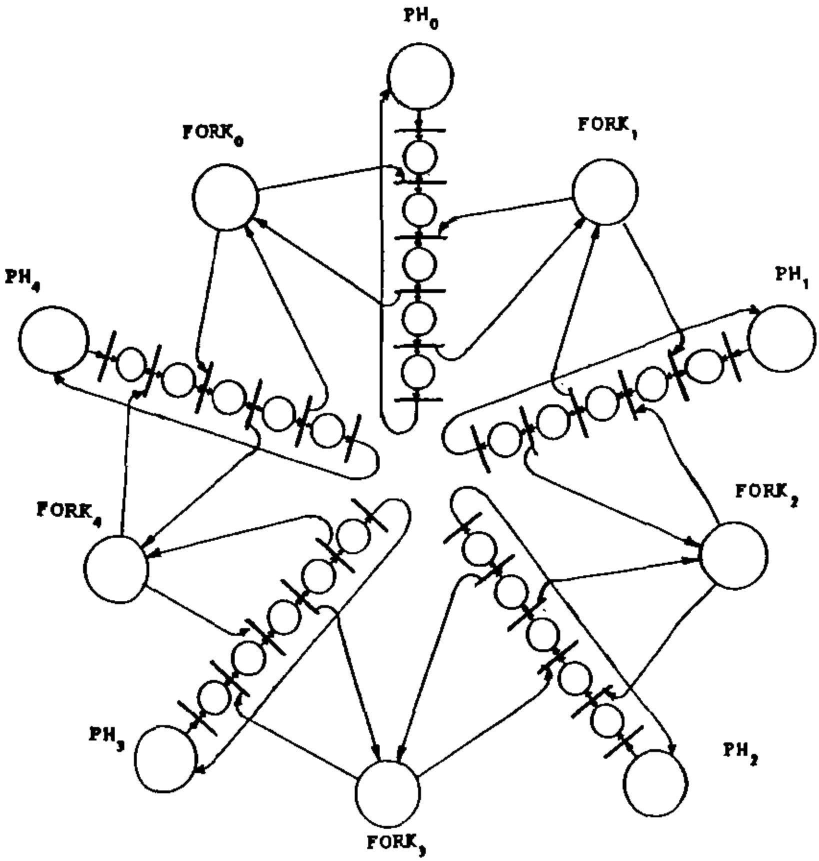
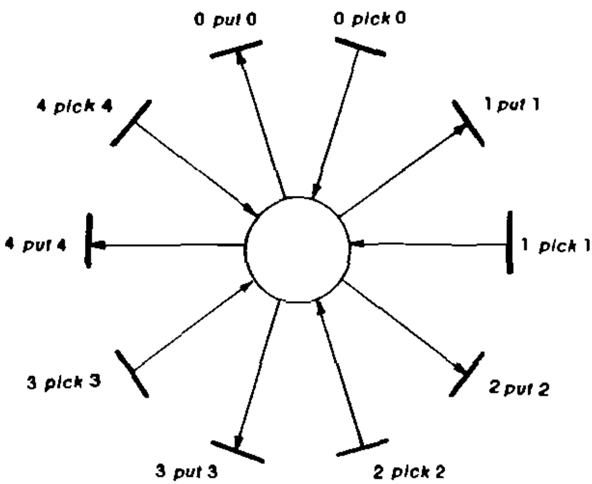
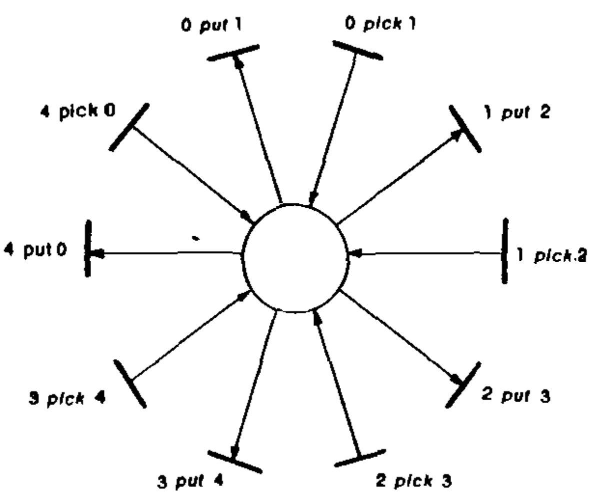


Fig.3.5 TABLE



LEFT, capacity of the piece = 4



RIGHT, capacity of the piece = 4

Fig.3.6 BUTLER

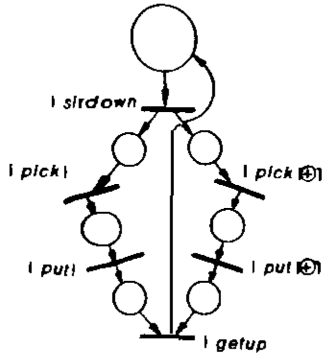


Fig.3.7 PH_i

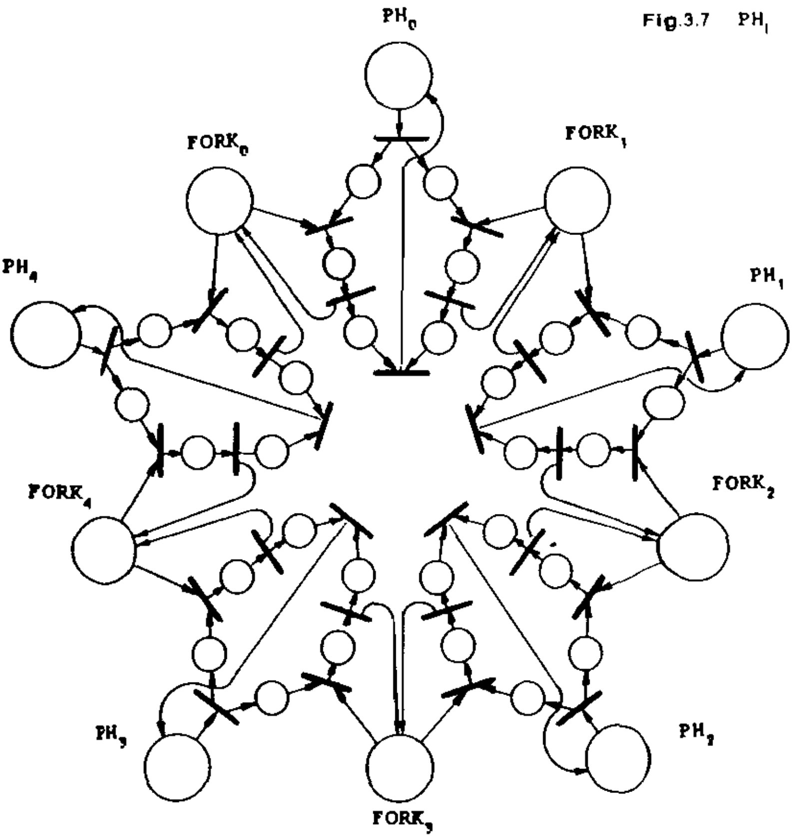


Fig.3.8 TABLE

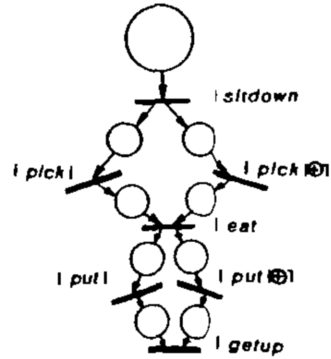


Fig.3.9 PH_i. Arrow from I getup to the top place is invisible.

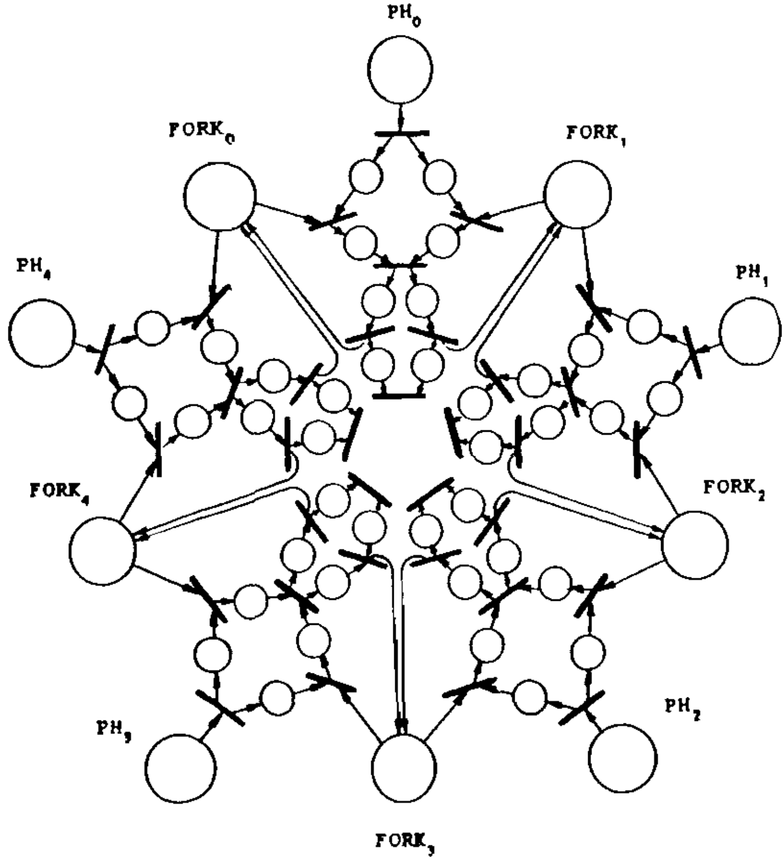


Fig 3.10 TABLE

4. Net decomposition

This section is concerned with decomposition of systems specified as abstract nets, into subsystems working in parallel: given net P, we look for nets $P_1, \ldots, P_n$ such that $P = \|_{i=1}^{n} P_i$. It turns out that, although net-schemas ⟨S,T,F⟩ can obviously be decomposed in as many ways as there are partitions of the set S, this is not so with interpretation. Nets in Petri's interpretation, for instance, can be decomposed in every way, even into single-place subnets (Example 4.2), nets computing arithmetic expressions cannot be decomposed at all (Example 4.3), whilst some others can, but only wrt specific partitions of places, S-partitions, in short (Example 4.4). Such decomposition, or ‖-factorisation, may usually be done in many ways, but it is unique for a fixed S-partition (Theorem 4.2). Theorem 4.3 gives a necessary and sufficient condition for a net to be decomposable wrt a given S-partition. Its easy proof is due to Theorem 4.1, considerably simplifying definition of ‖-operator. Theorem 4.1, in turn, follows directly from two natural properties, assumed as axioms for interpretation: Axiom (i) states that a transition attached to no place is firable regardless of marking, Axiom (ii) – that the effect of firing a transition confines to its neighbourhood ("locality axiom"). The section is concluded by introducing a canonic ‖-factorisation. Irrespective of S-partitions Theorem 4.4 states the uniqueness of this particular decomposition of nets into *atomic* subnets.

