ISSN 0105-8517

A THEORY FOR BIPOLAR SYNCHRONISATION SCHEMES

H.J. Genrich*
P.S. Thiagarajan

* Gesellschaft f. Mathematik und Datenverarbeitung,
5205 St. Augustin, Schloss Birlinghoven, W. Germany.

DAIMI PB-158
March 1983

Computer Science Department
AARHUS UNIVERSITY
Ny Munkegade – DK 8000 Aarhus C – DENMARK
Telephone: 06 – 12 83 55

A THEORY OF BIPOLAR SYNCHRONISATION SCHEMES

H.J. Genrich and P.S. Thiagarajan
Gesellschaft für Mathematik und Datenverarbeitung
5205 St. Augustin, Schloss Birlinghoven
Postfach 1240, W. Germany

A THEORY OF

BIPOLAR SYNCHRONISATION SCHEMES

Abstract

The aim is to better understand the relationships between choice and concurrency that lead to the good behaviour of distributed systems. In order to do so, we formulate a model based on Petri nets and develop its theory. The model is called bipolar synchronisation schemes (bp schemes) and the theory we construct is mainly devoted to synthesizing, in a systematic fashion, all well behaved bp schemes. We also provide a computational interpretation of well behaved bp schemes. Through this interpretation the insights gained by developing the theory of bp schemes can be transferred to concurrent programs.

Contents

0. INTRODUCTION

This paper presents the theory of a model of concurrent systems called bipolar synchronization schemes (bp schemes). The model is based on a class of Petri nets. The theory we develop is oriented toward the synthesis of well-behaved bp schemes.

The main motivation is to deepen our understanding of the relationships between choice and concurrency in a systems context. More specifically we wish to understand why certain means of combining alternative and concurrent courses of actions lead to "good" distributed systems. To achieve this goal, we ask: What are the proper means of combining alternative and concurrent courses of actions that result in only and all "good" systems?

A crude translation in programming terms would sound as follows. Consider a class of concurrent programs in which the order of execution of the statements is solely determined by the control structures of the programs; there is no scheduler, hidden in the background, which ensures that the value of a variable is not altered in two portions of the program concurrently during execution. Suppose that for each program its control structure is composed out of the ";" construct which enforces the "followed by" ordering relation; the IF (FI) construct which signals the initiation (termination) of one out of a set of alternative courses of actions; the PARBEGIN (PAREND) construct which signals the initiation (termination) of a set of concurrent courses of actions. We then ask: How should these constructs be put together so that the resulting program is "consistent"? By consistent, we mean at this stage that the control flow does not lead to deadlocks and it is guaranteed that two statements which share a variable will never execute concurrently.

We should like to caution the reader that we have given the above translation merely in order to convey our concerns in more familiar terms. Our statement of the problem in programming terms is naive and misleading in a number of important respects. In the latter part of the paper however, we attach a computational interpretation to our model. It will then be possible to discuss and evaluate in more precise terms, what implications our work has for the theory of concurrent programs.

Returning to our original line of thought, the problem as stated is very vague, general and hard. Hence as a first step, we propose to carry out our study in a specific and restricted setting. We formulate the bp scheme model precisely in order to set up such a setting.

Now for a few words about the model. A bp scheme consists of a directed graph which represents the structure of the system under study. Token distributions over the arcs capture the distributed, generalised states of the system. We consider directed graphs which have two kinds of nodes called <u>∇-nodes</u> and <u>&-nodes</u>. ∇-nodes are used to model the branching and merging of alternative courses of actions. &-nodes are used to model the forking and joining of concurrent courses actions. We place two kinds of tokens called H-tokens (HIGH tokens) and L-tokens (LOW tokens) on the arcs to represent the distributed state of the system. H-tokens are used to denote the commissions of actions. L-tokens are used to <u>explicitly</u> represent the <u>omissions</u> of actions (as a result of choices made between alternative courses of actions).

The dynamics of the system is modelled by the <u>transition rules</u> which specify how local transformations in the token distribution are effected through the firings of nodes. The transition rules exploit the notion of L-tokens to capture what a good flow of control is, in the presence of choice (∇) and division (&) of work in a concurrent system. This leads to the concept of a <u>well behaved</u> bp scheme. We then study the problem of synthesizing well behaved bp schemes.

Formulating interrelated models of concurrent systems is a major activity within the net theory of systems and processes [19, 4]. In this sense, our work may be viewed as a fragment of this general theory of systems. In particular, a bp scheme can be interpreted in terms of the basic system model of net theory called

Condition/Event systems [4]. Our schemes can also be viewed as a sub-class of the higher level net model called Predicate/ Transition nets [5] and the closely related Coloured Petri nets [10], and Relation nets [21].

In the context of the synthesis problem we propose to attack, the papers by Jadwani and Jump [12] and Valette [24] deserve mention. In both these papers, the problem of refining free choice nets (which are a restricted class of Petri nets) while preserving properties such as liveness and safety is considered. Our work is also, at least in spirit, related to that of Lauer et al. [15] who investigate the syntax of path expressions that have the so-called adequacy property. Finally, the approach taken by Milner [17] who studies concurrent systems from the very beginning from the synthesis standpoint has also been a source of inspiration.

The computational interpretation that we attach to well behaved bp schemes yields a kind of flow chart model of a class of "well formed" concurrent programs. This part of our research has been strongly influenced by the work Mazurkiewicz has carried out for a more general class of nets [16]. The key difference is the way in which our interpretation is interwoven with the theory of the underlying class of nets. It will be convenient to postpone the review of other related pieces of work to the concluding part of the paper.

The organisation of the paper is as follows. In the next section, we rapidly review the theory of live and safe marked graphs which are a well understood sub-class of nets. In section 2, we formulate the bp scheme model as a generalisation of live and safe marked graphs. We then define the notion of well behavedness. In sections 3 and 4 we develop some analysis results concerning bp schemes. In section 3, we show where bp schemes fit within the hierarchy of known classes of nets. Exploiting their respective theories, we then obtain an important necessary condition, essentially in terms of its structure, for a bp scheme to be well behaved. In section 4, we prove a sort of all-or-none property about the state space of a well behaved bp scheme and derive some useful consequences of this property.

Though these results are interesting in their own right, their main function is to aid in solving the synthesis problem. This is done in the subsequent two sections. In section 5 we present our synthesis procedure which basically consists of starting with a "trivial" well behaved bp scheme and repeatedly applying a small set of transformation rules. We show that this construction is consistent in that it yields only well behaved bp schemes. In section 6, the only part of the paper which is rather technical, we establish the completeness of the synthesis procedure; every well behaved scheme can be constructed using our technique.

Section 7 provides a computational interpretation for a class of well behaved bp schemes. Stated briefly, we allocate variables to the arcs, test predicates to the ∇-nodes and operations to the &-nodes of a scheme. What then obtains is a flow chart representation of a concurrent program, which by construction, is guaranteed to satisfy a set of consistency criteria.

In the last section, we summarize the contents of the paper and take one more look at related work.

1. LIVE AND SAFE MARKED GRAPHS

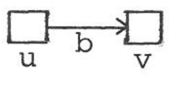
We shall formulate bp schemes as a generalisation of live and safe marked graphs. Hence as a first step, we briefly review the relevant portions of the theory of live and safe marked graphs, the earliest part of "token mathematics" of net theory. In doing so, we shall also develop some basic terminology that we need through the rest of the paper.

In what follows, $\mathbb{N}$ and $\mathbb{Z}$ denote the set of non-negative integers and integers respectively. If X is a set then $|X|$ denotes its cardinality and X^* the free monoid generated by X. If σ is a sequence of symbols then, risking confusion, we will let $|\sigma|$ denote the length of σ. For a symbol x, $|\sigma|_x$ is the number of times x appears in the sequence σ. λ is the null sequence and $|\lambda|=0$.

For our purposes, directed graphs which in general may contain multiple arcs and self-loops are required. So our notion of a directed graph (digraph) is:

Definition 1.1 A *digraph* is a quadruple $G = (V,A;Q,Z)$ where:

- V is a finite non-empty set of *vertices* (*nodes*);
- A is a finite non-empty set of *arcs* (*edges*) that is disjoint from V;
- $Q:A \rightarrow V$ is the *source* function;
- $Z:A \rightarrow V$ is the *target* function. ∎

In diagrams, the nodes are drawn as boxes and the arcs as lines with arrowheads such that 

stands for $Q(b)=u$ and $Z(b)=v$.

Let $G = (V,A;Q,Z)$ be a digraph and $v \in V$. Then,

${}^\bullet v = \{a \in A \mid Z(a)=v\}$ is the *input arcs* of v

$v^\bullet = \{a \in A \mid Q(a)=v\}$ is the *output arcs* of v.

A *directed path* is a *non-null* sequence of arcs $\Pi = a_0a_1...a_n$ such that for $0 \leq i < n$, $Z(a_i) = Q(a_{i+1})$. If $Q(a_0) = u$ and $Z(a_n) = v$, then Π is said to be a directed path of length n+1 *from* u *to* v. Π is said to *pass through* (*contain*) the arc b, iff $|\Pi|_b > 0$. Π is said to be a *directed circuit* iff $Q(a_0) = Z(a_n)$. Π is called an *elementary* directed circuit iff Π is a directed circuit and no proper subsequence of Π is also a directed circuit. If Π is not a directed circuit and no proper subsequence of Π is also a directed circuit, then Π is said to be an *acyclic* directed path.

The digraph G = (V,A;Q,Z) is said to be *connected* iff for every non-trivial partition {U,W} of V ($U,W \neq \emptyset$, $U \cap W = \emptyset$, $U \cup W = V$) there is an arc b with $Q(b) \in U$ and $Z(b) \in W$ *or* $Q(b) \in W$ and $Z(b) \in U$. G is said to be *strongly connected* iff for non-trivial partition {U,W} of V, there are arcs b_1 and b_2 such that $Q(b_1) \in U$, $Z(b_1) \in W$, $Q(b_2) \in W$ and $Z(b_2) \in U$.

All digraphs considered in this paper, unless otherwise stated, are assumed to be connected. In addition, we will be dealing with only directed paths. Hence, for brevity, we almost always drop the qualifying "directed" in talking about paths, circuits etc.

Let G = (V,A;Q,Z) be a digraph. Then a *marking* of G is a function $M: A \rightarrow \mathbb{N}$. If $b \in A$ and $M(b) = k$, then in diagrams this will be indicated by placing k *tokens* on b. An example of a digraph together with a marking is shown in fig. 1.1

Fig. 1.1

Definition 1.2 Let G = (V,A;Q,Z) be a digraph, M a marking of G and $\Pi = a_0a_1...a_n$ a path of G. Then,

(1) $M(\Pi)$ denotes the *token load* of Π under M and is defined as: $M(\Pi) = \sum_{i=0}^{n} M(a_i)$.

(2) Π is said to be *token free* iff $M(\Pi) = 0$.

(3) Π is called a *basic circuit* of G at M iff Π is an elementary circuit of G and $M(\Pi) = 1$. ∎

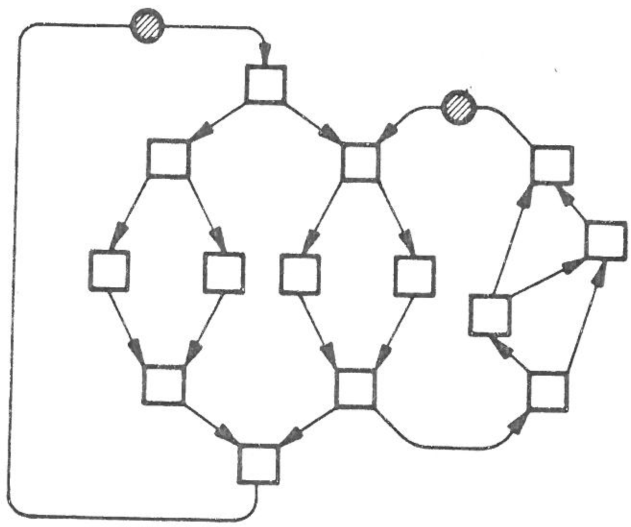


Fig. 1.1

Markings of a digraph may be changed through the firing of nodes according to the following rule.

Definition 1.3 Let $G = (V,A;Q,Z)$ be a digraph, M a marking of G and v a node. Then the transition rule for marked graphs is applicable to v (v may fire; v is firable) at M iff all input arcs of v are non-zero marked ($M(b)>0$ for each $b \in {}^{\bullet}v$). When v fires at M, a new marking M' is reached which is given by

$$\forall b \in A: \quad M'(b) = \begin{cases} M(b)-1, & \text{if } b \in {}^{\bullet}v - v^{\bullet} \\ M(b)+1, & \text{if } b \in v^{\bullet} - {}^{\bullet}v \\ M(b), & \text{otherwise} \end{cases}$$

∎

The transition from M to M' through a firing of v is denoted by $M[v>M'$. This notation is extended to a sequence of nodes as follows. Let M_0 be a marking of the digraph $G = (V,A;Q,Z)$ and $\sigma = v_1 v_2 \ldots v_n \in V^*$. Then σ is firable at M_0 (σ is a firing sequence at M_0) iff there exist markings $M_1, M_2, \ldots, M_{n+1}$ of G such that for $0 \leq i \leq n$, $M_i[v_i>M_{i+1}$. The change from M_0 into M_{n+1} through the firing of σ starting from M_0 is denoted by $M_0[\sigma>M_{n+1}$. By convention, for every marking M of G, $M[\lambda>M$.

In general, a set of nodes may fire independently in one step, at a marking. Consequently changes in a marking are, in general, effected through partially ordered sets rather than sequences of node firings. In this paper however, we are concerned mainly with establishing certain properties of the behaviours of our systems and not so much with the behaviours themselves. Hence for establishing our results we will employ firing sequences as the primary tool and on occasion rely on the notion of partially ordered sets of node firings.

We now specify two state spaces defined by a marking of a digraph.

Definition 1.4 Let $G = (V,A;Q,Z)$ be a digraph and M a marking of G. Then,

(1) $[M>$, the forward marking class of G defined by M is the smallest set of markings of G given by:

- $M \in [M>$
- If $M' \in [M>$, $v \in V$ and M'' is a marking of G such that $M'[v>M''$, then $M'' \in [M>$.

(2) $[M]$, the full marking class of G defined by M is the smallest set of markings of G given by:

- $M \in [M]$
- If $M' \in [M]$, $v \in V$ and M'' is a marking of G such that $M'[v>M''$ or $M''[v>M'$, then $M'' \in [M]$. ∎

We can at last start looking at marked graphs.

Definition 1.5 A marked graph is a quintuple $MG = (V,A;Q,Z,M_0)$ where $G = (V,A;Q,Z)$ is a digraph and M_0 is a marking of G called the initial marking. ∎

Liveness and safety are two fundamental properties of marked graphs.

Definition 1.6 Let $MG = (V,A;Q,Z,M_0)$ be a marked graph. Then,

(1) MG is live iff for every marking $M' \in [M_0>$ and for every node $v \in V$, there is a $M'' \in [M'>$ such that v is firable at M''.

(2) MG is safe iff for every marking $M' \in [M_0>$ and every arc $b \in A$, $M'(b) \leq 1$. ∎

The marked graph shown in fig. 1.1 is live and safe. Let $G = (V,A;Q,Z)$ be a digraph and M a marking of G. Then G is said to be live (safe) at M iff the marked graph $(V,A;Q,Z,M)$ is live (safe). The results concerning marked graphs that we now mention without proofs have been assembled from [3] and [6].

Theorem 1.1 Let $MG = (V,A;Q,Z,M_0)$ be a marked graph, Π a path in MG leading from node u to v and σ a firing sequence at M_0 leading to the marking M $(M_0[\sigma>M)$. Then,

$$M(\Pi) = M_0(\Pi) + |\sigma|_u - |\sigma|_v$$ ∎

Theorem 1.2 Let $MG = (V,A;Q,Z,M_0)$ be a marked graph and Π a circuit of MG. Then for every marking $M \in [M_0]$, $M(\Pi) = M_0(\Pi)$. ∎

Theorem 1.3 Let $MG = (V,A;Q,Z,M_0)$ be a marked graph. Then,

(1) MG is live iff for every circuit Π of MG, $M_0(\Pi)>0$.

(2) MG is live iff $G = (V,A;Q,Z)$ is live at every $M \in [M_0]$. ∎

Theorem 1.4 Let MG $(V,A;Q,Z,M_0)$ be a marked graph. MG is live and safe iff

(1) For every circuit Π of MG, $M_0(\Pi)>0$.

(2) For every arc b, there is a circuit Π with $M(\Pi)=1$ which passes through b. ∎

Theorem 1.5 Let $MG = (V,A;Q,Z,M_0)$ be a live and safe marked graph. Then,

(1) For every $M \in [M_0]$, $G = (V,A;Q,Z)$ is live and safe at M.