Axioms for interpretation

(i) If $\cdot t \cdot = \emptyset$ then $t^I(M) \neq \perp$

(ii) If $t^I(M) \neq \perp$ then $t^I(M)|S-\cdot t\cdot = M|S-\cdot t\cdot$

Lemma 4.1

Suppose interpretations of nets satisfy Axioms (i) (ii). Then the definition of interpretation I_R, introduced in Section 2.3 for the composite net $R = P\|Q$ simplifies to:

$$t^{I_R}(M_R) = t^{I_P}(M_P) \cup t^{I_Q}(M_Q)$$

Proof

It suffices to show that

$$t \notin T_P \Rightarrow t^{I_P}(M_P) = M_P, \quad t \notin T_Q \Rightarrow t^{I_Q}(M_Q) = M_Q$$

Suppose $t \notin T_P$. Then $nbh(t,P) = \emptyset$ (otherwise, there would exist $s \in nbh(t,P) \subseteq S_P$; but this is equivalent to $t \in nbh(s,P) \subseteq T_P$). By Axiom (i): $t^{I_P}(M_P) \neq \perp$ and by (ii): $t^{I_P}(M_P) = M_P$. This is shown analogously for Q.

q.e.d

From Lemma 4.1, by induction, the following useful theorem is obtained:

Theorem 4.1

Let $P_j = \langle\langle S_j, T_j, F_j\rangle, A_j, I_j\rangle$, $j=1,\dots,n$ be given nets with $S_k \cap S_j = \emptyset$ for $k \neq j$ and with interpretations I_j satisfying Axioms (i) (ii). Suppose $P = \Vert_{j=1}^{n} P_j$. Then

$t^I(M) = \bigcup_j t^{I_j}(M_j)$ for $M = \bigcup_j M_j$ and I – the interpretation in P.

This theorem, a direct consequence of Axioms (i) (ii), allows for a very simple construction of a net $P = \Vert_{j=1}^{n} P_j$ given P_j: the sets S, T, F, A, M, t^I for P are just unions of respective sets for P_j.

Definition 4.1

A net $P = \langle\langle S,T,F\rangle, A, I\rangle$ is decomposable wrt a partition $\{S_1,\dots,S_n\}$ of S iff there are nets $P_j = \langle\langle S_j, T_j, F_j\rangle, A_j, I_j\rangle$ $(j=1,\dots,n)$ such that $P = \Vert_{j=1}^{n} P_j$

Theorem 4.2

If a net P is decomposable wrt a partition of (the set of) its places then the decomposition is unique; strictly speaking, unique up to non-isolated transitions.

Proof

Suppose for $i=1,2$ and $k=1,\dots,n$: $P_{ik} = \langle\langle S_{ik}, T_{ik}, F_{ik}\rangle, A_{ik}, I_{ik}\rangle$ are nets such that

$S_{ik} = S_k \quad S = \bigcup_k S_{ik} \quad T = \bigcup_k T_{ik} \quad F = \bigcup_k F_{ik}$

$A = \bigcup_k A_{ik}$ and (due to Theorem 4.1) $t^I = \bigcup_k t^{I_{ik}}$

To be proved:

(1) $S_{1k} = S_{2k}$ (2) $T_{1k} = T_{2k}$ (3) $F_{1k} = F_{2k}$

(4) $A_{1k} = A_{2k}$ (5) $t^{I_{1k}} = t^{I_{2k}}$ for $k=1,\dots,n$.

Points (1) and (4) are obvious (one may assume $A_{ik} = A$). Points (3) and (5) are readily obtained: $\bigcup_j F_{1j} = \bigcup_j F_{2j}$ implies $(\bigcup_j F_{1j})|S_k = (\bigcup_j F_{2j})|S_k$ which with

$(\bigcup_j F_{ij})|S_k = F_{ik}$ implies (3); the same reasoning applies to $t^{I_{ij}}$ bringing (5). To check (2), note that if isolated transitions are not taken into account then

$T_{ik} = nbh(S_{ik}, P_{ik}) = \bigcup_{s \in S_{ik}} nbh(s, P_{ik})$ which, by definition of $nbh(s, P_{ik})$, by (1) and (3) brings (2).

Definition 4.2

A partition $\{S_1,\ldots,S_n\}$ of S is functional wrt a net $P=\langle\langle S,T,F\rangle,A,I\rangle$ if for every $t\in T$:

(a) If $M|S_k = M'|S_k$ then $t^I(M)|S_k = t^I(M')|S_k$

provided that $t^I(M) \neq \perp \iff t^I(M') \neq \perp$ for any M, M', $k=1,\ldots,n$

(b) If $t^I(M^k) \neq \perp$ for $k=1,\ldots,n$ then $t^I(M) \neq \perp$

where $M = \bigcup_k M^k|S_k$

Example 4.1

A "RELAY" passing numbers from input (in) to output (out) provided that a binary-valued control (c) holds 1, is specified as

```
RELAY = <<{in,out,c}, {t}, {<in,t>,<t,out>,<c,t>}>,R,I>    with
t^I(M) = ⊥  ⇔  M(c) = 0
t^I(M)(s) = [s=in v s=c → 0 | s=out → M(in)]
```

(R denotes the set of real numbers). The only functional partition of {in,out,c} wrt RELAY is {{c},{in,out}}.

Theorem 4.3

A net $P = \langle\langle S,T,F\rangle,A,I\rangle$ is decomposable wrt a partition $\{S_1,\ldots,S_n\}$ of S iff this is a functional partition wrt P.

Proof

Let P be decomposable wrt $S_1,\ldots S_n$.

Then, there exist nets $P_j = \langle\langle S_j,T_j,F_j\rangle,A_j,I_j\rangle$ such that $P = \|_{j=1}^n P_j$. Applying Theorem 4.1 we obtain

(*) $t^I(M) = \bigcup_j t^{I_j}(M|S_j)$ for any marking M in the net P.

Note that $M|S_j$ is a marking in the net P_j. Since t^{I_k} are functions, then

(**) $M|S_k = M'|S_k \implies t^{I_k}(M|S_k) = t^{I_k}(M'|S_k)$ for any markings M, M' in P.

Suppose $t^I(M) \neq \perp$, $t^I(M') \neq \perp$ and $M|S_k = M'|S_k$.

Then, by (*) $t^{I_j}(M|S_j) \neq \perp$ and similarly $t^{I_j}(M'|S_j) \neq \perp$

Thus, by (*) and property of restriction $\cdot|\cdot$:

$t^I(M)|S_k = (\bigcup_j t^{I_j}(M|S_j))|S_k = t^{I_k}(M|S_k)$

$t^I(M')|S_k = (\bigcup_j t^{I_j}(M'|S_j))|S_k = t^{I_k}(M'|S_k)$

By (**) we get $t^I(M)|S_k = t^I(M')|S_k$

Now, suppose $t^I(M) = \perp$, $t^I(M') = \perp$ and $M|S_k = M'|S_k$.

Clearly, by convention from Section 2.3, also $t^I(M)|S_k = t^I(M')|S_k$.

so, we proved that point (a) in Definition 4.2 holds. To prove (b)

suppose $t^I(M^k) \neq \perp$ and $M = \bigcup_k M^k|S_k$. Therefore $M|S_j = M^j|S_j$, $j=1,\dots n$

By Theorem 4.1: $t^I(M) = \bigcup_j t^{I_j}(M|S_j) = \bigcup_j t^{I_j}(M^j|S_j)$

and $\perp \neq t^I(M^k) = \bigcup_j t^{I_j}(M^k|S_j)$, for $k=1,\dots,n$, which implies $t^I(M) \neq \perp$

Let $(S_1,\dots S_n)$ *be a functional partition wrt* P.

We look for $P_k = \langle\langle S_k,T_k,F_k\rangle,A_k,I_k\rangle$ such that $P = \|_{k=1}^{n} P_k$.

Define: $F_k = F|S_k$, $T_k = nbh(S_k,P) \cup ISOL_P$ where

$ISOL_P = \{t \in P: t \notin nbh(S,P)\}$. (so, $ISOL_P$ is the set of transitions isolated in P, i.e. connected to no place $s \in S$). $A_k = A$.

$$t^{I_k}(M_k) = \begin{cases} t^I(M)|S_k, & \text{where } M \text{ is arbitrary marking in } P, \text{ satisfying} \\ & M|S_k = M_k, \text{ and } t^I(M) \neq \perp, \text{ iff such } M \text{ exists} \\ \perp, & \text{otherwise} \end{cases}$$

Due to (a) in Definition 4.2. $t^{I_k}(M_k)$ does not depend on the choice of M.

We show $P = \|_{k=1}^{n} P_k$. Evidently, $S = \bigcup_k S_k$, $F = \bigcup_k F_k$, $T = \bigcup_k T_k$,
$A = \bigcup_k A_k$. To check that for given markings M_k in P_k, $M = \bigcup_k M_k$ implies:

(***) $t^I(M) = \bigcup_k t^{I_k}(M_k)$, consider two cases:

Case $t^{I_k}(M_k) \neq \perp$ for every $k=1,\ldots,n$. By definition of t^{I_k}:

(****) $t^{I_k}(M_k) = t^I(M^k)|S_k$ where $M^k: S \to A$ is a certain marking in P satisfying $M^k|S_k = M_k$, and $t^I(M_k) \neq \perp$. Since $M|S_k = M^k|S_k$, then applying points (a) and (b) from Definition 4.2 and (****) we get $t^I(M)|S_k = t^I(M^k)|S_k = t^{I_k}(M_k)$ which implies (***)

Case $t^{I_j}(M_j) = \perp$ for a certain j. By definition of t^{I_j}, this means that $t^I(M) = \perp$ for each marking satisfying $M|S_j = M_j$. Thus, $t^I(M) = \perp$ for $M = \bigcup_k M_k$. On the other hand, $\bigcup_k t^{I_k}(M_k) = \perp$ thus equation (***) holds also in this case. This completes the proof of the theorem.

q.e.d

Example 4.2

Petri Nets (and their extensions like nets with multiplicities on arrows, with inhibiting arrows etc.) are decomposable wrt arbitrary partition of places. The reason is the following. The essential feature of any sort of Petri interpretation is that, although firability of a transition depends usually on several places, marking of a place after firing depends *solely* on its marking before the firing. Thus,

$M|s = M'|s \Rightarrow t^I(M)|s = t^I(M')|s$ (provided that t is firable in M iff it is firable in M') for any place s, transition t and markings M, M'. This implies (a) in Definition 4.2, for any partition $\{S_1,\ldots,S_n\}$. Holding of (b) is obvious. Therefore, arbitrary partition is functional wrt any Petri net. Hence, by Theorem 4.3 – our conclusion.

Example 4.3

Let a net be a tree representing an arithmetic expression, places hold numbers, transitions are operators +, × etc. Function t^I replaces a contents of t's output place by the result of corresponding arithmetic operation on t's inputs, leaving them unchanged. Such nets are not decomposable, regardless of a partition of places. Let us demonstrate this on the net ADD for x+y. Let z be the root of the tree for x+y and let transition t be +. So, ADD is specified as:

$$\mathrm{ADD} = \langle\langle\{x,y,z\},\ \{t\},\ \{\langle x,t\rangle,\langle y,t\rangle,\langle t,z\rangle\}\rangle,R,I\rangle \quad \text{with}$$
$$t^I(M) \neq \perp \quad \text{for all } M$$
$$t^I(M)(s) = [s=x \lor s=y \to M(s) \mid s=z \to M(x)+M(y)]$$

Suppose there is a functional partition $\{S_1,S_2,\ldots\}$ of $\{x,y,z\}$ and let $z\in S_k$. Then either $x\notin S_k$ or $y\notin S_k$. Let $x\notin S_k$ and consider markings M, M':

$M = \{\langle x,1\rangle,\langle y,2\rangle,\langle z,0\rangle\}$ $\qquad$ $M' = \{\langle x,2\rangle,\langle y,2\rangle,\langle z,0\rangle\}$

Therefore

(1) $M|S_k = M'|S_k$ $\qquad$ (2) $M(x)+M(y) \neq M'(x)+M'(y)$

The partition is functional, therefore, by (1) we have

$t^I(M)|S_k = t^I(M')|S_k$ which implies (3) $t^I(M)(z) = t^I(M')(z)$

By specification of t^I, (3) contradicts (2), hence there is no functional partition of $\{x,y,z\}$. By Theorem 4.3, the net ADD cannot be decomposed at all.