(2) For every $M,M' \in [M_0]$, $M' \in [M>$.

(3) MG is covered by basic circuits, i.e. G is covered by elementary circuits with token load 1 at M_0. Consequently, G is strongly connected. ∎

For stating the next result, we need some additional notions.

Let $G = (V,A;Q,Z)$ be a digraph and M a marking of G. Then the binary relation $<_M \subseteq V \times V$ is given by

$\forall v_1, v_2 \in V$: $v_1 <_M v_2$ iff there is a token free path from v_1 to v_2 at M.

Now let $MG = (V,A;Q,Z,M_0)$ be a live marked graph and $M \in [M_0]$. Then $<_M$ is irreflexive since we demand a path be composed out of a non-null sequence of arcs and theorem 1.3 tells us that for every circuit Π, $M(\Pi)>0$. $<_M$ is clearly transitive which then implies that $<_M$ is asymmetric. Thus $(V;<_M)$ is a strict partial order. We note that the minimal elements of $(V;<_M)$ are exactly the set of nodes that are concurrently firable at M. Given $M \in [M_0]$ and $v \in V$, the relation $<_M$ can be used to determine the "least work" one has to do to obtain a marking M' at which v becomes firable.

To bring this out, we need to introduce an equally convenient idea. Let $MG = (V,A;Q,Z,M_0)$ be a marked graph and $v \in V$. Let $M, M' \in [M_0]$ and $\sigma \in V^*$ such that $M[\sigma > M'$. Then σ is said to be a <u>v-enabling</u> sequence of M iff v is firable at M'. σ is a <u>minimal v-enabling</u> sequence at M iff it is v-enabling sequence at M and for every v-enabling sequence σ' at M, $|\sigma| \leqq |\sigma'|$.

There is a close and general relationship between the two notions introduced above. We shall however state the relevant result only for live marked graphs.

<u>Theorem 1.6</u> Let $MG = (V,A;Q,Z,M_0)$ be a live marked graph, $M \in [M_0]$ and $v \in V$. Then,

(1) There is a minimal v-enabling sequence at M.

(2) If σ is a minimal v-enabling sequence at M, then $\forall v' \in V$:

$$|\sigma|_{v'} = \begin{cases} 1, & \text{if } v' <_M v \\ 0, & \text{otherwise} \end{cases}$$

<u>Proof</u> The first part is trivial. To prove the second part let σ be a minimal v-enabling sequence at M. Set $V_1 = \{v' \in V \mid v' <_M v\}$. Suppose that $v' \in V_1$. Then by theorem 1.1, $|\sigma|_{v'} > 0$. We shall now show that, starting from M, it is sufficient to fire every node in V_1 exactly once to obtain a marking M' at which v becomes firable. To this end, let v_1 be a minimal element of $(V_1, <'_M)$ where $<'_M$ is $<_M$ restricted to $V_1 \times V_1$. v_1 exists because V_1 is a finite set. Now v_1 is firable at M. Let $M[v_1>M_1$. Set $V_2 = \{v' \in V \mid v' <_{M_1} v\}$. It is easy to see that $V_2 = V_1 - \{v_1\}$ and the rest of the proof is routine. ∎

The last result we wish to mention deals with extremal markings.

<u>Definition 1.7</u> Let $MG = (V,A;Q,Z,M_0)$ be a marked graph and $M \in [M_0]$. Let $v \in V$. M is said to be a <u>v-extremal</u> marking iff v is the only node which is firable at M. ∎

<u>Theorem 1.7</u> Let $MG = (V,A;Q,Z,M_0)$ be a live and safe marked graph, $M \in [M_0]$ and $v \in V$. Then there exists a v-extremal marking $M' \in [M>$. Moreover v is the minimum element of the strict partial order $(V;<_M)$. In other words, at M there is a token free path from v to every node $w \in V - \{v\}$.

<u>Proof</u> Follows easily from the previous results. ∎

The only result which is not explicitly mentioned in the published literature is:

<u>Theorem 1.8</u> Let $MG = (V,A;Q,Z,M_0)$ be a live and safe marked graph. Let $v \in V$ and $M_1, M_2 \in [M_0]$ such that M_1 and M_2 are v-extremal. Then $M_1 = M_2$.

Proof As pointed out earlier (Theorem 1.5), $G = (V,A;Q,Z)$ is strongly connected. Hence G is covered by the set of paths of finite length that lead to v. As a result, it is sufficient to show that every finite path leading to v is marked in the same way by M_1 and M_2. To this end let $\Pi = a_0a_1...a_n$ be a path with $Z(a_n) = v$. The proof is by induction on $l = |\Pi|$ (= n+1).

1) $l=1$ v is firable at M_1 and M_2; MG is safe. Hence $M_1(a_0) = 1 = M_2(a_0)$.

2) $l>1$ Then $\Pi_1 = a_1a_2...a_n$ is a path of length $l-1$ leading to v. By the induction hypothesis, for $1 \leq i \leq n$, $M_1(a_i) = M_2(a_i)$. Suppose that $Q(a_0) = v$. Then Π is a circuit and by theorem 1.2, $M_1(\Pi) = M_2(\Pi)$. We can now conclude that $M_1(a_0) = M_2(a_0)$.

So assume that $Q(a_0) = u \neq v$. Then there is a token free path Π' from v to u at M_1 since M_1 is v-extremal. Let Π'' be the concatenation of Π' with Π (Π' followed by Π). Then Π'' is a circuit and once again by theorem 1.2, $M_1(\Pi'') = M_2(\Pi'')$. This implies that, $M_1(\Pi'') = M_1(\Pi') + M_1(a_0) + M_1(\Pi_1) = M_2(\Pi') + M_2(a_0) + M_2(\Pi_1) = M_2(\Pi'')$. By the induction hypothesis, $M_1(\Pi_1) = M_2(\Pi_1)$; by the construction of Π', $M_1(\Pi') = 0$; by definition, $M_2(\Pi') \geq 0$. Hence $M_1(a_0) \geq M_2(a_0)$.

In a similar fashion, by considering a token free path from v to u at M_2, we can show that $M_2(a_0) \geq M_1(a_0)$. Hence $M_1(a_0) = M_2(a_0)$. ∎

We conclude this review of marked graphs with the adoption of a useful convention. Let $MG = (V,A;Q,Z,M_0)$ be a safe marked graph, and $M \in [M_0]$. Then for every arc $b \in A$, $M(b)=0$ or $M(b)=1$. Thus M is the characteristic function of the subset of arcs which carry a token at M. With this in mind, from now on we shall represent a marking in the full marking class of a safe marked graph in either one of two equivalent ways. As a subset of the arcs which carry a token at the given marking or as the corresponding characteristic function. Thus for example, if $(V,A;Q,Z,M)$ is a safe marked graph and $v \in V$, ${}^{\bullet}v \subseteq M$ would imply that v is firable at M. If $M[v>M'$, then $M' = (M - ({}^{\bullet}v)) \cup v^{\bullet}$.

2. THE MODEL

In our formalism, the structure of a system is represented by a digraph with two kinds of nodes.

Definition 2.1 A bipolar graph (bp graph) is a quintuple BG = $(V_\nabla, V_\&, A; Q, Z)$ where $V_\nabla \cap V_\& = \emptyset$ and $(V_\nabla \cup V_\&, A; Q, Z)$ is a digraph. ∎

The set of nodes of a bp graph BG, $V = V_\nabla \cup V_\&$ is divided into ∇-nodes and &-nodes. In diagrams, a ∇-node <&-node> will be drawn as a box carrying the inscription ∇ <&>. The graph theoretic terminology and notations that were introduced in the previous section are carried over to bp graphs in the obvious way. The underlying digraph of BG will be denoted as $\widetilde{BG}$.

As mentioned earlier, ∇-nodes will be used to model the branching and merging of alternative courses of actions. The &-nodes will be used to model the forking and joining of concurrent courses of actions. In addition, anticipating the contents of section 7, ∇-nodes will represent the tests and &-nodes the atomic actions (transformations of variables) associated with the system. To illustrate the main idea it is perhaps useful to consider some examples.

Loosely speaking, the construct if $P_1 \rightarrow O_1 \ \square \ P_2 \rightarrow (O_2; O_3)$ fi will be, in our approach, modelled by the subgraph shown in fig. 2.1(a). The construct parbegin $(O_1; O_2) \parallel O_3$ parend will be modelled by the subgraph shown in fig. 2.1(b). However, in our theory, the control flow represented by the subgraphs shown in fig. 2.1(c) and (d) will have the same prestige as those of (a) and (b).

Fig. 2.1

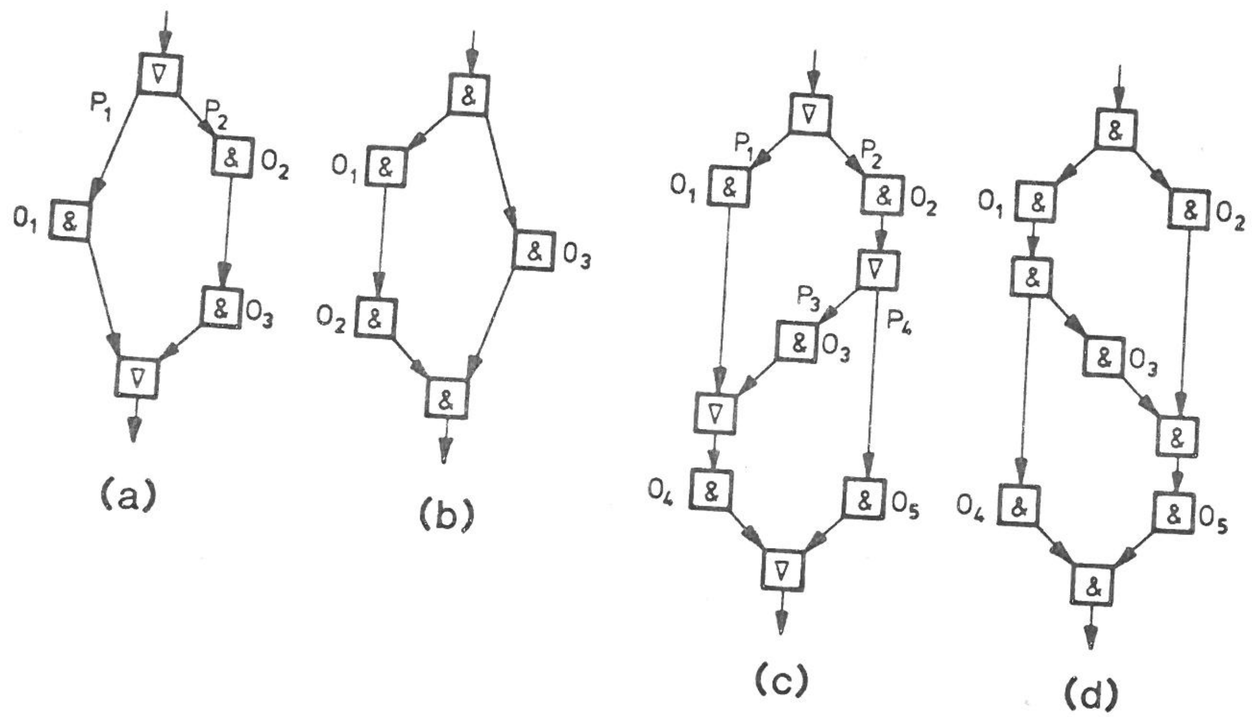


Fig. 2.1

What we are after is a set of notions through which we can formalize our intuition that the control flow represented in fig. 2.2(a) is a "good" combination of choice and concurrency whereas the control flow represented in fig. 2.2(b) is a "bad" one. (In fig. 2.2, for the sake of convenience, we have abstracted away the tests and actions.)

Fig. 2.2

The notion of a marking (control state), the firing rules (propagation of control) and the notion of well behavedness are together meant to serve that purpose.

We shall start with markings. Our notion of a marking, to represent the distributed control state of a system will enable us to view our model as a gentle generalisation of live and safe marked graphs.

<u>Definition 2.2</u> Let $BG = (V_{\nabla}, V_{\&}, A; Q, Z)$ be a strongly connected bp graph. A <u>marking</u> of BG is an ordered pair $M = (M_H, M_L)$ where $M_H, M_L \subseteq A$ and $M_H \cap M_L = \emptyset$ such that $(V_{\nabla} \cup V_{\&}, A; Q, Z, M_H \cup M_L)$ is a live and safe marked graph. ∎

Now strong connectedness of a digraph is a necessary and sufficient condition for a live and safe marking to exist [6] and recall that we have agreed to, where convenient, represent a marking of a sage marked graph through an appropriate subset of arcs. If $M = (M_H, M_L)$ is a marking of a bp graph BG, then the corresponding marking $M_H \cup M_L$ of the underlying digraph $\widetilde{BG}$ shall be denoted as $\widetilde{M}$.

Let $BG = (V_{\nabla}, V_{\&}, A; Q, Z)$ be a bp graph and $M = (M_H, M_L)$ be a marking of BG. Let b be an arc, $b \in A$. If $b \in M_H$ <M_L> then b is said to carry an <u>H-token</u> <u><L-token></u> under M. In diagrams, this will be indicated by placing a dark <plain> token on the line representing b. An example of a bp graph together with a marking is shown in fig. 2.3.

Fig. 2.3

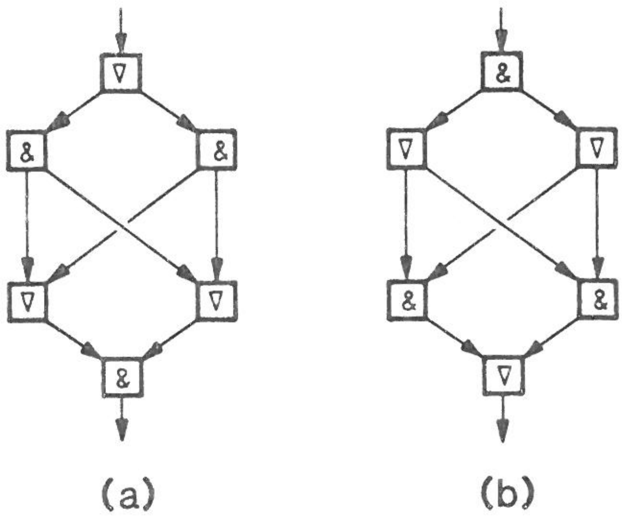


Fig. 2.2

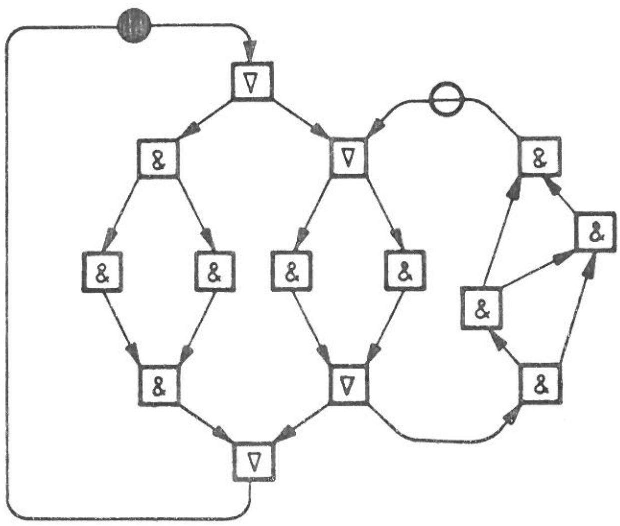


Fig. 2.3

Next we specify the firing rules by applying which a marking of a bp graph can be transformed into a new one. The definition might be easier to grasp, if the reader glances at the illustrative examples shown in fig. 2.3 and 2.4.

<u>Definition 2.3</u> Let $BG = (V_\nabla, V_\&, A; Q, Z)$ be a bp graph and $M = (M_H, M_L)$ a marking of BG. Let u be a &-node and v a ∇-node.

1) v is <u>enabled</u> (to fire) at M iff ${}^\bullet v \subseteq \tilde{M}$ and $|{}^\bullet v \cap M_H| \leq 1$.

2) u is enabled at M iff ${}^\bullet u \subseteq M_H$ or ${}^\bullet u \subseteq M_L$.

3) If a node (&-node or ∇-node) w fires, a new marking $M' = (M'_H, M'_L)$ is obtained which for the underlying digraph $\tilde{BG}$ and its marking $\tilde{M}$ corresponds to the firing rule for marked graphs. In other words,
   - $\tilde{M}' = (\tilde{M} - {}^\bullet w) \cup w^\bullet$. Moreover,
   - $\forall b \in A - ({}^\bullet w \cup w^\bullet)$: $b \in M'_H$ $\langle M'_L \rangle$ iff $b \in M_H$ $\langle M_L \rangle$.

4) If a node w is enabled and some input arc carries an H-token ($M_H \cap {}^\bullet w \neq \emptyset$), w may <u>H-fire</u>.
   When the ∇-node v H-fires, the new marking M' satisfies $|M'_H \cap v^\bullet| = 1$.
   When the &-node u H-fires, the new marking M' satisfies $u^\bullet \subseteq M'_H$.

5) If a node w is enabled and ${}^\bullet w \subseteq M_L$, w may <u>L-fire</u>.
   When w L-fires, the new marking M' satisfies $w^\bullet \subseteq M'_L$.

6) If $|M_H \cap {}^\bullet v| > 1$, then the ∇-node v is in <u>deadlock</u> at M.

7) If $M_H \cap {}^\bullet u \neq \emptyset$ and $M_L \cap {}^\bullet u \neq \emptyset$, then the &-node u is in deadlock at M.

8) A node which is in deadlock cannot become enabled any more. ∎

Fig. 2.4

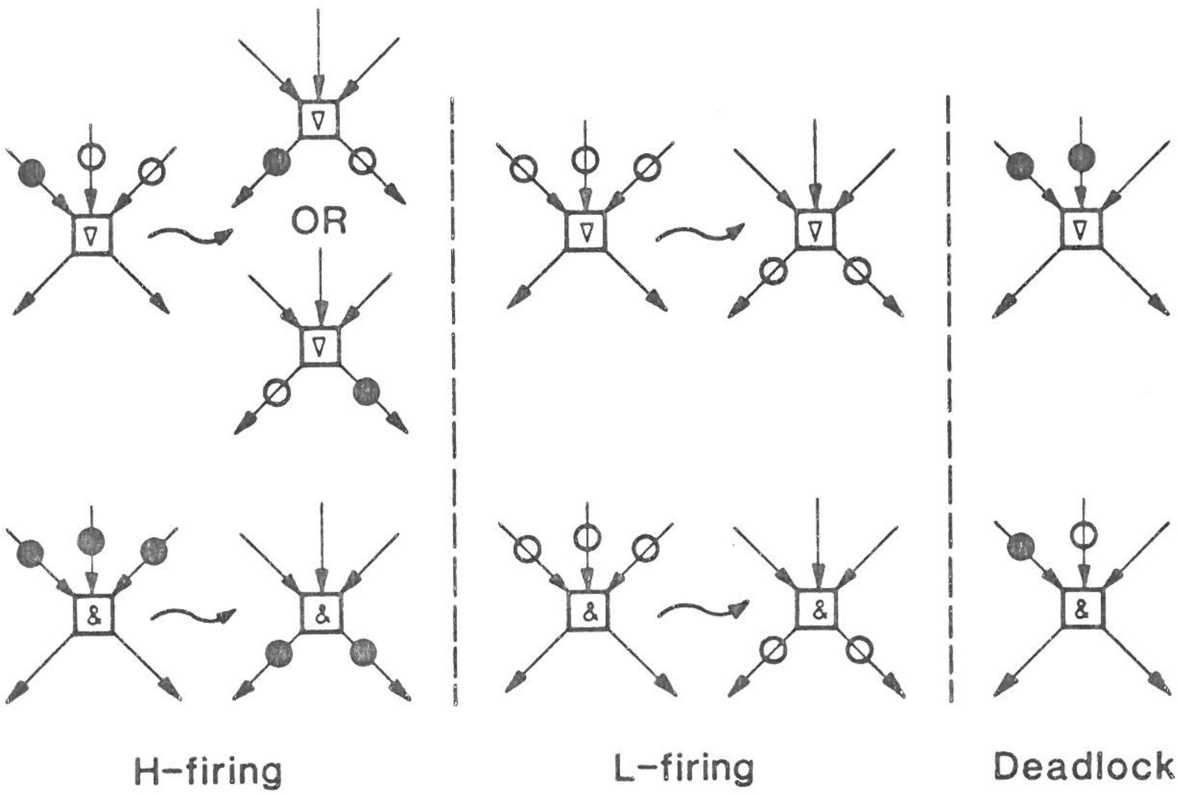


Fig. 2.4

First we note that if $M' = (M'_H, M'_L)$ is the result of a node firing at a marking of BG, then $\tilde{M}'$ is also a live and safe marking of $\widetilde{BG}$ and $M'_H \cap M'_L = \emptyset$. Thus M' is also a marking of BG.

Now for some explanations of the firing rules. When a ∇-node H-fires, the execution of one course of action and the omissions of the remaining ones within a set of alternatives are terminated at its input side, and at its output side a choice is made to follow one course and omit the remaining ones within another set of alternatives.

When a &-node H-fires, a set of concurrent courses of actions are terminated at its input side and another set of concurrent courses are started at its output side.

When a node L-fires, it propagates the omission of a whole substructure to which the node belongs.

In case we associate an atomic action (transformation of a variable state) with a &-node, an H-firing <L-firing> signals the execution <omission> of that action.

If a ∇-node has more than one input arc marked with an H-token, then this indicates that more than one of a set of alternative courses have been executed. Dually, if some input arc of a &-node carries an H-token and some arc carries an L-token, then this indicates that some but definitely not all courses of a set of concurrent courses have been accomplished. We have not provided special rules for dealing with these contradictory (and undesirable) situations. Instead we will look for ways of constructing marked bp graphs in which such situations can never occur.

We note that the notion of definite omission of actions which is modelled by the L-tokens comes in handy for determining one kind of bad control flow, in which a &-node gets into dead-lock. More importantly, it is through this second type of tokens, we hook up with the theory of live and safe marked graphs. How crucial this is will become clear by the way in which we exploit the results on marked graphs to build up our theory. Once a marked bp graph with the desired behaviour has been obtained however, we can if we wish to, discard the L-tokens. This is more or less what we do in section 7.

Let BG be a bp graph, w a node and $M = (M_H, M_L)$ a marking of BG. If w fires at M and leads to the marking $M' = (M'_H, M'_L)$, we denote this by M[w>M'. The two state spaces of interest associated with a marking are given in

<u>Definition 2.4</u> Let BG be a bp graph and M a marking of BG.

1) The <u>forward marking class</u> of M is denoted as [M> and is the smallest set of markings of BG given by

   a) $M \in$ [M>;

   b) if $M' \in$ [M> and for some node w, M'[v>M", then $M'' \in$ [M>.

2) The <u>full marking class</u> of M is denoted as [M] and is the smallest set of markings of BG given by

   a) $M \in [M]$;

   b) if $M' \in [M]$ and for some node w, M'[w>M" or M"[w>M', then $M'' \in [M]$. ∎

Finally, we can identify the objects of study of this paper.

<u>Definition 2.5</u> A <u>bipolar synchronization scheme</u> (bp scheme) is a 6-tuple $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$ where $BG = (V_\nabla, V_\&, A; Q, Z)$ is a <u>strongly connected</u> bp graph and M^0 is a marking of BG called the <u>initial marking</u> of BP. ∎

The notion of good behaviour is given in

Definition 2.6 Let $BP = (V_\nabla, V_\&, A;Q,Z;M^0)$ be a bp scheme. BP is well behaved iff for all forward reachable markings $M \in [M_0>$, there is, for all nodes w, a marking $M' \in [M>$ such that w may H-fire at M'.

∎

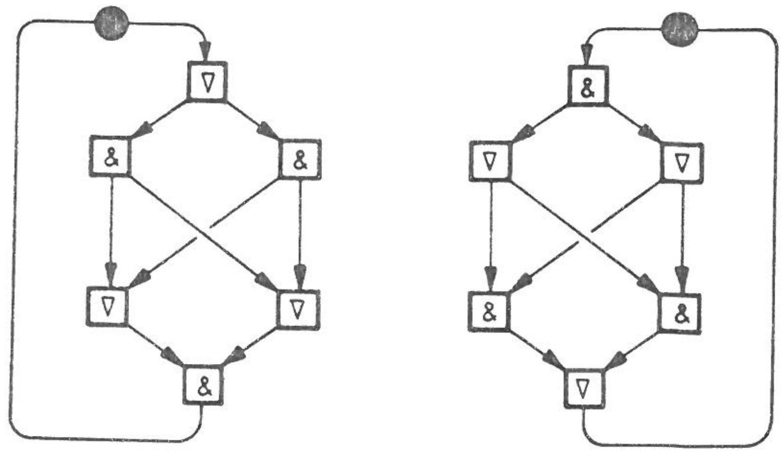
The bp scheme shown in fig. 2.3 is well behaved. Also, the bp scheme shown in fig. 2.5(a) is well behaved but not the one shown in fig. 2.5(b) (compare with fig. 2.2).

Fig. 2.5

Let $BG = (V_\nabla, V_\&, A;Q,Z)$ be a bp graph and M a marking of BG. Then BG is said to be well behaved at M iff the bp scheme $(V_\nabla, V_\&, A;Q,Z,M)$ is well behaved. We shall now work out an equivalent formulation of definition 2.6. This will reveal that our notion of good behaviour is intimately tied to the notion of deadlock. Before doing so it is necessary to introduce some notations that will be used throughout the paper.

Let $BP = (V_\nabla, V_\&, A;Q,Z,M^0)$ be a bp scheme. Then $BG = (V_\nabla, V_\&, A;Q,Z)$ is the supporting bp graph, $\widetilde{BG} = (V,A;Q,Z)$, where $V = V_\nabla \cup V_\&$, the underlying digraph, and $\widetilde{BP} = (V,A;Q,Z,\widetilde{M}^0)$, where $\widetilde{M}^0 = M_H^0 \cup M_L^0$, the underlying marked graph. The terminology concerning token loads on paths, basic circuits etc. that were introduced in section 1 are carried over to bp schemes, via their underlying marked graphs, in the obvious way. In particular, the strict partial order $<_M$ at a marking is carried over to bp schemes, too. Finally, the notion of a firing sequence is extended to bp schemes as follows:

Let BP be the generic bp scheme and $M^1 \in [M^0>$. Let $\sigma = w_1 \ldots w_n \in V^*$. Then σ is a firing sequence of BP at M^1 iff there exists a sequence of markings $M^2, \ldots, M^{n+1}$ such that for $1 \leq i \leq n$, $M^i[w_i>M^{i+1}$. The change from M^1 to M^{n+1} through σ is denoted as $M^1[\sigma>M^{n+1}$. Note that, unlike in the case of marked graphs, the resulting marking M^{n+1} is not uniquely determined by



(a) (b)

Fig. 2.5

M^1 and σ. This is due to the choice associated with the firing of a ∇-node. More precisely,

Proposition 2.1 Let BG be a bp graph, M a marking and w a node of BG. If $M[w>M'$, $M[w>M''$ and $M' \neq M''$, then w is a ∇-node, $|w^\bullet| > 1$ and w is enabled to H-fire at M.

Proof Follows easily from the firing rules. ∎

Two other useful observations are:

Proposition 2.2 Let BP be a bp scheme, $M \in [M^0>$ a marking of BP and $\sigma \in V^*$ a firing sequence of BP at M. Then σ is a firing sequence of the underlying marked graph $\widetilde{BP}$ at $\widetilde{M}$. Moreover, if $M[\sigma>M'$ in BP and $\widetilde{M}[\sigma>\hat{M}$ in $\widetilde{BP}$, then $\hat{M} = \widetilde{M}' = (M'_H \cup M'_L)$.

Proof Once again, follows from the firing rules for bp schemes and marked graphs. ∎

Proposition 2.3 Let BP be a bp scheme, $M \in [M^0>$ a marking of BP and $\sigma \in V^*$ a firing sequence of $\widetilde{BP}$ at $\widetilde{M}$. If there is no $M' \in [M^0>$ and no node w such that w is in deadlock at M', then σ is a firing sequence of BP at M.

Proof The result is again an immediate consequence of the firing rules. ∎

We shall encounter the last proviso of proposition 2.3 often in the sequel. So let us give it a name.

Definition 2.7 Let BP be the generic bp scheme. BP is said to be deadlock-free iff there is no marking $M \in [M_0>$ and no node w such that w is in deadlock at M. ∎

Our next result shows when, and how, in a deadlock-free bp scheme, a node can be enabled to H-fire and when to L-fire. In its proof, we use notations and results on marked graphs presented in the previous section.

Proposition 2.4   Let BP, the generic bp scheme, be deadlock-free, w a node and $M \in [M_0>$ a marking of BP. Let $\hat{A} = \{a \in \tilde{M} \mid Z(a) = w \text{ or } Z(a) <_{\tilde{M}} w\}$ and σ, a minimal w-enabling sequence of $\widetilde{BP}$ at $\tilde{M}$.

1) If $\hat{A} \cap M_H \neq \emptyset$, then there exists $M' \in [M>$ such that $M[\sigma>M'$ and w is enabled to H-fire at M'.

2) If $\hat{A} \cap M_H = \emptyset$, then for every marking $M' \in [M>$ with $M[\sigma>M'$, w is enabled to L-fire at M'.

Proof   Since BP is deadlock-free, σ is also a firing sequence of BP at M (proposition 2.3).

1) Let $b \in \hat{A} \cap M_H$, and $w' = Z(b)$. If $w' = w$, we are done because in this case w' is enabled to H-fire at all markings M' with $M[\sigma>M'$.
If $w' \neq w$, there exists a token-free path Π leading from w' to w at M. All nodes touched by Π, except w, occur in σ, and they occur in σ in the same order in which they are traversed by Π. Now during the firing of σ in BP, the H-token carried by b at M can be steered to flow along Π until it reaches w such that w is enabled to H-fire when the w-enabling firing sequence σ is completed. This is because, regardless of w' being a &-node or ∇-node, there is a firing of w' such that afterwards the output arc of w' belonging to Π (there is only one since the token-free path Π must be cycle-free) carries an H-token. So when the successor of w' on Π is to fire in σ, it is to H-fire. And again, there is a firing putting an H-token on its output arc belonging to Π. And so on, until finally w is enabled to H-fire at a marking $M' \in [M>$ ($M[\sigma>M'$).

2) Let $\hat{A} \cap M_H = \emptyset$ and $\sigma = w_1 \ldots w_n$. Then w_1 is enabled to L-fire at M because $^\bullet w_1 \subseteq \hat{A}$. Let $M[w_1>M^1$ and $\hat{A}^1 = \{a \in \tilde{M}^1 \mid Z(a) = w \text{ or } Z(a) <_{M^1} w\}$. Then it is easy to verify that $M_H^1 \cap \hat{A}^1 = \emptyset$, and $\sigma_1 = w_2 \ldots w_n$ is a minimal w-enabling sequence at $\tilde{M}^1$ in $\widetilde{BP}$ The required result is obtained by induction on $|\sigma|$.

∎

At long last we can state an equivalent and perhaps more illuminating version of well behavedness.

Theorem 2.5 A bp scheme BP is well behaved iff

a) it is deadlock-free, and
b) $M_H^0 \neq \emptyset$.

Proof

$\Rightarrow$: Assume that, at some $M \in [M^0>$, some node w is in deadlock. Then due to the firing rules, there is no $M' \in [M>$ at which w is enabled to fire let alone H-fire. This contradicts the well behavedness of BP.

Now assume that $M_H^0 = \emptyset$. Then due to the firing rules, for all $M \in [M^0>$, $M_H = \emptyset$. So for no node w there is marking in $[M^0>$ at which w is enabled to H-fire. This once again contradicts the well behavedness of BP.

$\Leftarrow$: Let $M \in [M^0>$ and $w \in V$. Since $M_H^0 \neq \emptyset$, $M_H \neq \emptyset$. Let x be a node with ${}^\bullet x \cap M_H \neq \emptyset$. There is a firing sequence σ not containing x leading in $\widetilde{BP}$ from $\widetilde{M}$ to an x-extremal marking $\widetilde{M}'$. Since BP is deadlock-free, σ is also a firing sequence of BP at M with $M[\sigma>M'$ for some M' (with $\widetilde{M}' = M_H' \cup M_L'$). Since x is not contained in σ, ${}^\bullet x \cap M_H' \neq \emptyset$. Let $\hat{A} = \{a \in \widetilde{M}' \mid Z(a) = w$ or $Z(a) <_{\widetilde{M}'} w\}$. Then ${}^\bullet x \subseteq \hat{A}$ because $\widetilde{M}'$ is x-extremal and thus $\hat{A} \cap M_H' \neq \emptyset$. So proposition 2.4 can be applied at M' to derive a marking $M'' \in [M'>$, at which w is enabled to H-fire. Clearly $M'' \in [M>$. ∎

The last result of this section characterizes a typical situation under which a ∇-node may get into deadlock.