Example 4.4

By Theorem 4.3, the only ||-factorisation of RELAY (Example 4.1) is:

RELAY = V||C with value (V) and control (C) factors specified as.

$$V = \langle\langle\{in,out\},\ \{t\},\ \{\langle in,t\rangle,\langle t,out\rangle\}\rangle,\ R,\ I_V\rangle \quad \text{with}$$

$$t^{I_V}(M_V) \neq \perp \quad \text{for all } M_V$$

$$t^{I_V}(M_V)(s) = [s=in \rightarrow 0 \mid s=out \rightarrow M_V(in)]$$

$$C = \langle\langle\{c\},\ \{t\},\ \{\langle c,t\rangle\}\rangle,\ \{0,1\},\ I_C\rangle \quad \text{with}$$

$$t^{I_C}(M_C) = \perp \iff M_C(c) = 0$$

$$t^{I_C}(M_C)(c) = 0$$

Now, one can look for an ultimate decomposition of nets, i.e. decomposition into a sort of *atomic* subnets, not further decomposable. It turns out to be unique, so we get a canonic representation of nets: every net is a parallel composition of a number of atomic nets.

Definition 4.3

A $\|$-factorisation $P = \|_{j=1}^{n} P_j$ is *atomic* iff none of P_j is $\|$-factorisable (regardless of partition of its places). Such P_j is called the *atomic* net.

Theorem 4.4

The atomic $\|$-factorisation of any net is unique, i.e. if $\|_{j=1}^{n} P_j^1$ and $\|_{j=1}^{m} P_j^2$ are two atomic $\|$-factorisations of P then $n = m$ and $P_1^1, \ldots, P_n^1$ is a permutation of $P_1^2, \ldots, P_m^2$.

Proof

Let $\|_{j=1}^{n} P_j^1$, $\|_{j=1}^{m} P_j^2$ be two distinct atomic $\|$-factorisations of P, with corresponding S-partitions $(S_1^1, \ldots, S_n^1)$, $(S_1^2, \ldots, S_m^2)$.

By Theorem 4.2, $(S_1^1, \ldots, S_n^1) \neq (S_1^2, \ldots, S_m^2)$, which means that there exist distinct and non-disjoint S_k^1 and S_i^2. Factorisations are atomic, so P_k^1 is not $\|$-factorisable, thus, no partition of S_k^1, in particular $(S_k^1 \cap S_i^2,\ S_k^1 - S_i^2)$ may be functional wrt P_k^1 (by Theorem 4.3).

Thus, there exist markings M, M' and a transition t firable in M and M' such that either:

$$(1) \quad M|S_k^1 \cap S_i^2 = M'|S_k^1 \cap S_i^2$$

$$(2) \quad t^I(M)|S_k^1 \cap S_i^2 \neq t^I(M')|S_k^1 \cap S_i^2$$

or:

$$(3) \quad M|S_k^1 - S_i^2 = M'|S_k^1 - S_i^2$$

$$(4) \quad t^I(M)|S_k^1 - S_i^2 \neq t^I(M')|S_k^1 - S_i^2$$

If (1), (2) hold then let M_1 be a marking coinciding with M on S^1_k and with M′ on $S-S^1_k$. Hence, by (1), M_1 coincides with M′ on S^2_i

Partition $(S^1_k, S-S^1_k)$ is functional wrt P, thus $t^I(M_1) \neq \perp$

(by (b) in Definition 4.2). Partitions $(S^1_1, \ldots, S^1_n)$, $(S^2_1, \ldots, S^2_m)$ are functional wrt P, thus (by (a) in Definition 4.2),

$$t^I(M_1)\,|\,S^1_k = t^I(M)\,|\,S^1_k$$

$$t^I(M_1)\,|\,S^2_i = t^I(M')\,|\,S^2_i$$

Therefore $t^I(M)\,|\,S^1_k \cap S^2_i = t^I(M')\,|\,S^1_k \cap S^2_i$

which is in contradiction with (2)

If (3), (4) hold then let M_2 be a marking coinciding with M on

$S^1_k \cup S^2_i$ and with M′ on $S-(S^1_k \cup S^2_i)$. Hence, M_2 coincides with M

on S^1_k and, by (3), with M′ on $S-S^2_i$. Partition $(S^1_k \cup S^2_i, S-(S^1_k \cup S^2_i))$

is functional wrt P, thus $t^I(M_2) \neq \perp$ (by (b) in Definition 4.2)

Partitions $(S^1_1, \ldots, S^1_n)$, $(S^2_i, S-S^2_i)$ are functional wrt P, thus (by (a) in Definition 4.2)

$$t^I(M_2)\,|\,S^1_k = t^I(M)\,|\,S^1_k$$

$$t^I(M_2)\,|\,S-S^2_i = t^I(M')\,|\,SS-S^2_i$$

These equations imply:

$$t^I(M_2)\,|\,S^1_k-S^2_i = t^I(M)\,|\,S^1_k-S^2_i$$

$$t^I(M_2)\,|\,S^1_k-S^2_i = t^I(M')\,|\,S^1_k-S^2_i$$

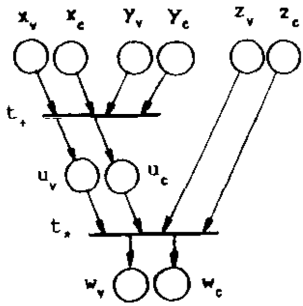
Therefore $t^I(M)\,|\,S^1_k-S^2_i = t^I(M')\,|\,S^1_k-S^2_i$

which is in contradiction with (4)

q.e.d.

Example 4.5

A small data-flow system is represented by the net in Fig.4.1. This is a computation of arithmetic expression x+y*z. Places holding values and control tokens are labelled with a letter subscripted by v and c respectively. Transitions are labelled with t subscripted by corresponding operators.



$$t_+^I(M)\neq\perp \iff M(x_c)=M(y_c)=1,\ M(u_c)=0$$

$$t_*^I(M)\neq\perp \iff M(u_c)=M(z_c)=1,\ M(w_c)=0$$

$$t_+^I(M)(s)=[s\in\{x_v,y_v,z_v,w_v,z_c,w_c\}\to M(s)\mid s\in\{x_c,y_c\}\to 0\mid s=u_v\to M(x_v)+M(y_v)\mid s=u_c\to 1]$$

$$t_*^I(M)(s)=[s\in\{x_v,y_v,z_v,u_v,u_v,x_c,y_c\}\to M(s)\mid s\in\{u_c,z_c\}\to 0\mid s=w_v\to M(u_v)*M(z_v)\mid s=w_c\to 1]$$

Fig. 4.1

The atomic decomposition of this net is shown in Fig.4.2

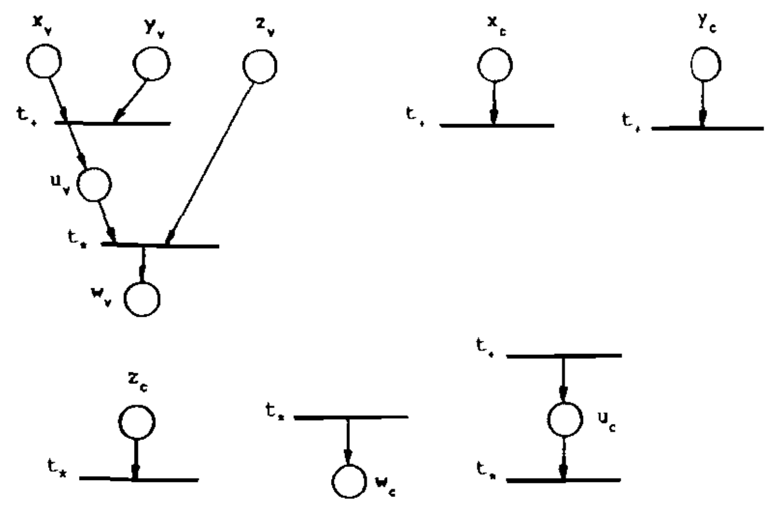


Fig. 4.2

5. An algebraic note

We conclude with a simple and rather loose observation. Every abstract net determines an abelian, partial semigroup of some of its subnets, where the semigroup operation (partial, because the operation $\|$ stipulates that the sets of places of its arguments be disjoint) is obtained as follows. If $\|_{j=1}^{n} P_j$ is the canonic (unique, by Theorem 4.4) representation of a given net P, then $\{P_1, \ldots, P_n\}$ is the set of all atomic subnets of P. Let SG[P] be the set of all nets of the form $P_{k_1} \| P_{k_2} \| \ldots \| P_{k_m}$ where $1 \leq k_i \leq n$,

$k_i \neq k_j$ for $i \neq j$ and $1 \leq i \leq m$, $1 \leq j \leq m$. The semigroup is then $\langle SG[P], \| \rangle$ and its set of generators is $\{P_1, \ldots, P_n\}$. For a net P with "Petri-like" interpretation, every such generator is a single-place net. Let us denote it $t_1, \ldots, t_{i-1} s t_i, \ldots, t_n$, where s stands for a place, $t_1, \ldots, t_{i-1}$ stand for entry to s transitions and $t_i, \ldots, t_n$ for exit from s transitions. The cho-sen notation suggests a language for writing nets $\|$-factorisable into single-place generators. Its alphabet consists of countable sets **S** and **T** of places and transitions respectively, concurrency symbol "$\|$" and comma ",". Its sentences are net-terms: a net-term is either $t_1, \ldots, t_{i-1} s t_i, \ldots, t_n$ or if Q and R are net-terms with disjoint sets of places then $Q \| R$ is also a net-term. Every net-term is a denotation of a net, but, clearly, this correspondence is many-to-one, since many net-terms may denote the same net. Considering the syntax only, it is easy to characterize them algebraically by providing a few equalities between terms and then stating that two terms are syntactically equivalent with respect to these equalities if and only if they represent the same net-structure. Syntactic equivalence, denoted by "$\leftrightarrow$", means that a term can be transformed into equivalent one by succesive application of given equalities. In language definition "," and "$\|$" are just syntactic formation symbols for terms. In algebraic considerations "," is an operation in the set $\mathbf{T}^*$ of all finite lists of transition names (with empty list λ) and "$\|$" is an operation in the set **NT** of all net-terms. The equalities are:

For any $u, v, w \in \mathbf{T}^*$, $P, Q, R \in \mathbf{NT}$

(a) $\lambda, u = u, \lambda = u$

(b) $u, (v, w) = (u, v), w$

(c) $u, v = v, u$

(d) $u, u = u$

(e) $P \| (Q \| R) = (P \| Q) \| R$

(f) $P \| Q = Q \| P$

Theorem 5.1

$$P \leftrightarrow Q \quad iff \quad \langle S_P, T_P, F_P \rangle = \langle S_Q, T_Q, F_Q \rangle$$

Proof outline

Let $P \leftrightarrow Q$. Note, that application of one equality of (a) – (l) to a term P does not change the net-structure $\langle S_P, T_P, F_P \rangle$. Indeed, in such one-step transformation the sets S_P and T_P remain unchanged – this follows from considering six cases of transformation, one case for one equality Also, F_P remains unchanged, since the one-step transformation leaves neighbourhoods of places unchanged By inductive argument we conclude that any finite number of single steps leave all the three sets unchanged, thus,

$$\langle S_P, T_P, F_P \rangle = \langle S_Q, T_Q, F_Q \rangle .$$

Conversely, let not $P \leftrightarrow Q$. Thus, there exists an atomic term in one net, which has no $\leftrightarrow$ equivalent counterpart in the other. This means that either

$$S_P \neq S_Q \quad or \quad T_P \neq T_Q \quad or \quad F_P \neq F_Q$$

or any combination thereof hold, therefore

$$\langle S_P, T_P, F_P \rangle \neq \langle S_Q, T_Q, F_Q \rangle$$

q.e.d.