Proposition 2.6 Let BP be the generic bp scheme and $M \in [M^0>$. Let $\Pi = a_0a_1...a_m$ and $\Pi' = b_0b_1...b_n$ be two acyclic paths such that (see fig. 2.6)

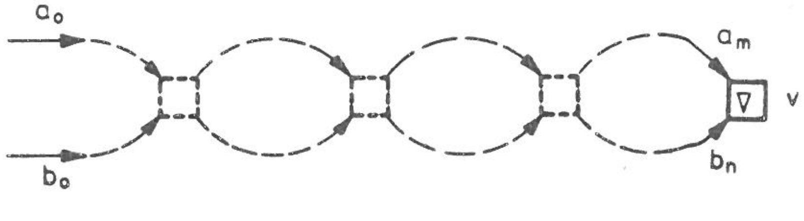


Fig. 2.6

a) Π and Π' are disjoint: for $0 \leq i \leq m$ and $0 \leq j \leq n$, $a_i \neq b_j$.

b) $Z(a_m) = Z(b_n) = v \in V_\nabla$.

c) $a_0, b_0 \in M_H$ and $M(\Pi) = M(\Pi') = 1$.

Then BP is not well behaved.

Fig. 2.6

Proof In the underlying marked graph $\widetilde{BP}$, there is at $\widetilde{M}$ a minimal v-enabling sequence $\sigma = w_0 \ldots w_p$. If σ is not a firing sequence of BP at M, then BP is not deadlock-free (proposition 2.3) and we are done. So let us exercise σ in BP starting from M. The two H-tokens can be steered along the two disjoint paths until they meet at the first ∇-node touched by both paths, and there is at least one such ∇-node, namely v. This ∇-node, however, is then in deadlock since more than one input arc of this node will carry an H-token. So BP is not well behaved.

∎

3. THE FC REPRESENTATION OF A BP SCHEME

In this section we develop a close connection between bp schemes and a class of Petri nets called (marked) free choice nets. Then using this connection, we exploit the theory of free choice nets to derive an important necessary condition for a bp scheme to be well behaved. This condition will help us establish the results of the next section. It also forms the basis of the computational interpretation of bp schemes worked out in section 7. To start with we introduce some terminology concerning nets.

3.1 Nets and Marked Nets

Definition 3.1 A directed net is a triple $N = (S,T;F)$ with

- $S \cap T = \emptyset$ and $S \cup T \neq \emptyset$;
- $F \subseteq (S \times T) \cup (T \times S)$ such that $dom(F) \cup rng(F) = S \cup T$. ∎

S is the set of S-elements, T is the set of T-elements and F is the flow relation; $X = S \cup T$ is the set of elements of N. In diagrams, the S-elements are drawn as circles and the T-elements are boxes. If $(x,y) \in F$ then there will be a directed line from x to y. Since we shall be dealing with only directed nets, from now on we will just say nets instead of directed nets.

A very useful notation is: Let $N = (S,T;F)$ be a net and $x \in X = S \cup T$. Then

$^{\bullet}x = \{y \in X \mid (y,x) \in F\}$ is the pre-set of x, and

$x^{\bullet} = \{y \in X \mid (x,y) \in F\}$ is the post-set of x.

For $Y \subseteq X$ we extend this to $^{\bullet}Y = \bigcup_{x \in Y} {^{\bullet}x}$ and $Y^{\bullet} = \bigcup_{x \in Y} x^{\bullet}$.

Definition 3.2 Let $N = (S,T;F)$ be a net. Then,

(1) N is an S-graph iff for all $t \in T$, $|^{\bullet}t|, |t^{\bullet}| \leq 1$;

(2) N is a T-graph iff for all $s \in S$, $|^{\bullet}s|, |s^{\bullet}| \leq 1$;

(3) N is a free-choice net iff for all $s \in S$, $|s^{\bullet}| > 1 \Rightarrow {^{\bullet}(s^{\bullet})} = \{s\}$. ∎

In fig. 3.1 we have shown the constraints placed on the structure of nets through conditions (1), (2), and (3) of the definition above.

Fig. 3.1

An S-graph represents the structure of a conventional sequential state machine. A T-graph is the dual of an S-graph. The structure of a marked graph can be represented as T-graph. (This will become clear toward the end of this section.) Thus marked graphs and sequential state machines are in some sense duals of each other. A free choice net is their common generalisation. This is brought out in the structural theory of live and safe marked free choice nets developed by Hack [8]. For now we merely observe that if in a free choice net N, two different T-elements t_1, t_2 share a pre-element $s(s \in {}^{\bullet}t_1 \cap {}^{\bullet}t_2)$, then ${}^{\bullet}t_1 = {}^{\bullet}t_2 = \{s\}$.

We shall now generalise the notion of a marked graph and the related notions to marked nets. Let $N = (S,T;F)$ be a net. A marking of N is a function $\mu: S \to \mathbb{N}$. In diagrams, we indicate a marking μ by placing on each S-element s, $\mu(s)$ tokens. A net together with a marking is shown in fig. 3.2.

Fig. 3.2

Definition 3.3 Let $N = (S,T;F)$ be a net, μ a marking of N and $t \in T$. Then t is enabled (to fire) iff for every $s \in {}^{\bullet}t$, $\mu(s)>0$. When t fires, a new marking μ' is obtained such that for all $s \in S$,

$$\mu'(s) = \begin{cases} \mu(s)-1, & \text{if } s \in ({}^{\bullet}t - t^{\bullet}); \\ \mu(s)+1, & \text{if } s \in (t^{\bullet} - {}^{\bullet}t); \\ \mu(s), & \text{otherwise.} \end{cases}$$ ∎

The transformation of μ into μ' through a firing of t will be denoted by $\mu[t>\mu'$. If μ_1 is a marking of N and $\sigma = t_1 \dots t_n \in T^*$, then σ is said to be a firing sequence at μ, iff there exist markings $\mu_2, \dots, \mu_{n+1}$ such that for $1 \leq i \leq n$, $\mu_i[t_i>\mu_{i+1}$. The transformation of μ_1 into μ_{n+1} through the firing sequence σ is denoted by $\mu_1[\sigma>\mu_{n+1}$. By convention for every marking μ of N, $\mu[\lambda>\mu$.

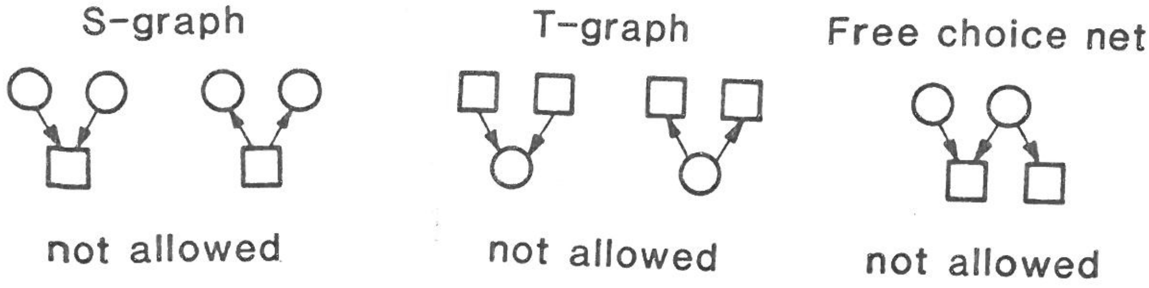

S-graph
not allowed
T-graph
not allowed
Free choice net
not allowed

Fig. 3.1

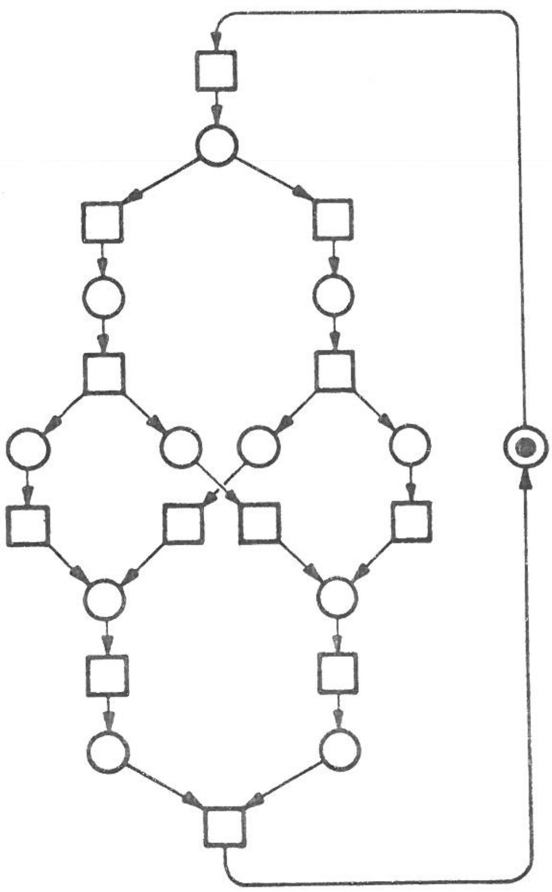


Fig. 3.2

Definition 3.4 Let $N = (S,T;F)$ be a net and μ a marking of N. Then $[\mu>$, the forward marking class of N defined by μ, is the smallest set of markings of N which satisfies:

- $\mu \in [\mu>$.
- If $\mu' \in [\mu>$ and for some $t \in T$ and some marking μ'' of N we have $\mu' \in t > \mu''$, then $\mu'' \in [\mu>$. ∎

Definition 3.5 A marked net is a quadruple $MN = (S,T;F,\mu_0)$ where $N = (S,T;F)$ is the underlying net of MN and μ_0 is a marking of N called the initial marking of MN. ∎

Fig. 3.2 can now be viewed as an example of a marked net whose underlying net is a free choice net. As in the case of marked graphs, two important behavioural properties of a marked net are liveness and safety.

Definition 3.6 Let $MN = (S,T;F,\mu_0)$ be a marked net. Then,

(1) MN is said to be live iff for every marking $\mu \in [\mu_0>$ and every $t \in T$, there is a marking $\mu' \in [\mu>$ at which t may fire.

(2) MN is said to be safe iff for every marking $\mu \in [\mu_0>$ and every $s \in S$, $\mu(s) \leq 1$. ∎

The marked net shown in fig. 3.2 is live and safe. If $N = (S,T;F)$ is a net and μ is a marking of N, N is said to be live <safe> at μ iff the marked net $(S,T;F,\mu)$ is live <safe>. This concludes our rapid introduction to marked nets. For more details, the interested reader is referred to [11].

3.2 Bp Schemes and Marked Free Choice Nets

The aim here is to represent a bp scheme as a marked net. More precisely, we wish to use a marked free choice net to simulate the flow of H-tokens in a well behaved bp scheme. The basic idea behind this representation is shown informally in fig. 3.3.

Fig. 3.3

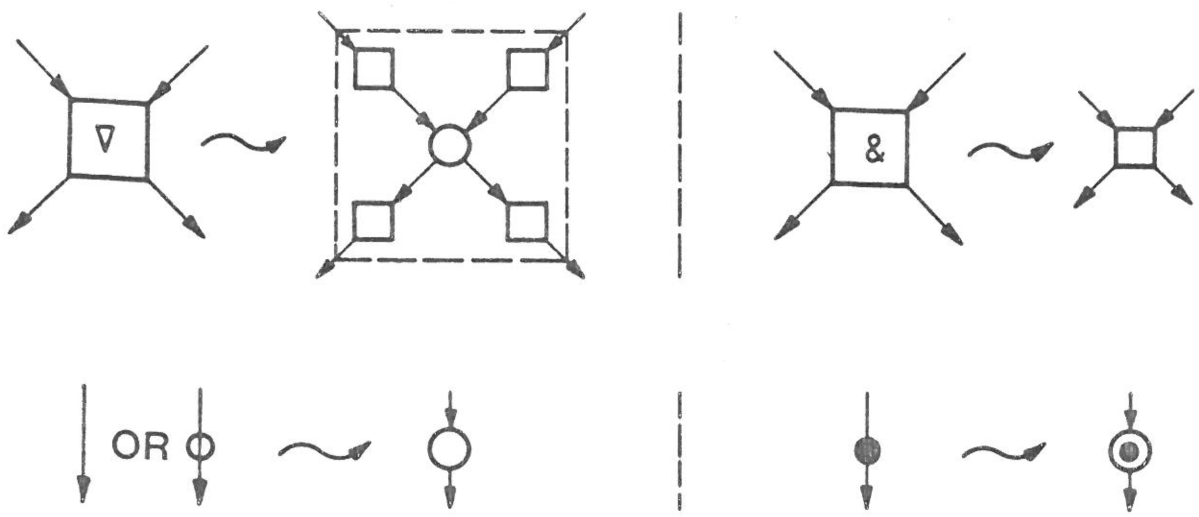


Fig. 3.3

To formalise this idea we need some additional notations. Let BG = $(V_\nabla, V_\&, A; Q, Z)$ be a bp graph. Viewing Q and Z as binary relations which are subsets of $A \times V$, we now define:

$$Q_\nabla = \{(v,a) \in Q^{-1} \mid v \in V_\nabla\}$$
$$Q_\& = \{(u,a) \in Q^{-1} \mid u \in V_\&\}$$
$$Z_\nabla = \{(a,v) \in Z \mid v \in V_\nabla\}$$
$$Z_\& = \{(a,u) \in Z \mid u \in V_\&\}$$

Definition 3.7   Let BG be a bp graph and N = (S,T;F) a net. N is called an fc representation of BG iff there exists a bijection $f: A \cup V_\nabla \cup V_\& \cup Q_\nabla \cup Z_\nabla \to S \cup T$ such that, with $\bar{x} = f(x)$ and $\overline{xy} = f((x,y))$,

(1) $f(A \cup V_\nabla) = S$ and $f(V_\& \cup Q_\nabla \cup Z_\nabla) = T$
(2) $(v,a) \in Q_\nabla \Leftrightarrow (\bar{v}, \overline{va}) \in F$ and $(\overline{va}, \bar{a}) \in F$
(3) $(a,v) \in Z_\nabla \Leftrightarrow (\bar{a}, \overline{av}) \in F$ and $(\overline{av}, \bar{v}) \in F$
(4) $(u,a) \in Q_\& \Leftrightarrow (\bar{u}, \bar{a}) \in F$
(5) $(a,u) \in Z_\& \Leftrightarrow (\bar{a}, \bar{u}) \in F$ ∎

The terminology "fc representation" is suggested by

Proposition 3.1   Let BG = $(V_\nabla, V_\&, A; Q, Z)$ be a bp graph and N = (S,T;F) be an fc representation of BG (by virtue of the bijection f ($\bar{\ }$)). Then N is a free choice net.

Proof   From def. 3.7 it is easy to verify that for every arc $a \in A$, $|\bar{a}^\bullet| = 1$. Thus if $s \in S$ with $|s^\bullet| > 1$, $s = \bar{v}$ for some $v \in V_\nabla$. But for every $t \in s^\bullet$, $t = \overline{va}$ for some $a \in v^\bullet$ ($(v,a) \in Q_\nabla$); since Q is a function, we have once again from def. 3.7 that ${}^\bullet t = \{s\}$. ∎

As in the proof above we shall let the $^\bullet$-notation do double duty; from the context it should be clear whether we are dealing with a bp graph or a net.

The notion of fc representation is extended to bp schemes as follows:

Definition 3.8 Let $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$ be a bp scheme and $MN = (S,T;F,\mu_0)$ a marked net. Then MN is said to be an fc representation of BP iff

- $(S,T;F)$ is an fc representation of BG, the supporting bp graph of BP, by virtue of some bijection f;
- for all $s \in S$, $\mu_0(s) = \begin{cases} 1, \text{ if } f^{-1}(s) \in M^0_H \\ 0, \text{ otherwise} \end{cases}$

An fc representation of the bp scheme of fig. 2.5(a) is shown in fig. 3.2.

3.3 A Necessary Condition for Well Behavedness

Through the remaining part of this section, we shall employ the $\bar{}$ notation to denote 'the' - up to isomorphy unique - fc representation of a bp graph. The main result we are after is that the fc representation of a well behaved bp scheme is live and safe. We first prove safety. To do so, we need two new notions concerning marked nets.

Let $MN = (S,T;F,\mu_0)$ be a marked net and $\mu \in [\mu_0\rangle$. μ is said to be a safe marking, in the forward marking class of MN, iff for every S-element s, $M(s) \leq 1$. Clearly MN is safe iff every $\mu \in [\mu_0\rangle$ is a safe marking. Now let $\mu_1 \in [\mu_0\rangle$ and $\sigma = t_1 \ldots t_n \in T^*$ be a firing sequence at μ_1 such that for $1 \leq i \leq n$, $\mu_i [t_i\rangle \mu_{i+1}$. Then σ is said to be a safe firing sequence at μ_1 iff for all $1 \leq i \leq n+1$, μ_i is a safe marking.

In order to show that the fc representation of a well behaved bp scheme is safe, it is convenient to first prove that the bp scheme can simulate the flow of tokens in its fc representation. The simulation idea is somewhat delicate and is contained in the proof of the following technical result.

<u>Lemma 3.2</u> Let BP = $(V_\nabla, V_\&, A; Q, Z, M^0)$ be a well behaved bp scheme and MN = $(S,T;F,\mu_0)$ its fc representation. Let $\sigma \in T^*$ be a safe firing sequence of MN at μ_0 with $\mu_0[\sigma>\mu$. Then,

(A) $\forall v \in V_\nabla$: $\mu(\bar{v}) + \sum_{a \in {}^\bullet v} \mu(\bar{a}) \leq 1$.

Moreover there exists, in BP, a marking $M \in [M^0>$ such that

(B) $\forall a \in A$: if $\mu(\bar{a})=1$ then $a \in M_H$;

(C) $\forall a \in A$: if $a \in M_H$ then either $\mu(\bar{a})=1$ or $[Z(a)=v \in V_\nabla$ and $\mu(\bar{v})=1]$;

(D) $\forall v \in V_\nabla$: if $\mu(\bar{v})=1$ then v is H-firable at M.

<u>Proof</u> The proof is by induction on the length of σ, $k = |\sigma|$. Since μ_0 is by definition (3.8) a safe marking in $[\mu_0>$ and $\mu_0[\lambda>\mu_0$, we can start with

<u>k=0</u>: Let $v \in V_\nabla$. Then once again by definition, $\mu_0(\bar{v})=0$. Suppose that $\sum_{a \in {}^\bullet v} \mu_0(\bar{a})>1$. Then there must be a $a_1, a_2 \in {}^\bullet v$ with $a_1 \neq a_2$ and $a_1, a_2 \in M_H^0$. This, however, implies that v is in deadlock at M^0, contradicting the well behavedness of BP. So

(A) $\mu_0(\bar{v}) + \sum_{a \in {}^\bullet v} \mu_0(\bar{a}) \leq 1$.

(B), (C), and (D) are trivially satisfied for $M^0 \in [M^0>$.

<u>k>0</u>: Let $\sigma = \sigma_1 t$ with $t \in T$. Then $\sigma_1 \in T^*$ and $|\sigma_1| = k-1 \geq 0$. Since σ is a safe firing sequence at μ_0, so is σ_1. Let $\mu_0[\sigma_1>\mu_1[t>\mu$. Then by the induction hypothesis, μ_1 satisfies part (A) and there is a marking $M^1 \in [M^0>$ which satisfies (B), (C) and (D) with respect to μ_1. To establish (A) through (D) for μ, we need to consider three cases according to the origin of t.

<u>Case 1</u> $t = \overline{va}$ with $(v,a) \in Q_\nabla$.
Then $v \in V_\nabla$, $a \in v^\bullet$, ${}^\bullet t = \{\bar{v}\}$ and $t^\bullet = \{\bar{a}\}$.

(A): Since μ_1, μ are safe markings and $\mu_1[t>\mu$, we can conclude that $\mu_1(\bar{v}) = \mu(\bar{a}) = 1$ and $\mu_1(\bar{a}) = \mu(\bar{v}) = 0$. Since $\mu_1(\bar{v}) = 1$ and μ_1 satisfies (A), $\sum_{b \in {}^\bullet v} \mu_1(\bar{b}) = 0$. And $t^\bullet = \{\bar{a}\}$, so even if $a \in {}^\bullet v$, we must have $\sum_{b \in {}^\bullet v} \mu(\bar{b}) \leq 1$. With $\mu(\bar{v}) = 0$, as already seen, μ indeed satisfies (A) for v.

So assume that $v' \in V_\nabla$ with $v' \neq v$. Once again we have to prove that $\mu(\bar{v}') + \sum_{b \in {}^\bullet v'} \mu(\bar{b}) \leq 1$.

If $a \notin {}^\bullet v'$, we are done because then $\mu(v') + \sum_{b \in {}^\bullet v'} \mu(\bar{b}) = \mu_1(v') + \sum_{b \in {}^\bullet v'} \mu_1(\bar{b})$ and μ_1 satisfies (A).

So assume that $a \in {}^\bullet v'$. Again since μ is a safe marking and $t^\bullet = \{\bar{a}\}$, we have $\mu_1(\bar{a})=0$ and $\mu(\bar{a})=1$. The situation is shown in fig. 3.4.

Fig. 3.4

We claim that $\mu_1(\bar{v}')=0$. Otherwise both v and v' are H-fireable at M^1 (part (D)). But this would imply that the underlying marked graph of BP is not safe contradicting the definition of a bp scheme.

So at this stage we have $\mu(\bar{a})=1$ and $\mu(\bar{v}')=0$. Now consider some $b \in {}^\bullet v'$, $b \neq a$. Suppose that $\mu(\bar{b})=1$. Then $\mu_1(\bar{b})=1$ also. So by part (B) for μ_1 we have $b \in M_H^1$, and $v \in V_\nabla$ is H-firable at M^1 (part (D)). Consequently there is a marking M^2 in BP with $M^1[v>M^2$ such that $a \in M_H^2$. Clearly $b \in M_H^2$ also because $b \in {}^\bullet v'$ and $v' \neq v$. This means that v' is in deadlock at M^2, contradicting the well behavedness of BP. Thus for every $b \in {}^\bullet v'$, $\mu_1(\bar{b}) = \mu(\bar{b}) = 0$. This at once lets us conclude that $\mu(\bar{v}') + \sum_{b \in {}^\bullet v'} \mu(\bar{b}) \leq 1$. So (A) holds for μ.

To show (B), (C) and (D), recall that v is H-firable at M^1 because $\mu_1(\bar{v})=1$. Let M be such that $M^1[v>M$ with $a \in M_H$. Then it is routine to verify that M satisfies (B), (C) and (D) w.r.t. μ.

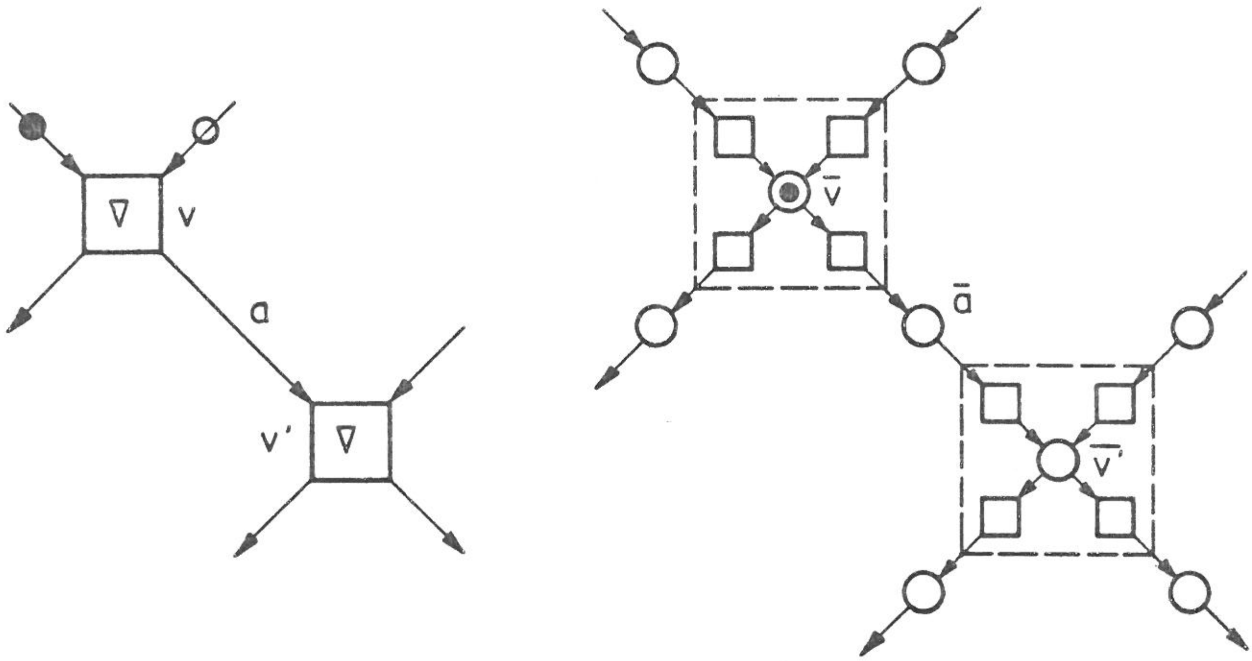


Fig. 3.4

Case 2 $t = \overline{av}$ with $(a,v) \in Z_\nabla$.
Then $v \in V_\nabla$, $a \in {}^\bullet v$, ${}^\bullet t = \{\bar{a}\}$ and $t^\bullet = \{\bar{v}\}$.
(A): Both μ_1 and μ are safe markings, so $\mu_1(\bar{a}) = \mu(\bar{v}) = 1$ and $\mu_1(\bar{v}) = \mu(\bar{a}) = 0$. Clearly we have for every ∇-node v', $\mu(\bar{v}') + \sum_{b \in {}^\bullet v'} \mu(\bar{b}) = \mu_1(\bar{v}') + \sum_{b \in {}^\bullet v'} \mu_1(\bar{b})$, consequently μ satisfies (A) because μ_1 does so.
(B), (C), (D): Let $A^1 = \{b \in A \mid Z(b) = v$ or $Z(b) <_{M^1} v\}$ (recall that M^1 is the marking in BP satisfying (B)-(D) w.r.t. μ_1).
Since $Z(a) = v$ we have $a \in A^1$.

Claim $A^1 \cap M_H^1 = \{a\}$.

Suppose that $b \in A^1 \cap M_H^1$ with $b \neq a$. If $Z(b) = v$, v is in deadlock at M^1 which is a contradiction.

So there is at M^1 a token-free path Π from $Z(b)$ to v with $b \in M_H^1$. We also have the path of length 1 consisting of the arc a leading from $Q(a)$ to v. Set $\Pi_1 = b\Pi$ and $\Pi_2 = a$. Then from proposition 2.6, it follows at once that BP is not well behaved which is a contradiction. This proves the claim.

Now let $\tau \in V^*$ be a minimal v-enabling sequence at M^1 and $M^1[\tau > M$. Because $A^1 \cap M_H^1 = \{a\}$, it is easy to see that in going from M^1 to M via τ, every node that appears in τ will only L-fire (the idea is contained in the proof of the second part of proposition 2.4). Thus M is the unique marking in $[M^1>$ such that $M^1[\tau>M$. More importantly, for every arc b we have $b \in M_H$ iff $b \in M_H^1$. It is straightforward to verify that M satisfies (B), (C) and (D) w.r.t. μ.

Case 3 $t = \bar{u}$ for $u \in V_\&$.
The proof is similar to and simpler than the proof of the first case and hence we omit it. ∎

Theorem 3.3 Let $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$ be a bp scheme and $MN = (S, T; F, \mu_0)$ its fc representation. Then MN is safe.

Proof Let $\sigma \in T^*$ be a firing sequence at μ_0. We shall show that σ is a safe firing sequence at μ_0. The proof is by induction on the length of σ, $k = |\sigma|$.

<u>k=0</u>: $\mu_0[\lambda>\mu_0$ and μ_0 is a safe marking by definition.

<u>k>0</u>: Let $\sigma = \sigma_1 t$ with $t \in T$. Then $\sigma_1 \in T^*$ and $|\sigma_1| = k-1 \geq 0$. Let $\mu_0[\sigma_1>\mu_1$. By induction hypothesis, σ_1 is a safe firing sequence at μ_0 so that μ_1 satisfies part (A) of lemma 3.2. Moreover we can find $M^1 \in [M^0>$ which satisfies parts (B), (C) and (D) of the lemma w.r.t. μ_1. Once again there are three cases to consider:

<u>Case 1</u> $t = \overline{va}$ with $(v,a) \in Q_\nabla$.
Then $v \in V_\nabla$, $a \in v^\bullet$, $t^\bullet = \{\bar{a}\}$, ${}^\bullet t = \{\bar{v}\}$ and $\mu_1(\bar{v}) = 1$.

First suppose that $a \in {}^\bullet v$ too (see fig. 3.5).

Fig. 3.5

Since $\mu_1(v) + \sum_{b \in {}^\bullet v} \mu_1(\bar{b}) \leq 1$, we have $\mu_1(\bar{a}) = 0$. But $t^\bullet = \{\bar{a}\}$. Hence $\sigma_1 t = \sigma$ is a safe firing sequence at μ_0.

Now assume that $a \notin {}^\bullet v$. If $\mu_1(\bar{a}) = 1$ then $\bar{a} \in M^1_H$. But v is H-firable at M^1 implying that ${}^\bullet v \subseteq \tilde{M}^1$ $(= M^1_H \cup M^1_L)$. Consequently the underlying marked graph BP is not safe which is a contradiction. Thus $\mu_1(\bar{a}) = 0$ which lets us conclude that $\sigma_1 t = \sigma$ is a safe firing sequence at μ_0.

<u>Case 2</u> $t = \overline{av}$ with $(a,v) \in Z_\nabla$.
Then $v \in V_\nabla$, $a \in {}^\bullet v$, ${}^\bullet t = \{\bar{a}\}$, $t^\bullet = \{\bar{v}\}$.

Now t is firable at μ_1, a safe marking, so that $\mu_1(\bar{a}) = 1$. By induction hypothesis, $\mu_1(\bar{v}) + \sum_{b \in {}^\bullet v} \mu_1(\bar{b}) \leq 1$. Thus $\mu_1(\bar{v}) = 0$. But $t^\bullet = \{\bar{v}\}$, hence $\sigma_1 t = \sigma$ is a safe firing sequence at μ_0.

<u>Case 3</u> $t = \bar{u}$ with $u \in V_\&$.

Once again we omit the proof because it is similar to and simpler than the proof of the first case. ∎

We now wish to prove that the fc representation of a well behaved bp scheme is live. To do so we shall first show that the fc representation can simulate the flow of H-tokens in the well behaved bp scheme. This is quite easy.

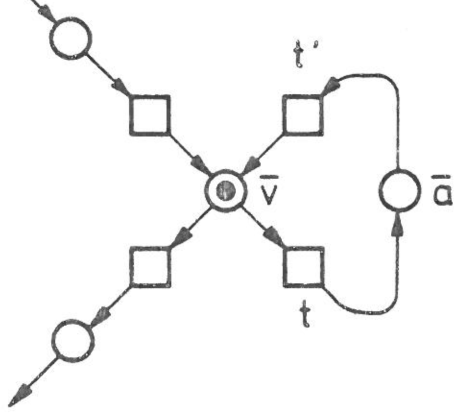


Fig. 3.5

<u>Lemma 3.4</u> Let $BP = (V_{\nabla}, V_{\&}, A; Q, Z, M^0)$ be a well behaved bp scheme and $MN = (S,T;F,\mu_0)$ its fc representation. Let $\sigma \in V^*$ be a firing sequence at M^0 with $M^0[\sigma>M$. Then there exists a $\mu \in [\mu_0>$ such that

(1) for all $a \in A$, $a \in M_H$ iff $\mu(\bar{a})=1$;

(2) for all $v \in V_{\nabla}$, $\mu(\bar{v})=0$.

<u>Proof</u> By induction on $k=|\sigma|$. ∎

<u>k=0:</u> Trivial.

<u>k>0:</u> Let $\sigma = \sigma_1 w$ with $w \in V$. Then $\sigma_1 \in V^*$ and $|\sigma_1| = k-1 \geq 0$. Let $M^0[\sigma_1>M^1$ and $M^1[w>M$. By induction hypothesis, we can find in MN a marking $\mu_1 \in [\mu_0>$ such that for all $a \in A$, $a \in M^1_H$ iff $\mu_1(\bar{a})=1$, and for all $v \in V_{\nabla}$, $\mu_1(\bar{v})=0$. Now consider three cases:

<u>Case 1</u> w L-fires at M^1 to lead to M.
Then $M_H = M^1_H$. Consequently, $\mu = \mu_1$ can serve as the required marking.

<u>Case 2</u> w is a ∇-node which H-fires at M^1 to yield M.
Let $a_1 \in {}^{\bullet}w$ and $a_2 \in w^{\bullet}$ such that $a_1 \in M^1_H$ and $a_2 \in M_H$.

First let us suppose that $a_2 \in M^1_H$ also. We claim that this is the case iff $a_1=a_2$. If $a_2 \notin {}^{\bullet}w$ then the underlying marked graph BP is not safe which is a contradiction. If $a_2 \in {}^{\bullet}w$ but $a_1 \neq a_2$ then w is in deadlock at M^1 which once again is a contradiction. And if it is the case $a_1=a_2 \in M^1_H$, then at μ_1 we can first let $t_1 = \overline{a_1 w}$ fire, followed by a firing of $t_2 = \overline{wa_1}$ to obtain the marking μ ($\mu_1[t_1,t_2>\mu$). Since MN is safe, $\mu(\overline{a_1})=1$ and $\mu(\bar{v})=0$. It is straightforward to check that $\mu \in [\mu_0>$ is the marking we are looking for w.r.t. $M \in [M^0>$.