The language and the algebra of its terms get a little more complicated if generators induced by a given interpretation are not necessarily single-place subnets.

References

[Hoa 81] C.A.R.Hoare, A Model for Communicating Sequential Processes, Oxford University Computing Laboratory, Technical Monograph PRG-22, June 1981

NETS AND THEIR RELATION TO CSP

Ludwik CZAJA

Oxford University Computing Laboratory
Programming Research Group
8-11 Keble Road, OXFORD OX1 3QD, U.K.

(on leave from Institute of Informatics
University of Warsaw, Poland)

June 1984

1. Introduction

Starting from a concept of "abstract" net - an *undirected* bipartite graph depicting a locality *relation* rather than flow relation, but interpreted quite generally (transitions represent arbitrary state transformations) - we define some CSP-like operations on nets. These are: parallel synchronised composition, external choice, asynchronous interleaving, prefix ("first fire transition t then behave like net P") and recursion. They are so defined that the set of firing sequences generated by a composite net equals the respective CSP combination of the sets of firing sequences generated by its components. Thus, the underlying model on which the relationship between nets and (a part of) CSP is investigated here is the trace model. The considerations, abstract and semantic at the beginning, become more specific and syntactic as the story proceeds. Firstly, introducing a "plug-in" constructor (Section 3) we come up

This work was supported by a Visiting Fellowship Research Grant from the UK Science and Engineering Research Council

with a syntax of general nets, but this will exhibit merely their structure, i.e. the locality relation. The behaviour of nets denoted by their syntactic forms is still almost entirely hidden in the interpretation. Next (Section 4), three CSP-like operators are defined for nets: parallel composition, choice and interleaving and this makes the first step towards syntactic facilities for interpretation. The next one is to define a counterpart of CSP's prefixing construct. We adopt the Petri net's flow of control mechanism for this, comming up with a syntax and semantics of *control* nets. These are essentially place/transition Petri nets with arrow-multiplicity 1 and are written as expressions which we call control terms. Their semantics will be defined in denotational manner (Section 5). Finally (Section 6), a class of nets, D-nets as we say, will be selected and their formal language defined. They are finite p/t Petri nets of restricted structure but augmented semantics and are targets which CSP-processes will be translated into. In dealing with recursion, we follow the technique of syntactic approximations from [Hoa-Olde 83], then single out a special case - looping - which allows for exceptionally simple translation. The translation function $\mathbb{F}$ may be thought of as an algorithm of assingning nets to CSP-procsses. Hence a kind of net-model of CSP. Unfortunately, there remain three CSP notions: divergence, internal choice and hiding, to make this model more complete. The paper would, however, grow unplausibly, so we leave them, as well as inference rules for D-terms, to a separate one. The proofs of propositions and theorems will be rather sketchy, a few more eleborate proofs are in Appendix

2. Basic concepts

Net: structure and interpretation

A net is a pair $P = \langle \sigma[P], \mu[P] \rangle$, its structure is $\sigma[P]$, its interpretation $\mu[P]$. The structure is a triple $\sigma[P] = \langle S,T,F \rangle$, where S is a set of variables (typical member: s), T is a set of operators (typical member: t) and $F \subseteq \{\{s,t\}: s \in S,\ t \in T\}$ is a bipartite relation called here a locality relation. Pictorially, F is a set of lines connecting variables with operators alternately. We use $\cdot t$ to stand for the set of variables attached to operator t, i.e. $\cdot t \cdot = \{s \in S: \{s,t\} \in F\}$, called a neighbourhood of t in the net P. The interpretation is a pair $\mu[P] = \langle A, \{t^P: t \in T\} \rangle$, where A is a set of values of variables; a function $M: S \rightarrow A$ is a valuation of variables and $\mathbf{M} = A^S$ is the set of all valuations. t^P is a binary relation in $\mathbf{M}$, associated with operator t: $t^P \subseteq \mathbf{M} \times \mathbf{M}$. The only requirement is that this relation be, in a sense, local: holding of $\langle M,M' \rangle \in t^P$ should be determined by a relationship between restrictions $M|\cdot t \cdot$ and $M'|\cdot t \cdot$ and by equality $M|S\text{-}\cdot t \cdot = M'|S\text{-}\cdot t \cdot$. This will be made formal in the next section. A valuation is also called a state of the net. Operationally, t^P may be seen as a nondeterministic transition from a state to another state. We write Mt^PM' for $\langle M,M' \rangle \in t^P$. Although nets differ here from Petri nets in structure (which here is a bipartite undirected graph) and in interpretation (which here is quite abstract), we adopt Petri's phraseology. Accordingly, we say "places" for variables, "transitions" for operators, "markings" for states and "firable" for executable. Places will then be drawn as circles and transitions as bars or boxes. Relation F however plays here a part of locality relation rather than flow relation, or casual dependency relation, which will later be derived from F.

Firability, stop, chaos, skip

Transition $t \in T$ is firable at a marking M iff Mt^PM' for a certain M'. Two extremes are: $t^P = \emptyset$ (t never firable) and $t^P = \mathbf{M} \times \mathbf{M}$ (t always firable). in logical notation respectively: t^P = FALSE and t^P = TRUE, so in the first case t is just a stop transition, in the second - a chaos transition. If $t^P = \{\langle M,M \rangle: M \in \mathbf{M}\}$, i.e if t^P is identity relation ID, then t is a skip transition.

Extension of interpretation to sequences of transitions

If $t \in T$ and $v \in T^*$ then $v^{P*} \subseteq \mathbb{M} \times \mathbb{M}$ is defined inductively:

$\lambda^{P*} = ID$, $(tv)^{P*} = t^P \circ v^{P*}$, where λ is the empty sequence, $\circ$ is composition of relations. If $M_0 v^{P*} M$ then v is a firing sequence leading from marking M_0 to M. In what follows, we drop the star, writing v^P for v^{P*}. Note that $(uv)^P = u^P \circ v^P$ and that the stop transition cannot occur in a firing sequence.

Language of firing sequences

If $M_0 \in \mathbb{M}$ then $L(M_0,P) = \{v \in T^*: \exists M: M_0 v^P M\}$ is the set of all firing sequences generated by the net P from M_0.

Example 2.1: $a^n b^n c^n$ - language

For the net P drawn in Fig.2.1,

$L(M_0,P) = \bigcup_{n \geq 0} t_1^n t_2^n t_3^n$

if $\mu[P] = \langle \mathbb{N}, \{t_i^P: i=1,2,3\} \rangle$ with

$M t_i^P M' \iff M(s_{i-1})=0 \wedge M(s_i)>0 \wedge \forall j=0..4: M'(s_j)=M(s_j)+k_{ij}$

where $k_{ij}=-1$ for $j=i$, $k_{ij}=1$ for $j=i+1$ and $k_{ij}=0$ otherwise

and with M_0 defined by: $M_0(s_1)=n$ and $M_0(s)=0$ for $s \neq s_1$.

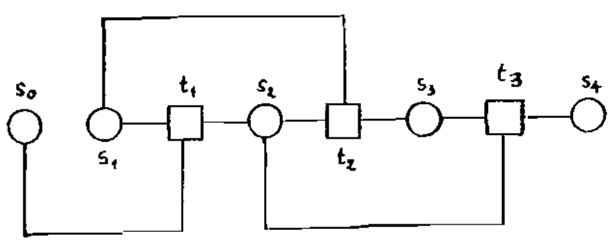


Fig. 2.1

Communicating processes "netted"

A net may be seen as a specification of a problem. Pictorially, such a specification exhibits a locality schema only: an operator can reach for an information stored in a place s, say, but not in s'. What the operator does, what can or must be done next or simultanously, is hidden in the interpretation, thus not readable from the picture. The level of such specification may vary from a rough schema of a (distributed) system to a program in machine code or a network of gates in a circuit. Take, for example, a flowchart P, Fig.2.2(a), of a sequential program with uniquely labelled boxes containing CSP's i/o instructions Q?, Q!,..(variables and expressions are dropped). P may be drawn as a net in Fig.2.2(c), where thansitions are labels, places are points of control flow in the flowchart and may hold a token indicating presence of the control. Conditional instructions (diamonds) are represented in the net as conflict places. Thus, at this level of specifying, the conflict is resolved nondeterministically. The net assumes Petri's interpretation, so we direct its lines (a syntactic provision). In Fig.2.2(b) there is another flowchart, Q and its net-representation is in Fig.2.2(d). Now, we wish to make a net for parallel composition P||Q of communicating processes. Subscripts will indicate a net to which subscripted things belong and "t_P matches t_Q" means:

either t_P labels a Q? and t_Q labels a P!

or t_P labels a Q! and t_Q labels a P?

The net P||Q is defined as follows:

$S_{P||Q} = S_P \cup S_Q$

$T_{P||Q} = \{\{t_P, t_Q\}: t_P \text{ matches } t_Q\}$

$F_{P||Q} = F^+ \cup F^-$ where

$F^+ = \{\langle s, \{t_P, t_Q\}\rangle : \langle s, t_P\rangle \in F_P \vee \langle s, t_Q\rangle \in F_Q\}$

$F^- = \{\langle \{t_P, t_Q\}, s\rangle : \langle t_P, s\rangle \in F_P \vee \langle t_Q, s\rangle \in F_Q\}$

So, a transition $t = \{t_P, t_Q\}$ in P||Q identifies a pair of instructions capable of communicating mutually and t is firable (read: the communication may occur) iff both t_P and t_Q are (read: control reached t_P and t_Q). The net P||Q is in Fig.2.2(e). Summerising, we define interpretation in P||Q :

$M t^{P||Q} M' \iff (M|S_P) t^P_P (M'|S_P) \wedge (M|S_Q) t^Q_Q (M'|S_Q)$

where $t = \{t_P, t_Q\}$, M, M' are markings in P||Q and $M|S_P$ is a restriction of M to S_P etc. Places in P and Q should be distinct: $S_P \cap S_Q = \emptyset$.