In case $a_2 \notin M^1_H$, then at μ_1 we can fire $t_1 = \overline{a_1 w}$ followed by a firing of $t_2 = \overline{wa_2}$ to obtain the required marking μ.

Case 3 w is a &-node which H-fires at M^1.
Then $t = \bar{u}$ may fire at μ_1. Clearly μ with $\mu_1[t>\mu$ satisfies the requirements w.r.t. $M\in[M^0>$. ∎

At long last we are ready to prove the central result of this section:

Theorem 3.5 Let $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$ be a well behaved bp scheme and $MN = (S,T;F,\mu_0)$ its fc representation. Then MN is a live and safe marked net.

Proof From theorem 3.3 we know that MN is safe. So let $\mu_1\in[\mu_0>$ and $t\in T$. We shall show that for some $\mu\in[\mu_1>$, t may fire at μ.

To begin with we should like to go over from μ_1 to a marking μ_2 at which for every $v\in V_\nabla$, $\mu_2(\bar{v})=0$. Let $V_\nabla^1 = \{v\in V_\nabla \mid \mu_1(\bar{v})=1\}$. If $V_\nabla^1 = \emptyset$ we are done with $\mu_2 = \mu_1$. For $V_\nabla^1 = \{v_1,\ldots,v_n\}$, $n>0$, choose $a_1,\ldots,a_n$ such that for $1\leq i\leq n$, $a_i\in v_i^\bullet$. Set $t_i = \overline{v_i a_i}$ for $1\leq i\leq n$. Clearly each t_i may fire at μ_1 and for $1\leq i,j\leq n$, $i\neq j$ implies ${}^\bullet t_i \cap {}^\bullet t_j = \emptyset$. Consequently, $t_1 t_2 \ldots t_n = \sigma' \in T^*$ is a firing sequence at μ_1. Let $\mu_1[\sigma'>\mu_2$. Now MN is safe so that for $1\leq i\leq n$, $\mu_2(\bar{v}_i)=0$. Furthermore, $\{v\in V_\nabla \mid \mu_2(\bar{v})=1\} = \emptyset$.

Since $\mu_2\in[\mu_0>$ and MN is safe, there is a safe firing sequence σ such that $\mu_0[\sigma>\mu_2$. By lemma 3.2, there exists a marking $M^2\in[M^0>$ such that for all $a\in A$, $a\in M_H^2$ iff $\mu_2(\bar{a})=1$. (The 'only if' part follows from part (C) and the fact that $\mu_2(\bar{v})=0$ for every $v\in V_\nabla$.) The point is, $MN_2 = (S,T;F,\mu_2)$ is the fc representation of $BP^2 = (V_\nabla, V_\&, A; Q, Z, M^2)$.

We need to show that for some $\mu\in[\mu_2>$, t may fire at μ. Since BP^2 is well behaved, lemma 3.4 tells us what to do in the three cases we need to consider.

Case 1 $t = \overline{va}$ with $(v,a)\in Q_\nabla$.
Let $M^3\in[M^2>$ such that v may H-fire at M^3. Then by lemma 3.4, we can find a $\mu_3\in[\mu_2>$ such that for all $b\in A$, $\mu_3(\bar{b})=1$ iff $b\in M_H^3$.

Let b be the one arc with $b \in {}^{\bullet}v \cap M_H^3$. Then $\mu_3(\bar{b})=1$ and $t' = \overline{bv}$ may fire at μ_3 since ${}^{\bullet}t' = \{\bar{b}\}$. For $\mu_3[t'>\mu$, $\mu(\bar{v})=1$ so that t may fire at μ because ${}^{\bullet}t = \{\bar{v}\}$.

Case 2 $\quad t = \overline{av}$ with $(a,v) \in Z_\nabla$.
Since BP^2 is well behaved, there is a marking $M^3 \in [M^2>$ such that Q(a) may H-fire at M^3 in such a way that for the resulting marking $M \in [M^2>$, $a \in M_H$. By lemma 3.4, there is a marking $\mu \in [\mu_2>$ at which $\mu(\bar{a})=1$. Clearly t may fire at μ since ${}^{\bullet}t = \{\bar{a}\}$.

Case 3 $\quad t = \bar{u}$ with $u \in V_\&$.
There is a marking $M \in [M^2>$ at which u may H-fire, so that ${}^{\bullet}u \subseteq M_H$. By lemma 3.4, there is a marking $\mu \in [\mu_2>$ such that for each $b \in {}^{\bullet}u$, $\mu(\bar{u})=1$. Since ${}^{\bullet}t = \{\bar{b} \mid b \in {}^{\bullet}u\}$, t may fire at μ.

To conclude this proof we observe that in all three cases, $\mu \in [\mu_1>$ since $\mu \in [\mu_2>$ and $\mu_2 \in [\mu_1>$. ∎

The converse of theorem 3.5 is false.It will be convenient, however, to bring out this fact in a later subsection.

3.4 The Structural Components of a Well Behaved bp Scheme

Theorem 3.5 makes the rich and elegant theory of free choice nets accessible for our study of bp schemes. In this subsection, Hack's decomposition results for live and safe free choice nets [8] will serve as a basis for deriving quite analogous and very useful results concerning the structure of well behaved bp schemes. As before we have to start with introducing several notions.

Definition 3.9 $\quad$ Let $BG = (V_\nabla, V_\&; a; Q, Z)$ and $BG' = (V'_\nabla, V'_\&, A'; Q', Z')$ be bp graphs. Then BG' is said to be a (bp) subgraph of BG iff

- $V'_\nabla \subseteq V_\nabla$, $V'_\& \subseteq V_\&$
- $A' \subseteq \{b \in A \mid Q(b) \in V' \text{ and } Z(b) \in V'\}$ $(V' = V'_\nabla \cup V'_\&)$
- $Q' = Q \cap (A' \times V)$ and $Z' = Z \cap (A' \times V)$. ∎

In what follows, whenever we talk about a bp graph BG and one of its sub-graphs, the $^\bullet$-notation denoting the input and output arcs of a node will always refer to the incidence structure of the original bp graph BG.

<u>Definition 3.10</u> Let $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$ be a bp scheme and $BG' = (V'_\nabla, V'_\&, A'; Q', Z')$ be a bp subgraph of the supporting bp graph of BP. We call BG' a <u>∇-component</u> of BP iff

- $\forall v \in V'_\nabla : ({}^\bullet v \cup v^\bullet) \subseteq A'$
- $\forall u \in V'_\& : |{}^\bullet u \cap A'| = |u^\bullet \cap A'| = 1$
- BG' is strongly connected
- $|A' \cap M^0_H| = 1$ ∎

Every ∇-node of a ∇-component has the same set of input arcs and set of output arcs as in the given scheme. A &-node has exactly one input arc and one output arc. Moreover, a ∇-component is strongly connected and exactly one arc carries an H-token at M^0. The key feature of a ∇-component is that w.r.t. node firings in the original scheme, it can neither gain nor lose H-tokens. More precisely,

<u>Proposition 3.6</u> $BP = (V_\nabla, V_\&, A, Q, Z, M^0)$ be a bp scheme and BG' a ∇-component of BP. Then for every $M \in [M^0]$, BG' is a ∇-component of the bp scheme $(V_\nabla, V_\&, A; Q, Z, M)$.

<u>Proof</u> Follows easily from the definition. ∎

The dual notion is that of an &-component. Here however we cannot say anything definite about the submarking acquired by the component.

<u>Definition 3.11</u> Let $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$ be a bp scheme and $BG' = (V'_\nabla, V'_\&, A'; Q', Z')$ be a subgraph of the underlying bp graph of BP. We call BG' an <u>&-component</u> of BP iff

- $\forall u \in V'_{\&}$: ${}^{\bullet}u \cup u^{\bullet} \subseteq A'$
- $\forall v \in V'_{\nabla}$: $|{}^{\bullet}v \cap A'| = |v^{\bullet} \cap A'| = 1$
- BG' is strongly connected.

The bp scheme of fig. 2.5.a has two ∇-components and two &-components. They are shown, together with the corresponding submarkings in fig. 3.6.

Fig. 3.6

It turns out that every arc - and thus every node - of a well behaved bp scheme is contained in a ∇-component as well as an &-component. To derive this result we need to first develop the corresponding notions for marked nets.

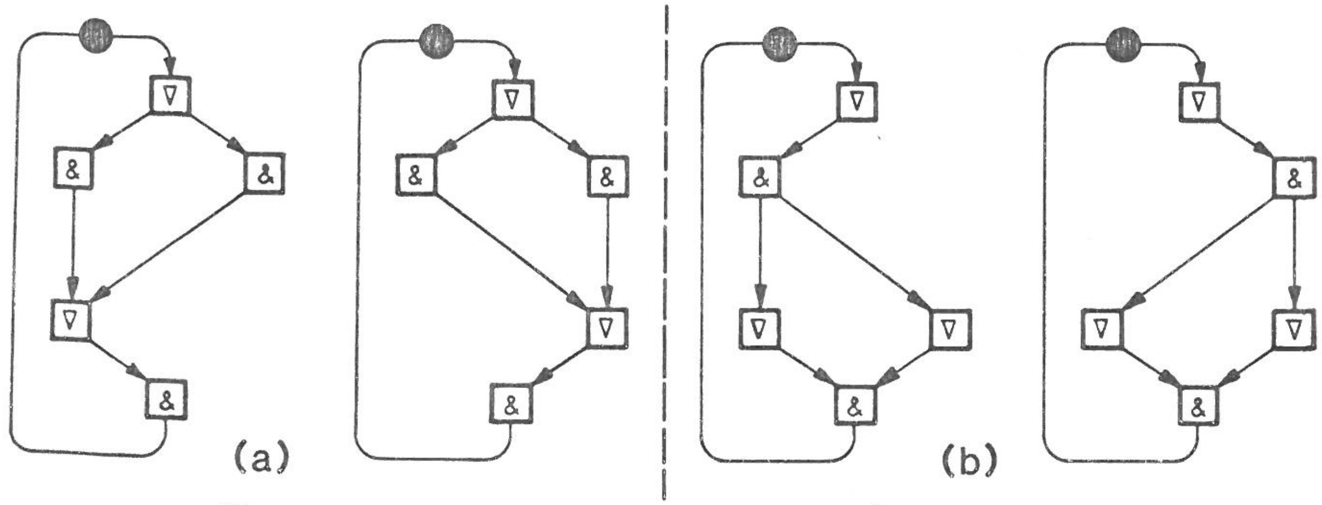
Definition 3.12 Let N = (S,T;F) and N' = (S',T';F') be nets. N' is a subnet of N iff

- $S' \subseteq S$ and $T' \subseteq T$
- $F' = F \cap ((S' \times T') \cup (T' \times S'))$ ∎

Once again, when using the $^{\bullet}$-notation in connection with a net and one of its subnets, we shall be referring to the incidence structure of the original net.

Definition 3.12 Let MN = $(S,T;F,\mu_0)$ be a marked net. Then a 1-token, strongly connected S-graph component (S-component for short) of MN is a subnet N' = (S',T';F') of (S,T;F) which satisfies

- $\forall s \in S'$: ${}^{\bullet}s \cup s^{\bullet} \subseteq T'$;
- $\forall t \in T'$: $|{}^{\bullet}t \cap S'| = |t^{\bullet} \cap S'| = 1$;
- The graph of F' is strongly connected;
- $\sum_{s \in S'} \mu_0(s) = 1$. ∎



∇ -components &-components

Fig. 3.6

<u>Definition 3.14</u> Let MN = $(S,T;F,\mu_0)$ be a marked net and N' = (S',T';F') a subnet of (S,T;F). Then N' is a <u>strongly connected T-graph component</u> (<u>T-component</u> for short) of MN iff

- $\forall t \in T'$: $^{\bullet}t \cup t^{\bullet} \subseteq S'$;
- $\forall s \in S'$: $|^{\bullet}s \cap T'| = |s^{\bullet} \cap T'| = 1$;
- The graph of F' is strongly connected. ∎

A T-component is sometimes referred to as a strongly connected marked graph component [8]. A marked graph, however, comes with a marking class while we cannot say anything definite about the induced submarkings of a T-component.

There is a close relationship between the notions of ∇-components and &-components on the one hand and the S-components on the other hand. In particular we have

<u>Theorem 3.7</u> Let BP = $(V_\nabla, V_\&, A; Q, Z, M^0)$ be a bp scheme and MN = $(S,T;F,\mu_0)$ its fc representation. Let $N_1 = (S_1,T_1;F_1)$ be an S-component and $N_2 \doteq (S_2,T_2;F_2)$ a T-component of MN. Set for i = 1,2

$$V_\nabla^i = \{v \in V_\nabla \mid \bar{v} \in S_i\}$$
$$V_\&^i = \{u \in V_\& \mid \bar{u} \in T_i\}$$
$$A^i = \{a \in A \mid \bar{a} \in S_i\}$$
$$Q^i = Q \cap (A^i \times V)$$
$$Z^i = Z \cap (A^i \times V)$$

Then $BG^1 = (V_\nabla^1, V_\&^1, A^1; Q^1, Z^1)$ is a ∇-component of BP and $BG^2 = (V_\nabla^2, V_\&^2, A^2; Q^2, Z^2)$ is a &-component of BP.

<u>Proof</u> Follows easily from the definitions. ∎

The following fact, among many others, is known about the structure of live and safe marked free choice nets.

Theorem 3.8 (Hack) Let $MN = (S,T;F,\mu_0)$ be a marked free choice net which is live and safe. Then for every element of MN, $x \in S \cup T$, there is an S-component $(S^1,T^1;F^1)$ and a T-component $(S^2,T^2;F^2)$ of MN such that $x \in S^1 \cup T^1$ and $x \in S^2 \cup T^2$.

Proof See [8]. ∎

We can now extract the result that we have been after all along.

Theorem 3.9 Let $BP = (V_\nabla,V_\&,A;Q,Z,M^0)$ be a well behaved bp scheme. Then there exists for every arc $a \in A$ a ∇-component $(V^1_\nabla,V^1_\&,A^1;Q^1,Z^1)$ and a &-component $(V^2_\nabla,V^2_\&,A^2;Q^2,Z^2)$ such that $a \in A^1$ and $a \in A^2$.

Proof Let $MN = (S,T;F,\mu_0)$ be the fc representation of BP. By proposition 3.1 and theorem 3.5, MN is a live and safe free choice net. The required result now follows easily from theorems 3.8 and 3.7. ∎

3.5 The Relative Expressive Power of Well Behaved bp Schemes

We shall conclude this section with some remarks about the relationship between well behaved bp schemes and known classes of live and safe marked nets. For convenience, we will assume that every net we refer to is connected. Also, instead of repeatedly saying live and safe, we will shortly say ls.

The first class of marked nets of interest are state machines. A state machine is a marked net $SM = (S,T;F,\mu_0)$ in which the underlying net $SG = (S,T;F)$ is an S-graph (see definition 3.2). It is trivial to verify that SM is live and safe iff SG is strongly connected and $\sum_{s \in S} \mu_0(s)=1$.

The dual notion corresponding to an ls state machine is an ls marked T-graph. To start with, we observe that a marked graph can be viewed as a marked net as follows. Let $MG = (V,A;Q,Z,M_0)$ be a marked graph. Then the net representation of MG is the marked net MN = $(S,T;F,\mu_0)$ where S = A, T = V, $F = Q^{-1}UZ$ and $\mu_0 = M_0$. That (S,T;F) is a T-graph is trivial to verify. And we have seen in section 1 that an ls marked graph is strongly connected.

Now under some reasonable definition of the notion "equivalently represented by", every ls state machine as well as every ls marked graph can be equivalently represented by a well behaved bp scheme. We shall not work out the details here. Rather, we indicate the main ideas through two examples shown in fig. 3.7.

Fig. 3.7

In fig. 3.7, diagram (b) shows the bp representation of the marked graph shown in (a), and (d) is a bp representation of the marked S-graph of (c).

Now we already know what it means to represent a well behaved bp scheme as an ls marked free choice net. The simulation ideas contained in the proofs of lemmas 3.2 and 3.4 can be used to prove that every well behaved bp scheme has an 'equivalent' representation as an ls marked free choice net.