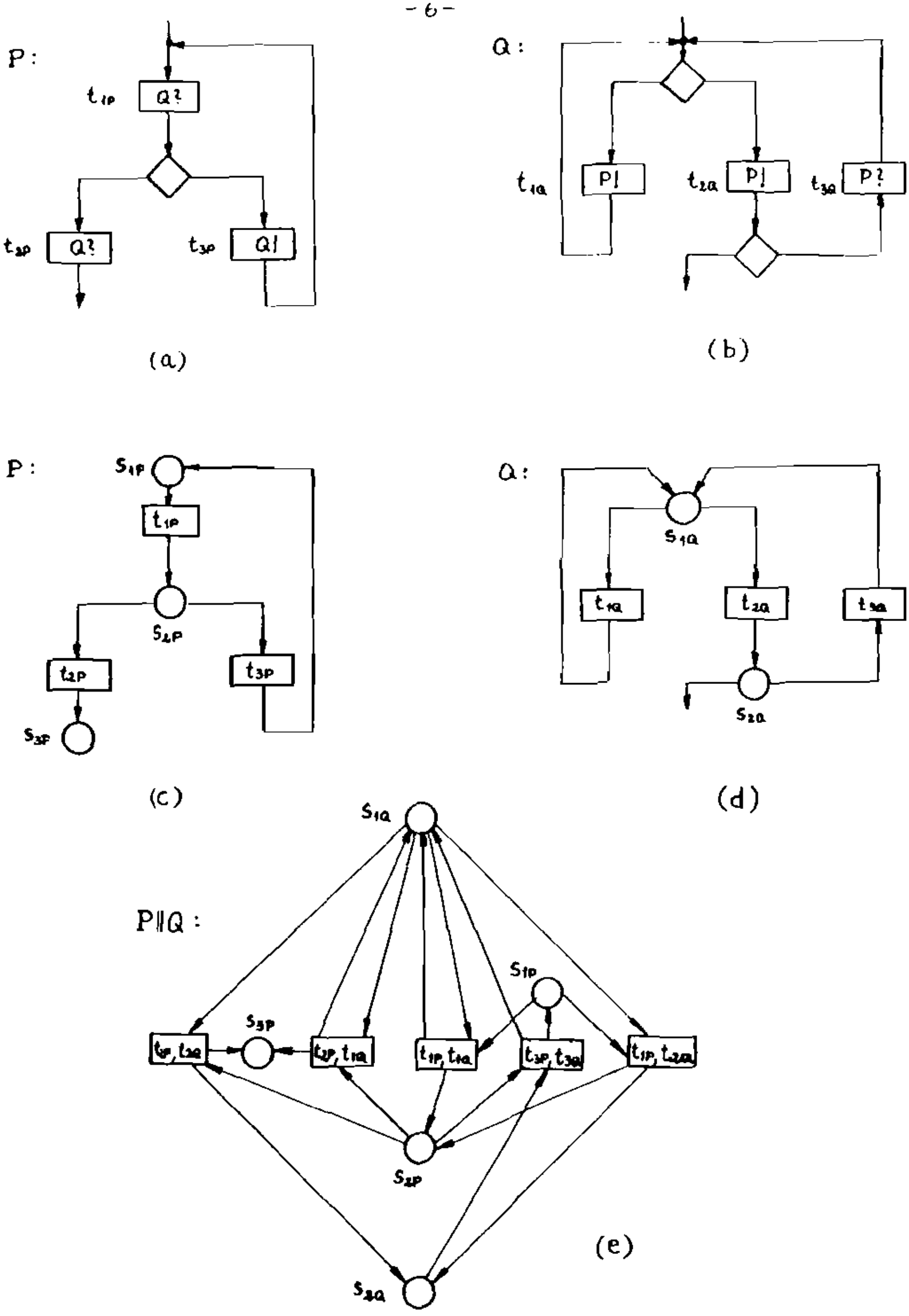

P :
t1P
Q?
t2P
Q?
t3P
Q!
(a)
Q :
t1Q
P!
t2Q
P!
t3Q
P?
(b)
P :
s1P
t1P
s2P
t2P
t3P
s3P
(c)
Q :
s1Q
t1Q
t2Q
t3Q
s2Q
(d)
P‖Q :
s1Q
s1P
s3P
t2P, t2Q
t2P, t1Q
t1P, t1Q
t3P, t3Q
t1P, t2Q
s2P
s2Q
(e)

Fig. 2.2

In Section 4 the story is a little simplified: $T_{P||Q} = T_P \cup T_Q$ and there may be common transitions in P and Q. Synchrinisation will then occur on those common transitions. Further results, however, can be strightforwardly quoted, starting from the above general definition of P||Q. The experiment with "||" encourages to try also some other constructors: we construct some classes of nets from one or a few atoms.

Labelling

Transitions in a net P are then labels of instructions from a set I in the "netted" system, hence a labelling function: $h: T \rightarrow I$ which may be extended to sequences by:

$h(\lambda) = \lambda$ (λ - empty sequence of transitions or instructions)

$h(vt) = h(v)h(t)$ ($v \in T^*$ $t \in T$)

and to sets of sequences $X,Y \subseteq T^*$:

$h(XY) = h(X)h(Y)$ (the symbol h is ueed also for extensions).

Thus, $traces(P') = h(L(M_0,P))$, where P' is a system represented as a net P with initial marking M_0 and traces(P') is the set of traces generated by P', i.e. instruction sequences recorded in the order of their execution. Following the above mentioned simplification we will assume: $traces(P') = L(M_0,P)$.

Are nets predicates?

We conclude this section with the following remark, which, at least in its first part touches an issue, for programs expressed in [Hoa-Olde 83], [Hoa 84]. In operational terms, net's activity may be obeerved either as a progress of subsequent firings, or as a progressive changee of the state. In the first case, our "observation space" is T^* (with the usual tree-like ordering), in the second it is $\mathbb{M}$ (with an ordering derived from locality P). Fixing a marking $M_0 \in \mathbb{M}$ as initial, we might abstractly identify a net P with a predicate P(v): $\exists M \in \mathbb{M}: M_0 v^P M$ - if we are concernad with observing firing sequences (traces) as P's behaviours, or we might identify it with a predicate P(M): $\exists v \in T^*: M_0 v^P M$ - if our concern is to observe changes of the state. In both cases, however, when making this identification, we consent on losing a relativistic aspect of net's behaviour: two observers may see it quite differently, hence quite different are predicates accounting for their observations. In other words, we assume the "absolute time scale", or a "global clock" ensuring total ordering of events. This involves a simulation of concurrency by interleaving rather than a direct description. To capture timing aspects (e.g. concurrency) more adequately, one might incorporate some concepts from the fundamental work [Petri 76] and follow the technique developed e.g. in [Maz 77], [Lauer 75], [Win 77]. That is, instead of "total sequences", to assume "partial sequences" as observable objects. These objects are equivalence classes of a predefined independence relation between events. However, the free variable of our net-predicates would then range over constructe much more elaborate than firing sequences or markings - an increase of adequacy at the price of complexity of (hardly) observable objects.

3. Constructive description

A net-term is a notation for a net. To simplify writing, we use the same symbols P, Q,... for (names of) net-terms and nets for which they stand. Two construction devices are chosen in this Section: tupling of places and a "plug-in" $\leftrightarrow$ constructor, which attaches a given transition to a net already built up.

Example 3.1

$t_3(s_2,s_3,s_4)\leftrightarrow(s_0,s_1,s_2,s_3,s_4)$ denotes a net consisting of places

$s_0,\ldots,s_4$ and transition t_3 "plugged into" places s_2,s_3,s_4 and

$t_1(s_0,s_1,s_2)\leftrightarrow t_2(s_1,s_2,s_3)\leftrightarrow t_3(s_2,s_3,s_4)\leftrightarrow(s_0,s_1,s_2,s_3,s_4)$

denotes the $a^n b^n c^n$ net in Fig 2.1.

Definition 3.1 (net-terms)

1. $(s_1,\ldots,s_n)$ is a net-term denoting a net P with:

$\sigma[P] = \langle\{s_1,\ldots,s_n\},\emptyset,\emptyset\rangle$

$\mu[P] = \langle A,\emptyset\rangle$, where A is a set.

2. Let be given: a net $Q = \langle\langle S,T,F\rangle,\langle A,\{t^Q:\ t\in T\}\rangle\rangle$,

a transition $t_0 \notin T$, places $\{s_1,\ldots,s_n\} \subseteq S$ and

a relation $\rho \subseteq A^n \times A^n$ (put ρ=TRUE for n=0). Then:

$t_0(s_1,\ldots,s_n)\leftrightarrow Q$ is a net-term denoting a net P with:

$\sigma[P] = \langle S,T\cup\{t_0\},F\cup\{(s_i,t_0):\ i=1,\ldots,n\}\rangle$

$\mu[P] = \langle A,\{t^P:\ t\in T\cup\{t_0\}\}\rangle$, where $t^P = t^Q$ for $t \neq t_0$ and

$Mt_0^P M' \iff (M(s_1),\ldots,M(s_n))\rho(M'(s_1),\ldots,M'(s_n)) \wedge M|S_0 = M'|S_0$

where $S_0 = S-\{s_1,\ldots,s_n\}$.

Obviously, for every net there is a net-term denoting it. Definition 3.1 gives then a construction mechanism for nets. Almost as evident is the following

Proposition 3.1

(1) Suppose for $i,j \in \{1,2\}$:

(a) $M_{i1}|{}^\cdot t^\cdot = M_{j1}|{}^\cdot t^\cdot$

(b) $M_{i1}|S-{}^\cdot t^\cdot = M_{i2}|S-{}^\cdot t^\cdot$

Then: $M_{11}t^P M_{12} \iff M_{21}t^P M_{22}$

(2) If ${}^\cdot t^\cdot = \emptyset$ then t = skip

(3) If t = chaos then ${}^\cdot t^\cdot = S$

Proposition 3.1 may be verbalised as follows. (1) - two markings are in relation t^P only and when two other markings, coinciding respectively with the former on t's neighbourhood, are.
(2) - isolated transitions are firable but do nothing.
(3) - chaos is attached to all places.

<u>Nondeterminism and functions</u>

Regarding nets operationally, we find two sorts of nondeterminism in their activity. One is at the level of the whole net, since it may be more than one firable transition at a given marking and the other is at the level of single transitions because t^P is a relation, not necessarily being a function. Notice that in Petri nets it is a partial function $t^P: \mathbf{M} \nrightarrow \mathbf{M}$, hence - the first sort of nondeterminism only. However, assuming t^P to be a relation is beneficial when we introduce in the next section some CSP-like constructors for nets. It turns out that the concurrency constructor $\|$ preserves the functionality (i.e. $t^{P\|Q}$ is a function provided that t^P and t^Q are), while the interleaving $|||$ and the choice $\Box$ do not. This property simplifies its definition. If relation ρ in Definition 3.1 is a partial function $\rho: A^n \nrightarrow A^n$ then t^P is a partial function $t^P: \mathbf{M} \nrightarrow \mathbf{M}$.
We write then $M' = t^P(M)$ whenever Mt^PM'. If t^P is undefined at $M \in \mathbf{M}$, i.e. when t is not firable at M, it is written $t^P(M) = \bot$ and we assume $\bot \notin \mathbf{M}$. Two conventions are adopted: $M \cup \bot = \bot \cup M = \bot$ and $\bot|S_0 = \bot$, for any $M \in \mathbf{M}$ and $S_0 \subseteq S$.

Proposition 3.2

Let t^P be a partial function $t^P: \mathbf{M} \nrightarrow \mathbf{M}$. Then:

(1) If $M|{}^{\cdot}t^{\cdot} = M'|{}^{\cdot}t^{\cdot}$ then $t^P(M)|{}^{\cdot}t^{\cdot} = t^P(M')|{}^{\cdot}t^{\cdot}$

(2) If $t^P(M) \neq \bot$ then $t^P(M)|S-{}^{\cdot}t^{\cdot} = M|S-{}^{\cdot}t^{\cdot}$

Proof - direct from Definition 3.1.

Operationally, Proposition 3.2 verbalises: (1) - the effect of firing a transition <u>depends</u> solely on its neighbourhood,
(2) - the effect of firing a transition <u>confines</u> to its neighbourhood. This expresses the local character of transition's activity.