It is trivial to observe that every ls state machine and every ls marked graph (through its net representation) can be viewed as an ls marked free choice net. The interesting point about well behaved bp schemes is that they lie properly between ls marked free choice nets on the one hand and ls state machines and ls marked graphs on the other hand. In other words, the class of fc representations of well behaved bp schemes is properly included within the class of ls marked free choice nets. Once again we will not express this formally but rather, illustrate the idea by an example.

The bp scheme shown in fig. 3.8 (a) is not well behaved but its fc representation is live and safe. And we feel confident that under any reasonable choice of definitions, one can show that the ls marked free choice net shown in fig. 3.8 (b) is not an fc representation of any well behaved bp scheme.

Fig. 3.8

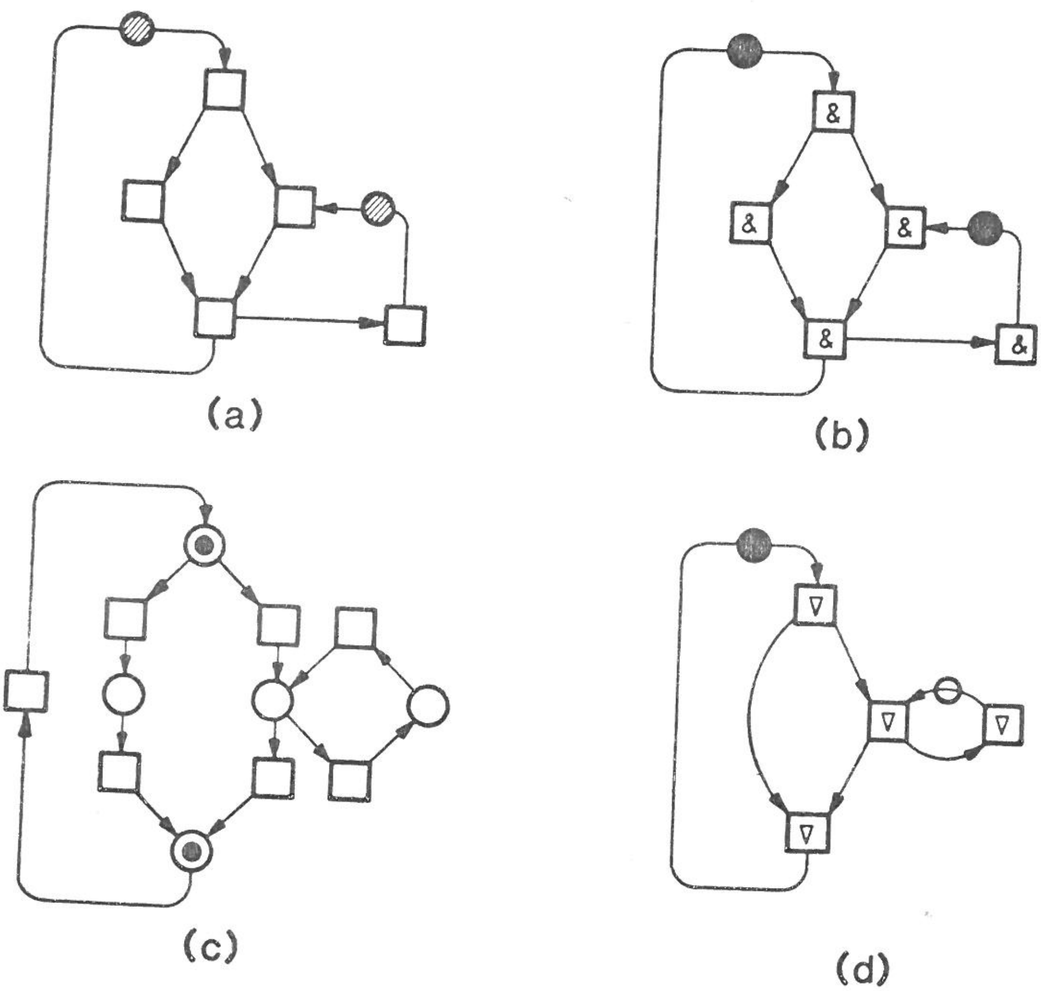


Fig. 3.7

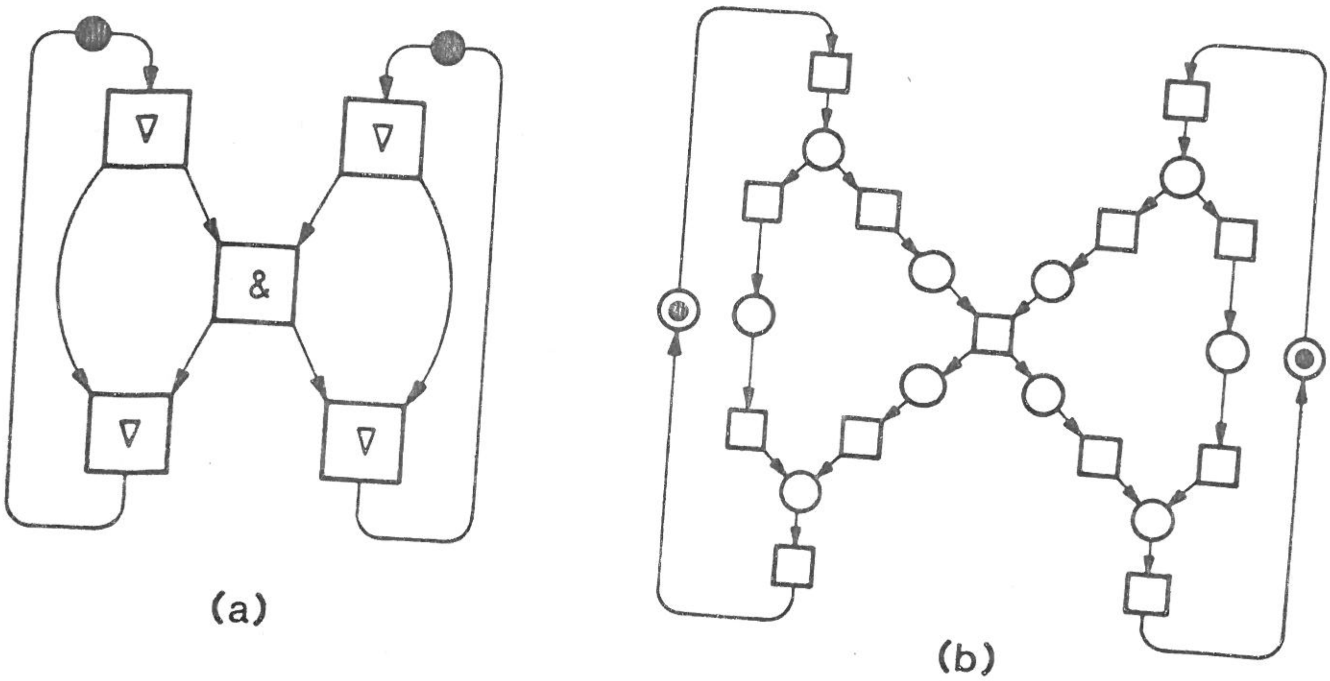


Fig. 3.8

Incidentally, this example also shows that the converse of theorem 3.5 is indeed false. Thus in well behaved bp schemes we have identified a more restricted way (than the one expressed by marked free choice nets) of combining state machines and marked graphs. The schemes fit neatly within the hierarchy of the three well understood classes of ls marked nets (fig. 3.9).

Fig. 3.9

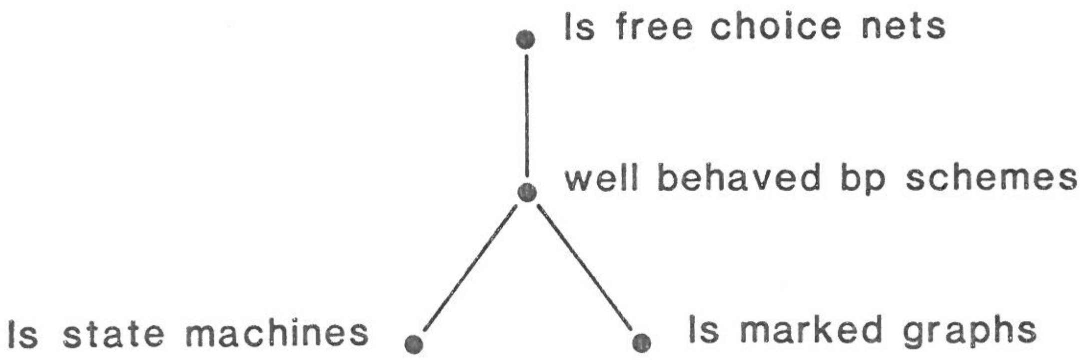

ls free choice nets
well behaved bp schemes
ls state machines
ls marked graphs

Fig. 3.9

4. THE MARKING CLASS OF A WELL BEHAVED BP SCHEME

In this section we establish a number of properties of the two marking classes associated with a bp scheme. In particular, we wish to prove that a bp scheme BP is well behaved iff the underlying bp graph is well behaved at every marking of the full marking class of BP. We have called this

4.1 The All-or-None Property

First we show that if the result we are after is false, the choice between good and bad behaviour can be made to hinge on just one decision. The following result based on the theory of marked graphs will be often appealed to in the proofs of the subsequent results. As before, BP will denote our generic bp scheme, $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$, and BG its underlying bp graph, $BG = (V_\nabla, V_\&, A; Q, Z)$.

Proposition 4.1 Let BP be a bp scheme, $w \in V$ and $\sigma \in V^*$ with $|\sigma|_w = 0$, and $M, M^1, M^2 \in [M^0\rangle$ such that $M[w\rangle M^1$ and $M[\sigma\rangle M^2$. Then there exists $M^3 \in [M^0\rangle$ such that $M^1[\sigma\rangle M^3$ and $M^2[w\rangle M^3$.

Proof We merely need to observe that in a marked graph a node can lose its concession to fire only through its own firing.

∎

Proposition 4.2 Let BP be a bp scheme and $M, M' \in [M^0]$ - the full marking class of BP - such that BG is well behaved at M but not at M'. Then there exist $\hat{M}^1, \hat{M}^2 \in [M^0]$ and $w \in V$ such that $\hat{M}^1[w\rangle \hat{M}^2$ and BG is well behaved at $\hat{M}^2$ but not at $\hat{M}^1$.

Proof Since $M, M' \in [M^0]$ we can find $w_1, w_2, \ldots, w_n \in V$ and markings $M^1, M^2, \ldots, M^{n+1} \in [M^0]$ such that $M^1 = M$, $M^{n+1} = M'$ and for $1 \leq i \leq n$, $M^i[w_i\rangle M^{i+1}$ or $M^{i+1}[w_i\rangle M^i$. Since BG is well behaved at M^1 but not at M^{n+1}, we must have for some i, $1 \leq i \leq n$, that BG is well behaved at $\bar{M}^i$ but not at M^{i+1}. If $M^i[w_i\rangle M^{i+1}$, BG would also be well behaved at M^{i+1}, so $M^{i+1}[w_i\rangle M^i$. Now set $\hat{M}^1 = M^{i+1}$, $\hat{M}^2 = M^i$ and $w = w_i$.

∎

<u>Proposition 4.3</u> Let BP be a bp scheme and as before, $M, M' \in [M^0]$ such that BG is well behaved at M but not at M'. Then there exist $\hat{M}^1, \hat{M}^2, \hat{M}^3 \in [M^0]$ and $v \in V_\nabla$ such that:

- v is enabled to H-fire at $\hat{M}^1$ and $|v^\bullet|>1$,
- $\hat{M}^1[v>\hat{M}^2$ and $\hat{M}^1[v>\hat{M}^3$,
- BG is well behaved at $\hat{M}^2$ but not at $\hat{M}^3$.

<u>Proof</u> By the previous proposition, we can find $M^1, M^2 \in [M^0]$ and a node w such that $M^1[w>M^2$, and BG is well behaved at M^2 but not at M^1. By theorem 2.5 we have that $M_H^2 \neq \emptyset$ so that $M_H^1 \neq \emptyset$. Since BG is not well behaved at M^1, we can find, once again due to theorem 2.5, a marking $M^3 \in [M^1>$ such that some node w' is in deadlock at M^3. Let $\sigma \in V^*$ be such that $M^1[\sigma>M^3$.

We claim that $|\sigma|_w>0$. If $|\sigma|_w=0$, there is - due to proposition 4.1 - a marking M^4 which satisfies $M^2[\sigma>M^4$ and $M^3[w>M^4$. Hence w' would be in deadlock at M^4 contradicting the well behavedness of BG at M^2 ($M^4 \in [M^2>$). So indeed $|\sigma|_w>0$.

Let $\sigma = \sigma_1 w \sigma_2$ such that $|\sigma_1|_w=0$. Let $M^4, M^5 \in [M^1>$ such that $M^1[\sigma_1>M^4[w>M^5[\sigma_2>M^3$ (see fig. 4.1).

Fig. 4.1

Since $|\sigma_1|_w=0$, we have from proposition 4.1 that for some $M^6 \in [M^1>$, $M^2[\sigma_1>M^6$ and $M^4[w>M^6$. BG is well behaved at M^6 since $M^6 \in [M^2>$ and BG is well behaved at M^2. BG is not well behaved at M^5 since $M^3 \in [M^5>$, w' is in deadlock at M^3. So $M_5 \neq M_6$ although $M^4[w>M^5$ and $M^4[w>M^6$. It follows from proposition 2.1 that w is a ∇-node which is enabled to H-fire at M^4 and $|w^\bullet|>1$.

Now set $\hat{M}^1=M^4$, $\hat{M}^2=M^5$, $\hat{M}^3=M^6$ and v=w to get the required result. ∎

To establish the all-or-none property, we need to go one step further; our proof strategy depends on the fact that the marking $\hat{M}^1$ mentioned in the above lemma can be assumed to be v-extremal.

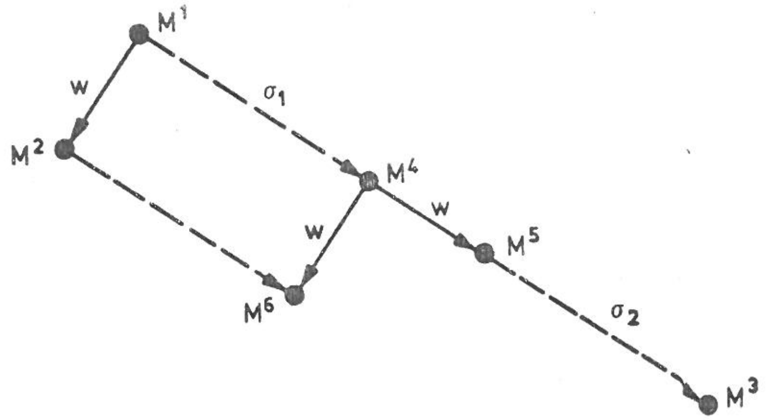


Fig. 4.1

<u>Definition 4.1</u> Let BP be a bp scheme, $M \in [M^0]$ and $w \in V$. M is said to be <u>w-extremal</u> iff

- w is enabled to H-fire at M,
- no other node is enabled at M, and
- no node is in deadlock at M. ∎

We note that if M is w-extremal in BP, $\tilde{M}$ (= $M_H \cup M_L$) is w-extremal in the underlying marked graph $\widetilde{BP}$. The converse is in general not true, due to the first and the third restriction in the above definition. We wish to show that the choice between good and bad behaviour can be made at an extremal marking associated with a ∇-node. We do this in two steps.

<u>Lemma 4.4</u> Let BP be a bp scheme with $M_H^0 \neq \emptyset$. If BP is not well behaved then there exists a marking $M \in [M^0\rangle$ such that no node is enabled to fire at M.

<u>Proof</u> By theorem 2.5, we find $M^1 \in [M^0\rangle$ and $w \in V$ such that w is in deadlock at M^1. Now in the underlying marked graph $\widetilde{BP}$, if starting from any marking in the marking class, some node is prevented from firing, the remaining nodes can each fire at most a bounded number of times. This is an easy consequence of theorem 1.1 since the graph is strongly connected.

Thus starting from M^1 nodes other than w can fire at most a bounded number of times while w remains in deadlock. Hence eventually, we can reach a marking M at which no node can fire. ∎

<u>Proposition 4.5</u> Let BP be a bp scheme and $M, M' \in [M^0]$ such that BG is well behaved at M but not at M'. Then there exists a ∇-node v and markings $\hat{M}^1, \hat{M}^2, \hat{M}^3 \in [M^0]$ such that

- $\hat{M}^1$ is a v-extremal marking,
- $\hat{M}^1[v\rangle\hat{M}^2$ and $\hat{M}^1[v\rangle\hat{M}^3$,
- BG is well behaved at $\hat{M}^2$ but not at $\hat{M}^3$.