4. CSP-like operations on nets

Definition 3.1 allows to construct arbitrary net but, provides no syntactic means for specifying its behaviour, except that each transition acts locally. The behaviour is determined by interpretation of transitions. To illustrate this, recall Example 3.1, where the net-term determines the structure of $a^n b^n c^n$ net, but its behaviour, e.g. sequencing, is enforced externally by a rule separate for each transition. In this section we make a first step towards some syntactic facilities for interpretation, adopting three CSP-like constructors for nets: parallel composition $\|$, asynchronous interleaving $|||$ and general choice $\Box$. Their definitions are so chosen that behaviour of composite nets $P\|Q$, $P|||Q$ and $P\Box Q$ correspond to that of respective composite CSP processes, if this correspondence holds for the components P and Q. However, the correspondence may be established merely if a common observation space is taken for nets and CSP and as the one we take here the set of traces. Therefore we define $\|$, $|||$ and $\Box$ constructors in such a way that if $\langle M^P,P\rangle$, $\langle M^Q,Q\rangle$ are nets with fixed initial markings and P', Q' are CSP processes such that

$L(M^P,P) = traces(P')$ and $L(M^Q,Q) = traces(Q')$ then:

$L(M^{P\|Q},P\|Q) = traces(P'\|Q')$, $L(M^{P|||Q},P|||Q) = traces(P'|||Q')$,

$L(M^{P\Box Q},P\Box Q) = traces(P'\Box Q')$, where $M^{P\|Q}$ etc. is the

initial marking in $P\|Q$ and operators $\|$, $|||$, $\Box$ on CSP processes P' and Q' are from [Hoa 83]. The next step in providing syntactic facilities for interpretation is to derive sequencing, or flow of control, from locality relation F. Thus, to model the CSP's concept of prefixing. This is in the next section.

Superscript notation

For a net $P = \langle\langle S,T,F\rangle,\langle A,\{t^P\colon t\in T\}\rangle\rangle$ and marking M we use M^P to stress that this is a marking in P. Tu put it differently, $M^P = M|S$ is a restriction of M to S. This is a profitable notation when several nets with disjoint sets of places are combined into one. For example, as we will see in a while, $M^{P\|Q} = M^P \cup M^Q$. Note that the notation "t^P" also stresses that transition t is interpreted in P.

Definition 4.1

Let nets P and Q be given with:

$\sigma[P] = \langle S_P, T_P, F_P \rangle$, $\mu[P] = \langle A_P, \{t^P : t \in T_P\} \rangle$

$\sigma[Q] = \langle S_Q, T_Q, F_Q \rangle$, $\mu[Q] = \langle A_Q, \{t^Q : t \in T_Q\} \rangle$

and let $S_P \cap S_Q = \emptyset$. Nets $P \| Q$, $P \||| Q$ and $P [] Q$ are defined as follows. Their structure is identical:

$\sigma[P \| Q] = \sigma[P \||| Q] = \sigma[P [] Q] = \langle S_P \cup S_Q, T_P \cup T_Q, F_P \cup F_Q \rangle$

Their interpretations:

the set A of values of places is $A_P \cup A_Q$ and transitions in composite nets are interpreted as follows:

$M t^{P\|Q} M' \iff M^P t^P M'^P \wedge M^Q t^Q M'^Q$

$M t^{P\|||Q} M' \iff (M^P t^P M'^P \wedge M^Q = M'^Q) \vee (M^Q t^Q M'^Q \wedge M^P = M'^P)$

$M v^{P[]Q} M' \iff ((M^P v^P M'^P \wedge M^Q = M'^Q) \vee (M^Q v^Q M'^Q \wedge M^P = M'^P)) \wedge v \in T_P^* \cup T_Q^*$

where $M = M^P \cup M^Q$, $M' = M'^P \cup M'^Q$ are markings in the composite nets.

<u>Comments</u>

(1) $M = M^P \cup M^Q$, the union of functions, is meant as the union of relations. Due to $S_P \cap S_Q = \emptyset$, M again is a function - this motivates the assumption.

(2) The choice of the meaning of $\|$, $|||$, $[]$ for nets is formally justified by Theorem 4.1. Informally, in operational terms, it may be understood as follows. For $t^{P\|Q}$ notice that if $t \notin T_Q$ then $Mt^{P\|Q}M' \iff M^P t^P M'^P \wedge M^Q = M'^Q$ (by Proposition 3.1(2)). Thus, the activity of t in P$\|$Q is just its activity in P and has no effect for Q. Similarly is when $t \notin T_P$. If $t \in T_P \cap T_Q$ the activity of t in P$\|$Q is its simultanous activity in P and Q. A similar argument convinces that P and Q work entirely independently in their composition P$|||$Q. The motivation of $\|$ for nets is to provide a syntactic support for synchronisation (by "handshaking"). On the other hand, the simplicity of its definition is due to simplifying convention made in Section 2 and to a set-theoretic principle: if t occurs in T_P and in T_Q then it occurs in $T_P \cup T_Q$ as a <u>one</u> member, being an identification of these two occurences. The motivation of $|||$ is to express independent work of several copies of one net without renaming transitions in the different copies. Again, the definition makes use of the mentioned principle. For P$[]$Q, the argument is a little more subtle and the reader is advised to recall definition of extension of interpretation to sequences (Section 2). Firstly, notice that the interpretation of single transitions cannot be primitive, but must be derived from a stronger information, i.e. from the interpretation of firing sequences (traces). This must be so, if we wish to retain the abstract interpretation in the components P and Q, rather than to specify it somehow for the purpose of a definition of P$[]$Q. For example, in Petri nets, the choice is realised by the concept of conflict, but this is due to the very specific interpretation. In contrast, the above definition of P$[]$Q allows for determining $v^{P[]Q}$ ($v \in T^*$) without any specific assumption on interpretations v^P, v^Q. Realising P$[]$Q by conflict may be seen as the implementation of $[]$ in terms of Petri nets. Secondly, every behaviour, i.e. $v \in T^*$, of P$[]$Q should comprise firings in exactly one component, either P or Q. Hence, if v is a firing sequence either in P or in Q then $v \in T_P^* \cup T_Q^*$. Thirdly, if $v = t_0 t_1 \ldots t_{n-1}$ is made by P$[]$Q then looking at t_0 one knows whether v is made by P or by Q provided that $t_0 \notin T_P \cap T_Q$. Otherwise, to find out this, one has to look at t_1, then, perhaps at t_2 etc. This suggests to call $[]$ a "general choice" or "external choice", after [Hoa 83].

Proposition 4.1

(1) Operations $\|$, $|||$, $[]$ are associative and symmetric.

(2) If $T_P \cap T_Q = \emptyset$ then $P|||Q = P\|Q$

(3) If t^P and t^Q are functions then so is $t^{P\|Q}$, but $t^{P|||Q}$ and $t^{P[]Q}$ need not be.

(4) If $t \notin T_P$ then t^P is (total) function $t^P(M) = M$

(5) If t^P and t^Q are functions then $t^{P[]Q}(M) = t^P(M^P) \cup t^Q(M^Q)$

Proof - direct from definitions.

The following result justifies definition of operations $\|$, $|||$, $[]$ for nets:

Theorem 4.1

Let M_0 be a marking (initial) in either of composite nets $P\|Q$, $P|||Q$, $P[]Q$. Then:

(1) $L(M_0, P\|Q) = L(M_0^P, P)\|L(M_0^Q, Q)$

(2) $L(M_0, P|||Q) = L(M_0^P, P)|||L(M_0^Q, Q)$

(3) $L(M_0, P[]Q) = L(M_0^P, P) \cup L(M_0^Q, Q)$

where operations $\|$ and $|||$ on languages, i.e. sets of traces, are defined in [Hoa 83].

Proof - Appendix.

5. Control

Sequencing

So far we could express flow of control specifying it in interpretation. To provide a suitable syntax, notice that unlike other CSP-like operators, the prefixing "$t \rightarrow P$" makes no sense for nets unless one indicates a marking at which "first fire t then behave like P". So, we should rather write $t \rightarrow (M_0, P)$, but this also is unfortunate notation as it involves marking, which is not a syntactic object. A way is to adopt the Petri-like flow of control mechanism, thus to select a restricted class of nets as "control patterns". Such a control net C, when combined with a net P by concurrency operator, enforces a sequencing discipline in $C \| P$. (Notice that the dataflow concept is expressible this way). Definition 5.1 is motivated by the fact that every place/transition Petri net (assume that multiplicities on arrows are 1) may be built from one place, one transition Petri-like interpreted, applying $\leftrightarrow$ and $\|$ constructors. We consider only pure nets, i.e. those without tight loops $st \leftrightarrow ts$. By Proposition 5.1, the definition provides denotational semantics for a language of Petri nets.

Definition 5.1 (Control terms and nets)

Syntax

$C ::= f \mid f \| C$

$f ::= (s) \mid st \leftrightarrow f \mid ts \leftrightarrow f$

The syntactic categories C, f, s, t are read respectively: control-term, factor, place, transition. Context-dependent restrictions for $st \leftrightarrow f$ and $ts \leftrightarrow f$ are: s must occur in f but t must not (hence, factor is a single-place construct with at most one occurence of every transition). And for $f \| C$: the place in f must not occur in C. A factor in which s occurs will be called s-factor.

Semantics

Control-terms describe directed nets called control nets. Take natural numbers as values of places. The structure and interpretation of control nets is defined as follows:

Case $f ::= (s)$: $S_f = \{s\}$, $T_f = \emptyset$, $F_f = \emptyset$, $Mt^f M' \iff M = M'$

Case $f ::= st_0 \leftrightarrow f_0$: $S_f = S_{f_0} \cup \{s\}$, $T_f = T_{f_0} \cup \{t_0\}$,

$$F_f = F_{f_0} \cup \{\langle s, t_0 \rangle\}$$

$$Mt^f M' \iff \begin{cases} M(s) = M'(s)+1, & \text{if } t = t_0 \\ Mt^{f_0} M', & \text{if } t \neq t_0 \end{cases}$$

Case $f ::= t_0 s \leftrightarrow f_0$: $S_f = S_{f_0} \cup \{s\}$, $T_f = T_{f_0} \cup \{t_0\}$,

$$F_f = F_{f_0} \cup \{(t_0, s)\}$$

$$M t^f M' \iff \begin{cases} M'(s) = M(s)+1, & \text{if } t = t_0 \\ M t^{f_0} M', & \text{if } t \neq t_0 \end{cases}$$

Case $C ::= f \,\|\, C_0$: $S_C = S_f \cup S_{C_0}$, $T_C = T_f \cup T_{C_0}$,

$$F_C = F_f \cup F_{C_0}$$

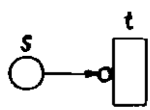
$$M t^C M' \iff M^f t^f M'^f \wedge M^{C_0} t^{C_0} M'^{C_0}$$

where $M^f = M|S_f$ etc.

Abbreviation: omit "$\leftrightarrow\{s\}$" in "$st\leftrightarrow \ldots \leftrightarrow\{s\}$" etc.

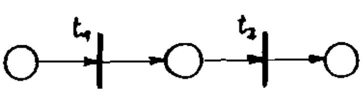
<u>Remarks</u>

(1) Racall Definitions 3.1 and 4.1: a factor is thus a net-term with $n = 1$, $A = \mathbb{N}$, <u>directed</u> locality relation F, ρ given by either $x\rho y \iff x = y + 1$ or $x\rho y \iff y = x + 1$ and with condition $M|S_0 = M'|S_0$ ensured by the case $f ::= \{s\}$. A control net is a parallel combination of factors. The context-dependent restrictions in Definition 5.1 conform to requirements in Definitions 3.1 and 4.1.

(2) Notice that inhibiting arrows: 

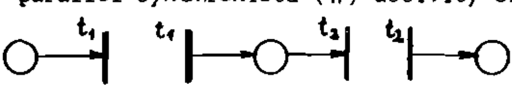
(t cannot fire if $M(s)>0$) from [Age-Fly 7] is just another sort of a control net, which is like st except that $x\rho y \iff x=y=0$.

(3) Notice that sequencing, or control flow through a net, e.g.



is actually <u>simulated</u> by

parallel synchronised (||) activity of three nets:



and this is essentially what is going on in Petri nets.

Proposition 5.1 (A characterisation of Petri nets)

Every control net is a finite pure Petri net and conversely.

Proof

(s) is a no-transitions-one-place Petri net, constructors ↔ and || preserve property of being Petri net. Conversely, every Petri net is (uniquely) ||-factorisable into one-place nets (Theorem 4.3 and Example 4.2 in [Cza 84]). Every pure one-place Petri net is described by a factor f.

q.e.d

Example 5.1

The net in Fig. 5.1 is described by the control-term:

$st_1 \leftrightarrow st_2 \leftrightarrow t_3 s \leftrightarrow t_4 s \parallel t_1 s_1 \leftrightarrow s_1 t_3 \parallel t_2 s_2 \leftrightarrow s_2 t_4$

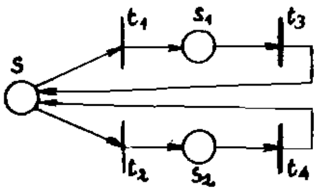


Fig. 5.1 Behaviour of one fork on Philosophers' table

Deadlock and recurrence

A deadlock occurs if a marking is reached when no transition can fire:

$\exists M \in [M_0\rangle \forall t \in T_P$: t is not firable at M

where $[M_0\rangle = \{M : \exists v \in T_P^* : M_0 v^P M\}$

is the set of all markinge reachable from M_0. This is a total deadlock. Othere, when only some transitions will never fire are defined obviously.
A recurrence occurs if initial marking is restored:

$\exists v \in T_P^* : M_0 v^P M_0 \wedge v \neq \lambda$

Deadlock and recurrence can occur in arbitrary nets, not necessarily control nets. However:

Proposition 5.2

It is decidable whether a control net with initial marking M_0 can reach a deadlock or rsstors M_0.

Proof : Reachability problem is decidable for such nets [Kos 82].

6. Some CSP-processes and nets: a relationship

Prefix, postfix

These are transitions beginning and ending net's activity. To be formal, firstly say that a given place s is a source (sink) if $\{t: F(t,s)\} = \emptyset$ $(\{t: F(s,t)\} = \emptyset)$ and for transitions similarly. Syntactically, s is a source (sink) place in a given control term if no ts (st) occur in its s-factor. Secondly, for a term C and transition t denote by t.C (C.t) a term obtained from C by replacing every s-factor f_s, where s is a source (sink) place, by s-factor $ts \leftrightarrow f_s$ $(st \leftrightarrow f_s)$.
Restriction: t must not occur in f_s. Example:

$$t.(s_1t_1 \| s_2t_1 \| t_1s_3) = ts_1 \leftrightarrow s_1t_1 \| ts_2 \leftrightarrow s_2t_1 \| t_1s_3$$

Notice that:

(1) Operation "." means creating arrows from transition t to source places in a net C

(2) If there are no source places in C then $^{\cdot}t^{\cdot} = \emptyset$ in the net t.C

(3) There are no source places in the net t.C

D-terms and D-nets

Definition 6.1

Syntax

$$D ::= (s) \mid st \mid t.D \mid D \square D' \mid D \| D' \mid D ||| D'$$

where s is a place, t is a transition, D and D' are D-terms.

Semantics

D-terms describe D-nets. Their semantics is determined by the meaning of prefixing (.) and by Definitions 5.1, 4.1.

Comment: D-nets are finite Petri nets with restricted structure and augmented semantics: there is at most one entry and exit arrow from each place, there are no cycles and $t^{D \square D'}$, $t^{D \| D'}$ are not Petri interpretations, even if t^D, $t^{D'}$ are (see Proposition 4.1(3)).

The following theorem states that the D-net $s_0t \| t.D'$, without source transitions, marked 1 on s_0 and 0 elsewhere, behaves like CSP's prefix construct: "first t then behave like D".

Theorem 6.1

Let: D' be a D-net without source transitions, $D = s_0 t \| t.D'$,

and $M_D(s_0) = 1$, $M_D(s) = 0$ for $s \neq s_0$. Then:

$L(M_D,D) = \{\lambda\} \cup \{t\} \bigcup_{M \in \mathbf{M}_{D'}} L(M,D')$

where $\mathbf{M}_{D'} = \{M : M$ is a marking in D' and $M_D t^D (M \cup \{\langle s_0, 0\rangle\})\}$

and $M(s) = 1$ if s is a source place in D' and $M(s) = 0$

if not, for every $M \in \mathbf{M}_{D'}$

Proof Appendix

Restricted CSP

Considering a relationship between CSP and nets, we take a subset of an "abstract" CSP [Hoa-Olde 83] (but with |||):

$P ::= stop \mid t \rightarrow P \mid P \square Q \mid P \| Q \mid P ||| Q \mid \xi \mid \mu\xi.P$

Here, P‖Q is a parallel composition with synchronised communications in the intersection of P's and Q's alphabets. Other CSP constructs, like divergence, local nondeterminism and hiding are left to a separate paper. We assume that informal meaning of CSP processes is known to the reader and define a:

Translation function $\mathcal{F}$: CSP-processes → nets

Firstly, we translate finite CSP-expressions: stop, $t \rightarrow P$, $P \square Q$, $P \| Q$, $P ||| Q$. Event-letters are translated into names of net's transitions, but recall labelling and simplifying convention from Section 2. During the translation, new places are created. The translation procedure is:

$\mathcal{F}(stop) = (s)$ (creation of a new place)

$\mathcal{F}(t \rightarrow P) = st \| t.\mathcal{F}(P)$ (creation of a new place)

$\mathcal{F}(P \square Q) = \mathcal{F}(P) \square \mathcal{F}(Q)$

$\mathcal{F}(P \| Q) = \mathcal{F}(P) \| \mathcal{F}(Q)$

$\mathcal{F}(P ||| Q) = \mathcal{F}(P) ||| \mathcal{F}(Q)$

Secondly, turning to infinite CSP-expressions notice that "statically", $\mu\xi.P$ represents not just an infinite expression but even uncountable one - if P contains more than one free occurence of ξ (e.g. with two free ξ's, $\mu\xi.P$ is isomorphic to complete infinite binary tree, with as many branches as there are real numbers). Thus, we do not translate it directly into an uncountable net (we do not have such among D-nets!) in the full generality. Instead, we follow a technique of "syntactic approximations" from [Hoa-Olde 83]. And in the next paragraph we make a direct translation, which, although inadequate in some exceptionally malicious cases, seems to be sufficient in most "normal" ones.

Let Q_{stop} be a CSP expression which results from Q by replacing every occurrence of a $\mu\xi.R$ in Q by stop and let $P \vdash Q$ means that Q results from P by replacing one occurence
of a $\mu\xi.R$ by $R(\mu\xi.R)$. Here, $R(\mu\xi.R)$ is an expression obtained from R by replacing all free occurences of ξ by $\mu\xi.R$. If $P \vdash^{*} Q$ (transitive closure of $\vdash$) then Q_{stop} is called a <u>syntactic approximation</u> of P.
To translate $\mu\xi.P$ in general, we need infinite nets. One way is to make use of cpo's of occurence nets [Wins84], [Gol-Myc 84]. Instead, we extend (commutative and associative) operators $\|$, $\square$ to any collection of arguments:

$$E = \|_{z\in Z} D_z, \qquad E' = \square_{z\in Z} D_z$$

where $\{D_z : z \in Z\}$ is an indexed family of D-nets.

E is defined obviously:

$$S_E = \bigcup_{z\in Z} S_{D_z}$$

$$T_D = \bigcup_{z\in Z} T_{D_z}$$

$$P_D = \bigcup_{z\in Z} P_{D_z}$$

$$Mt^E M' = \bigvee_{z\in Z} M^{D_z} t^{D_z} M'^{D_z}$$

and similarly E' (i.e. by suitable extending Definition 4.1 to many arguments). Denote:

$$Z = \{Q_{stop} : (\mu\xi.P) \vdash^{*} Q\}$$

The translation is now simple:

$$\mathcal{F}(\mu\xi.P) = \square_{z\in Z} \mathcal{F}(z)$$

Theorem 6.2

If P is a CSP expression then traces(P) = $L(M_0, \mathcal{F}(P))$

where M_0 is a marking in the net $\mathcal{F}(P)$, such that $M_0(s) = 1$

if s is a source place and $M_0(s) = 0$ if not.

Proof

Notice that the Theorem holds for P = stop, then apply Theorems 6.1 and 4.1.

q.e.d

A special case of recursion: looping

In some cases (perhaps many) it suffices to translate $\mu\xi.P$ into a D-net (thus finite) as follows. Suppose P may be translated if we define additionally a translation of ξ. Suppose that free occurences of ξ in P are guarded: $t \rightarrow \xi$ and denote by

$g_1(\xi),\ldots,g_r(\xi)$ guards of all ξ's free occurences in P.

Thus, ξ occurs in P in contexts $g_1(\xi)\rightarrow\xi,\ldots,g_r(\xi)\rightarrow\xi$.

As a translation of $\mu\xi.P$ we admit the D-net $\mathcal{F}(P)$ with all these guards connected to source places of $\mathcal{F}(P)$. To do this formally, we write [t].D to mean the same as t.D but:

(1) it is legal to write [t].D even if t occurs in an s-factor of D, where s is a source place,

(2) source places in D remain those in [t].D.

Now, we have the translation:

$\mathcal{F}(\xi) = (s)$ (creation of a new place)

$\mathcal{F}(\mu\xi.P) = [g_1(\xi)]\ldots[g_r(\xi)].\mathcal{F}(P)$

The depth of recursion is in this case modelled by natural numbers - the values of places ("tokens" stored in places).

Acknowledgment

I am grateful to Jifeng He for many discussions on CSP. They helped to clarify some issues.

Appendix

Proof of Theorem 4.1(1) : $L(M_0, P \| Q) = L(M_0^P, P) \| L(M_0^Q, Q)$

Let $v \in L(M_0, P \| Q)$. This means there is a marking M and a firing sequence $v \in (T_P \cup T_Q)^*$ such that $M_0 v^{P\|Q} M$.

Denote by $v_1 = v \restriction T_P$ and $v_2 = v \restriction T_Q$ restrictions of v to T_P and T_Q respectively and observe, by Definition 4.1, that

$M_0 v^{P\|Q} M \iff M_0^P\ v_1^P\ M^P \wedge M_0^Q\ v_2^Q\ M^Q$

Thus, $v_1 \in L(M_0^P, P) \wedge v_2 \in L(M_0^Q, Q) \wedge v \in (T_P \cup T_Q)^*$

which, by definition of $\|$ for CSP [Hoa 81] means

$v \in L(M_0^P, P) \| L(M_0^Q, Q)$.

Let $v \in L(M_0^P, P) \| L(M_0^Q, Q)$ where M_0^P, M_0^Q are markings in P and Q respectively and let $v_1 = v \restriction T_P$, $v_2 = v \restriction T_Q$.

Then, by definition of $\|$ for CSP [Hoa 81] :

$v_1 \in L(M_0^P, P) \wedge v_2 \in L(M_0^Q, Q) \wedge v \in (T_P \cup T_Q)^*$

which means there is a marking M such that

$M_0^P\ v_1^P\ M^P \wedge M_0^Q\ v_2^Q\ M^Q \wedge v \in (T_P \cup T_Q)^*$

Thus, $M_0 v^{P\|Q} M \wedge v \in (T_P \cup T_Q)^*$

which means $v \in L(M_0, P \| Q)$.

q.e.d

Proof of Theorem 4.1(2) : $L(M_0, P|||Q) = L(M_0^P, P)|||L(M_0^Q, Q)$

Let $v = t_0 t_1 \ldots t_{n-1} \in L(M_0, P|||Q)$. This means there is a marking M_n such that $M_0 v^{P|||Q} M_n$. By Definition 4.1 this is equivalent to the following conjunction:

$$\bigvee_{i=0,n-1} (M_i^P t_i^P M_{i+1}^P \wedge M_i^Q = M_{i+1}^Q) \vee (M_i^Q t_i^Q M_{i+1}^Q \wedge M_i^P = M_{i+1}^P)$$

for some markings $M_0, M_1, \ldots, M_n$.

Let $v_1 = t_{k_0} t_{k_1} \ldots t_{k_p}$ be a subsequence of v such that

$$M_{k_j}^P \; t_{k_j}^P \; M_{k_j+1}^P \wedge M_{k_j}^Q = M_{k_j+1}^Q \wedge t_{k_j} \in T_P$$

holds for $j=0,1,\ldots,p$ and let $v_2 = t_{l_0} t_{l_1} \ldots t_{l_q}$ be a subsequence of v such that

$$M_{l_j}^Q \; t_{l_j}^Q \; M_{l_j+1}^Q \wedge M_{l_j}^P = M_{l_j+1}^P \wedge t_{l_j} \in T_Q$$

By definition of v_1 : $M_l^P = M_{l+1}^P$ for $k_j < l < k_{j+1}$ and
by definition of v_2 : $M_l^Q = M_{l+1}^Q$ for $l_j < l < l_{j+1}$, therefore

$$M_{k_0}{}^P v_1^P M_{k_p+1}^P \quad \text{and} \quad M_{l_0}{}^Q v_2^Q M_{l_q+1}^Q$$

Thus, $v_1 \in L(M_{k_0}^P, P)$ and $v_2 \in L(M_{l_0}^Q, Q)$.

Notice that $M_0 = M_{k_0}^P \cup M_{l_0}^Q$, hence

$M_0^P = M_{k_0}^P$ and $M_0^Q = M_{l_0}^Q$. Therefore

$v_1 \in L(M_0^P, P)$ and $v_2 \in L(M_0^Q, Q)$.

By definition of interleaving for traces [Hoa 83], v is an interleaving of v_1 and v_2 ("v interleaves(v_1, v_2)")

thus, $v \in L(M_0^P, P)|||L(M_0^Q, Q)$.

Let $v = t_0 t_1 \ldots t_{n-1} \in L(M_P, P) ||| L(M_Q, Q)$ where M_P, M_Q are markings in P and Q respectively. Then v is an interleaving of some sequences:

$v_1 = t_{k_0} t_{k_1} \ldots t_{k_p} \in L(M_P, P)$

$v_2 = t_{l_0} t_{l_1} \ldots t_{l_q} \in L(M_Q, Q)$

Thus, there exist markings M'_P and M'_Q such that

$M_P v_1^P M'_P$ and $M_Q v_2^Q M'_Q$

This means there are markings

$M_{Pk_0}, M_{Pk_1}, \ldots M_{Pk_p}$ in P and markings

$M_{Ql_0}, M_{Ql_1}, \ldots M_{Ql_q}$ in Q such that

(1) $\forall_{j=0,p-1}\ M_{Pk_j} t_{k_j}^P\ M_{Pk_{j+1}}$ and

(2) $\forall_{j=0,q-1}\ M_{Ql_j} t_{l_j}^Q\ M_{Ql_{j+1}}$ where

$M_{Pk_0} = M_P$ $\qquad M_{Ql_0} = M_Q$

$M_{Pk_p} = M'_P$ $\qquad M_{Ql_q} = M'_Q$

Due to $S_P \cap S_Q = \emptyset$, one can define markings $M_0, M_1, \ldots M_n$ in $P|||Q$ as follows:

$M_0 = M_{Pk_0} \cup M_{Ql_0}$ and for $i=1,2,\ldots n-1$

$$M_i = \begin{cases} M_{Ql} \cup M_{Pk_{j+1}} & \text{if } k_j < i < k_{j+1} \text{ for a certain } j < p \\ \\ M_{Pl} \cup M_{Ql_{j+1}} & \text{if } l_j < i < l_{j+1} \text{ for a certain } j < q \end{cases}$$

Notice that this definition is correct, since every $i=1,2,\ldots,n-1$ belongs to exactly one of open intervals (k_j,k_{j+1}) for $j=0,1,\ldots,p-1$, (l_j,l_{j+1}) for $j=0,1,\ldots,q-1$.

From definition of M_i : $M_0^P = M_P$ and $M_0^Q = M_Q$ and for $i=1,2,\ldots n-1$:

$$M_i^P = \begin{cases} M_{Pl_j} & \text{if} \quad l_j < i < l_{j+1} \\ \\ M_{Pk_{j+1}} & \text{if} \quad k_j < i < k_{j+1} \end{cases}$$

$$M_i^Q = \begin{cases} M_{Qk_j} & \text{if} \quad k_j < i < k_{j+1} \\ \\ M_{Ql_{j+1}} & \text{if} \quad l_j < i < l_{j+1} \end{cases}$$

Thus, if $l_j < i < l_{j+1}$ $(j=0,1,\ldots,q-1)$ then

$M_i^Q = M_{Ql_{j+1}}$ (= constant marking $M_{Ql_{j+1}}$) and

$M_i^P = M_{Pl_j}$ hence, by (1), $M_i^P t_i^P M_{i+1}^P$ holds. Therefore:

(3) $M_i^P t_i^P M_{i+1}^P \wedge M_i^Q = M_{i+1}^Q$ for $l_j < i < l_{j+1}$.

Similarly:

(4) $M_i^Q t_i^Q M_{i+1}^Q \wedge M_i^P = M_{i+1}^P$ for $k_j < i < k_{j+1}$.

From (3) and (4):

$$(M_i^P t_i^P M_{i+1}^P \wedge M_i^Q = M_{i+1}^Q) \vee (M_i^Q t_i^Q M_{i+1}^Q \wedge M_i^P = M_{i+1}^P)$$

for every $i=0,1,\ldots,n-1$. Applying Definition 4.1 (of $|||$ for nets) we obtain:

$\forall_{i=0,n-1}\ M_i t_i^{P|||Q} M_{i+1}$ which means $M_0 v^{P|||Q} M_n$ thus $v \in L(M_0, P|||Q)$.

q.e.d

Proof of Theorem 4.1(3) : $L(M_0, P[]Q) = L(M_0^P, P) \cup L(M_0^Q, Q)$

<u>Let</u> $v \in L(M_0, P[]Q)$. This means $v \in (T_P \cup T_Q)^*$ and there is a marking M such that $M_0 v^{P[]Q} M$. Hence, by Definition 4.1 (of $[]$ for nets), $v \in T_P^* \cup T_Q^*$, and

$(M_0^P\ v^P M^P \wedge M_0^Q = M^Q) \vee (M_0^Q\ v^Q M^Q \wedge M_0^P = M^P)$

Therefore, $v \in L(M_0^P, P) \cup L(M_0^Q, Q)$.

<u>Let</u> $v \in L(M_P, P) \cup L(M_Q, Q)$, where M_P, M_Q are markings in P and Q respectively. Suppose $v \in L(M_P, P)$. Thus, $v \in T_P^*$ and $M_P v^P M_P'$ for a certain M_P'. Defining (due to $S_P \cap S_Q = \emptyset$):

$M_0 = M_P \cup M_Q$

$M = M_P' \cup M_Q$

we obtain:

$M_0^P = M_P$ $\qquad\qquad M_0^Q = M_Q$

$M^P = M_P'$ $\qquad\qquad M^Q = M_Q$

Therefore:

$M_0^P\ v^P M^P \wedge M_0^Q = M^Q$ and thus,

$((M_0^P\ v^P M^P \wedge M_0^Q = M^Q) \vee (M_0^Q\ v^Q M^Q \wedge M_0^P = M^P)) \wedge v \in T_P^* \cup T_Q^*$

holds. Hence, $M_0 v^{P[]Q} M$ which means $v \in L(M_0, P[]Q)$.

The is shown analogously for $v \in L(M_Q, Q)$.

Proof of Theorem 6.1 : $L(M_D, D) = \{\lambda\} \cup \{t\} \bigcup_{M \in \mathbb{M}_{D'}} L(M, D')$

<u>Let</u> $v \in L(M_D, D)$. This means $M_D v^D M'$ and $v = tu$ for certain M' and u, since the only firable at M_D transition is t. Thus,

$\exists_{M''} : M_D t^D M'' \wedge M'' u^D M'$

Notice that $M'' u^D M' \iff M''^{D'} u^{D'} M'^{D'}$

(since after firing t, place s_D will always hold 0)

and $M''^{D'} \in \mathbb{M}_{D'}$ hence $u \in \bigcup_{M \in \mathbb{M}_{D'}} L(M, D')$.

Therefore $v \in \{\lambda\} \cup \{t\} \bigcup_{M \in \mathbb{M}_{D'}} L(M, D')$.

<u>Let</u> $v \in \{\lambda\} \cup \{t\} \bigcup_{M \in \mathbb{M}_{D'}} L(M, D')$. Thus, $v = \lambda$ or $v = tu$

where $u \in \bigcup_{M \in \mathbb{M}_{D'}} L(M, D')$ which means $M u^{D'} M'$ for some

$M \in \mathbb{M}_{D'}$ and M'. Therefore

$M_D t^D (M \cup \{\langle s_D, 0\rangle\}) \wedge (M \cup \{\langle s_D, 0\rangle\}) u^D (M' \cup \{\langle s_D, 0\rangle\})$

hence $M_D v^D (M' \cup \{\langle s_D, 0\rangle\})$ which means $v \in L(M_D, D)$.

q.e.d

References

[Age-Fly 76] T.Agerwala, M.Flynn. Comments on capabilities, limitations and "correctness" of Petri Nets. Proc. of I-st Ann.Symp. of Computer Architecture. Univ of Florida, 1973 pp 81-86

[Cza 84] L.Czaja. Making nets abstract and structured. Oxford Univ Comp Lab PRG Internal report Jan. 1984

[Gol-Myc 84] U.Goltz, A.Mycroft. On the relationship of CCS and Petri nets - manuscript 1984

[Hoa 83] C.A.R.Hoare. Notes on Communicating Sequential Processes. Oxford Univ Comp Lab Technical Monograph PRG-33, Aug.1983

[Hoa-Olde 83] C.A.R.Hoare E-R.Olderog. Specification-oriented semantics for Communicating Processes. Oxford Univ Comp Lab. Thecnical Monograph PRG-37, Feb.1964

[Hoa 84] C.A.R.Hoare. Programs are predicates. Oxford Univ Comp Lab PRG Internal report 1984

[Kos 82] R.S.Kosaraju. Decidability of reachability in Vactor Addition Systems. Proc of the Fourteen Annual ACM Symposium on Theory of Computing. San Francisco. Calif. May 5-7 1982

[Lau 75] P.E.Lauer. A project to investigate a design technique for asynchronous systems of processes. Univ of Newcastle upon Tyne memo 1975

[Maz 77] A.Mazurkiewicz. Concurrent program schemes and their interpretations. Arhus Univ. Internal report 1977

[Petri 76] C.A.Petri. Non-sequential processes. GMD Internal report Bonn, 1976

[Win 77] J.Winkowski. An algebraic characterization of the behaviour of non-sequential systems. Inf.Proc.Lett. 6(4)8(1977)

[Wins 84] G.Winskel. A new definition of morphism on Petri nets. to appear STACS 1984