Proof By proposition 4.3, we find a ∇-node v and markings $M^1, M^2, M^3 \in [M^0]$ such that v can H-fire at M^1, $M^1[v>M^2$ and $M^1[v>M^3$, and BG is well behaved at M^2 but not at M^3. Then $M^2_H \neq \emptyset$ so that $M^1_H \neq \emptyset$, and BG is not well behaved at M^1 because $M^3 \in [M^1>$. Thus by lemma 4.4 we find $M^d \in [M^1>$ such that no node can fire at M^d.

Let $\sigma \in V^*$ such that $M^1[\sigma>M^d$. Since v can lose its concession to fire (at M^1) only by its own firing, we have $|\sigma|_v>0$. So there are σ_1, σ_2 such that $\sigma = \sigma_1 v \sigma_2$ and $|\sigma_1|_v=0$. We now assume that the firing of v in σ has been postponed as much as possible: If $\sigma' \in V^*$ such that $M^1[\sigma'>M^d$, and $\sigma' = \sigma_1' v \sigma_2'$ with $|\sigma_1'|_v=0$, then $|\sigma_1'| \leq |\sigma_1|$.

Let M^4 and M^5 be such that $M^1[\sigma_1>M^4[v>M^5[\sigma_2>M^d$. Because $|\sigma_1|_v=0$, we can find M^6 such that $M^4[v>M^6$ and $M^2[\sigma_1>M^6$. (See fig. 4.2.)

Fig. 4.2

Since $M^6 \in [M^2>$ and BG is well behaved at M^2, it is also well behaved at M^6, but not at M^5. So all that remains to be shown is that M^4 is v-extremal.

Firstly we note that v can H-fire at M^4. Secondly, no node is in deadlock at M^4 because it would also be in deadlock at M^6 but BG is well behaved at M^6.

Now suppose that some node w≠v can also fire at M^4. Since no node can fire at M^d, w must appear in σ_2. Let $\sigma_2 = \sigma_{21} w \sigma_{22}$ where $|\sigma_{21}|_w=0$. Then it is easy to see that $\sigma' = \sigma_1 w v \sigma_{21} \sigma_{22}$ is also a firing sequence at M^1 with $M^1[\sigma'>M^d$. The existence of σ', however, contradicts our assumption that the firing of v in σ is as late as possible. So no node different from v is enabled to fire at M^4; M^4 is v-extremal.

Set $\hat{M}^2 = M^6$ and $\hat{M}^3 = M^5$ to complete the proof. ∎

The second step in our proof of the all-or-none property is to establish that the situation at an extremal marking as outlined in proposition 4.5 can never arise. First a simple observation.

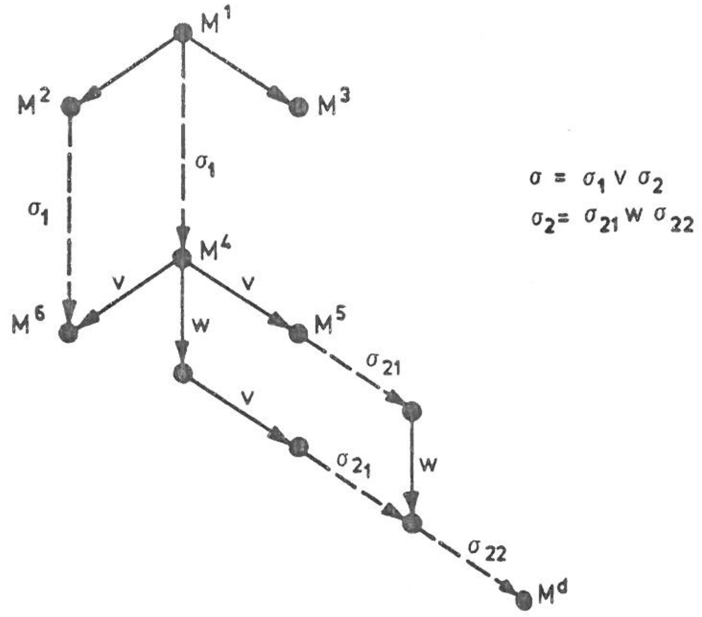


Fig. 4.2

Lemma 4.6 Let BP be a well behaved bp scheme, $a \in A$ and $Z(a)=w$. Then there exists a w-extremal marking $M \in [M^0>$ with $a \in M_H$.

Proof First we can find a marking $M^1 \in [M^0>$ at which $Q(a)=w'$ can H-fire. Then we choose a marking M^2 with $M^1[w'>M^2$ such that $a \in M_H^2$. At M^2, there is a minimal w-enabling firing sequence σ; let $M^2[\sigma>M^3$. Then w can H-fire at M^3 and $a \in M_H^3$.

At M^3, there is a firing sequence σ' of maximum length which satisfies $|\sigma'|_w=0$ (follows easily from theorem 1.1 and proposition 2.3; see proof of lemma 4.4). Let $M^3[\sigma'>M$. Then M is w-extremal with $a \in M_H$. ∎

The next result is crucial for our current purposes. Furthermore, it is quite an interesting result in its own right. The result states that in a well behaved bp scheme, the extremal markings are - almost - uniquely determined by the extremal markings of the underlying marked graph.

Theorem 4.7 Let BP be a well behaved bp scheme, w a node and $M \in [M^0>$ a w-extremal marking. Let b be an arc which is not an input arc of w which carries a token at M ($b \in (M_H \cup M_L) - {}^\bullet w$). Then $b \in M_H$ iff Z(b) is a &-node; and consequently, $b \in M_L$ iff Z(b) is a ∇-node.

Proof Let $w' = Z(b)$. Since $b \notin {}^\bullet w$, $w' \neq w$. From the fact that M is w-extremal it follows easily that there is, at M, a token-free path from w to w' ($w <_M w'$). From theorem 1.6 we know that in the underlying marked graph $\widetilde{BP}$ (which is live) there is, at $\widetilde{M}$, a minimal w'-enabling firing sequence σ, and for $\widetilde{M}[\sigma>M_1$ there is a marking M^1 of BP with $M[\sigma>M^1$ such that $M_1=\widetilde{M}^1$ (prop. 2.3).

In the proof of part 1) of proposition 2.4 we have seen that an H-token sitting in front of w may be steered, during the firing of σ, along the token-free path leading from w to w'. So we can choose M^1 with $M[\sigma>M^1$ such that for some input arc of w', say b' - which is not marked at M and hence which is different from b - $b' \in M_H^1$. Since σ is minimal w'-enabling, $|\sigma|_{w'}=0$. Thus, b is marked in the same way at M^1 as at M.

If $w' = Z(b)$ is a &-node, then $b \in M_H$ because otherwise w' would be in deadlock at M^1 ($^\bullet w' \cap M_H^1 \neq \emptyset$ and $^\bullet w \cap M_L^1 \neq \emptyset$ contradicting the well behavedness of BP.) Conversely, if $b \in M_H$ then $w' \in V_\&$ because a ∇-node w' would be in deadlock at M^1 ($|^\bullet w \cap M_H| > 1$).

So $b \in M_H$ iff $Z(b) \in V_\&$, and this clearly implies that $b \in M_L$ iff $Z(b) \in V_\nabla$. ∎

At last we can demonstrate that the situation as outlined in prop. 4.5 is impossible.

<u>Proposition 4.8</u> Let BP be a bp scheme, $v \in \nabla$-node and $M \in [M^0>$ a v-extremal marking. Suppose that $M[v>M'$ such that BG is well behaved at M'. Then $M \in [M'>$. As a result, BG is well behaved at M, too.

<u>Proof</u> Let a be the input arc of v that carries an H-token at M. Since BG is well behaved at M' we find $M'' \in [M'>$ such that once again M" is v-extremal and $a \in M''_H$ (lemma 4.6). We shall prove that M" is the same marking as M.

First we note that $\tilde{M}$ and $\tilde{M}''$ are v-extremal markings in the underlying marked graph $\tilde{BP}$. So by theorem 1.8 we have $\tilde{M} = \tilde{M}''$, i.e. $M_H \cup M_L = M''_H \cup M''_L$. Thus the same subset of A is marked at both M and M". We just need to show that the 'colour' of the tokens match up for each marked arc at M and M".

We shall do so by induction. As a suitable index we propose the 'depth' of each arc in $\tilde{M}$ $(=\tilde{M}'')$.

Let depth: $\tilde{M} \to \mathbb{N}$ be, inductively, given by

$$\text{depth}(b) = \begin{cases} 0, \text{if } b \in {}^\bullet v \\ 1+\max\{\text{depth}(b') \mid Z(b') <_{\tilde{M}} Z(b),\ b' \in \tilde{M}\}, \text{otherwise} \end{cases}$$

Let $b \in \tilde{M}$. We need to prove that $b \in M_H$ iff $b \in M''_H$. The proof is by induction on $k = \text{depth}(b)$.

<u>k=0</u> Then $b \in {}^{\bullet}v$. M" was chosen in such a way that $b \in M_H$ iff $b \in M''_H$.

<u>k>0</u> There are two cases to consider.

<u>Case 1: $Z(b)=v' \in V_\nabla$</u>
Since depth(b)=k>0, we have that $v' \neq v$. By theorem 4.7, we also know that $b \in M''_L$. Hence we need to show that $b \in M_L$.

The bp scheme $BP' = (V_\nabla, V_\&, A; Q, Z, M')$ is well behaved and consequently, covered by its set of ∇-components (theorem 3.9). In particular, we can find a ∇-component $BG^1 = (V^1_\nabla, V^1_\&, A^1; Q^1, Z^1)$ such that $b \in A^1$. By proposition 3.6 we know that BG^1 is also a ∇-component of BP. Since $b \in A^1$ and $Z(b) = v' \in V_\nabla$, we have that $v' \in V^1_\nabla$.

Now let $\Pi \in (A^1)^*$ be a token-free path at M (completely contained in BG^1) leading to v' of maximum length. The existence of Π is guaranteed by the fact that, since v' cannot fire at M, there is a token-free input arc of v';which belongs to BG^1 because v' is a ∇-node.

Let $\Pi = a_1 a_2 \ldots a_n$ and $w = Q(a_1)$. Since $Z(a_n) = v'$ and Π is token-free at M, and $\widetilde{BP}$ is live, we have that $w \neq v'$.

Suppose that w=v. Then ${}^{\bullet}v \subseteq A^1$ because $a_1 \in A^1$ and $Q(a_1) = w = v \in V^1_\nabla$. Now ${}^{\bullet}v \cap M_H \neq \emptyset$ and $|A^1 \cap M_H| = 1$. Consequently, $(A^1 - {}^{\bullet}v) \cap M_H = \emptyset$ and thus, ${}^{\bullet}v' \cap M_H = \emptyset$. As a result $b \in M_L$ and we are done.

Now suppose that $w \neq v$ but w is a ∇-node, i.e. $w \in V^1_\nabla$. Since Since M is v-extremal, at least one input arc of w, say a_0, is unmarked at M. a_0 belongs to BG^1 since $w \in V^1_\nabla$. Consequently, the token-free path $\Pi' = a_0 \Pi$ leading to v' is also contained in BG^1 contradicting our assumption that Π is such a path of maximum length. Hence if $w \neq v$ then w is not a ∇-node.

So assume that w is a &-node. Since $w = Q(a_1)$ and $a_1 \in A^1$, $w \in V^1_\&$ and, by definition of a ∇-component, $|{}^{\bullet}w \cap A^1| = \{b'\}$ for some arc b'. Because of the maximality of Π, $b' \in M_H \cup M_L = M''_H \cup M''_L$ and consequently, by theorem 4.7, $b' \in M''_H$.

Now because of the existence of Π, $w <_{\widetilde{M}} v'$ and hence depth(b') < depth(b) = k. So by induction hypothesis, we have

$b' \in M_H$ because as already observed $b' \in M''_H$. But this implies that $(A^1 - \{b'\}) \cap M_H = \emptyset$ because BG^1 is a ∇-component. Thus $b \in M_L$ and we are done with the first case.

Case 2: $Z(b) = u \in V_{\&}$

Again from theorem 4.7 we have $b \in M''_H$. So we have to show that $b \in M_H$.

Clearly $u \neq v$. So there is an input arc b' of u which is unmarked at any v-extremal marking such as M and M''. Let $w = Q(b')$.

Suppose $w = v$. Then by theorem 4.7 and the fact that v has more than one output arc, we have at M'', a situation as shown in fig. 4.3 which obviously may lead u into a deadlock. But this is contradicting the well behavedness of BG at M''.

Fig. 4.3

Consequently, $Q(b') = w \neq v$. Now let $a_1 \in v \cap M'_H$ be the output arc of v that receives the H-token in the transition $M[v>M'$. Furthermore, let M''' be the result of v firing at M'' in the same way as at M, i.e. $M''[v>M'''$ and $a_1 \in M'''_H$. Clearly $\tilde{M}' = \tilde{M}'''$.

Define $\bar{A} = \{a \in \tilde{M}' \mid Z(a) = w \text{ or } Z(a) <_{\tilde{M}'} w\}$. Then clearly for all $a \in (\bar{A} - v^{\bullet})$, $\text{depth}(a) < \text{depth}(b) = k$. Hence, by induction hypothesis, for all $a \in (\bar{A} - v^{\bullet})$, $a \in M_H \langle M_L \rangle$ iff $a \in M''_H \langle M''_L \rangle$. Consequently for all $a \in \bar{A}$, $a \in M'_H \langle M'_L \rangle$ iff $a \in M'''_H \langle M'''_L \rangle$.

Now assume that $b \in M_L$ and, consequently, $b \in M'_L$. Consider first the case that $\bar{A} \cap M'_H \neq \emptyset$. Then by proposition 2.4 we find a minimal w-enabling firing sequence σ at M' such that $M'[\sigma>M^1$ and w can H-fire at M^1. Since $|\sigma|_u = 0$ and $b \in M'_L$, $b \in M^1_L$ also. At M^1, w may H-fire w in such a way that b' acquires an H-token. Then u is in deadlock, at the resulting marking; a contradiction, because BG is supposed to be well behaved at M'.

Hence we have $\bar{A} \subseteq M'_L$ and as a result $\bar{A} \subseteq M'''_L$ also. Once again by proposition 2.4, we can find a minimal w-enabling firing sequence σ at M''' such that $M'''[\sigma>M^2$ and w can L-fire at M^2. At the marking obtained by w L-firing (at M^2), however, b' carries an L-token while b still carries an H-token. So u is in deadlock at a marking reachable, via M'', from M'. A contradiction.

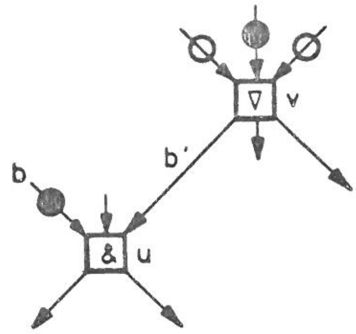


Fig. 4.3

Thus indeed $b \in M_H$. This completes the induction step and we have $M = M''$. Consequently $M \in [M'\rangle$ and because BG is well behaved at M', BG is well behaved at M, too. ∎

The all-or-none property follows now easily.

Theorem 4.9 Let $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$ be a bp scheme. BP is well behaved iff for every $M \in [M^0]$, $BG = (V_\nabla, V_\&, A; Q, Z)$ is well behaved at M.

Proof Suppose that BP is well behaved, but BG is not well behaved at some $M \in [M^0]$. Then due to proposition 4.5 we find a ∇-node v, a v-extremal marking $M^1 \in [M^0]$ and two other markings $M^2, M^3 \in [M^0]$ such that $M^1[v\rangle M^2$ and $M^1[v\rangle M^3$, and BG is well behaved at M^2 but not at M^3. According to proposition 4.8, however, $M^1 \in [M^2\rangle$ so that BG is also well behaved at M^1. Consequently BG must be well behaved at M^3, too. A contradiction.
The second half of the theorem is trivial. ∎

4.2 Some Additional Properties of the Marking Class

As we have seen already, the result stated in theorem 4.7 is a very useful one. Here we shall bring out just one of its main consequences. It is: The marking class of a well behaved bp scheme is uniquely determined by the marking class of the underlying marked graph (which is not determined uniquely by the supporting strongly connected digraph).

Theorem 4.10 Let $BP_1 = (V_\nabla, V_\&, A; Q, Z, M^1)$ and $BP_2 = (V_\nabla, V_\&, A; Q, Z, M^2)$ be a pair of well behaved bp schemes based on the same bp graph. Then $[M^1] = [M^2]$ iff for the underlying marked graphs, $[\tilde{M}^1] = [\tilde{M}^2]$.

Proof If $[M^1] = [M^2]$ then clearly $[\tilde{M}^1] = [\tilde{M}^2]$. So assume that $[\tilde{M}^1] = [\tilde{M}^2]$. To start with we observe that by definition of the full marking class, $[M^1] = [M^2]$ iff $[M^1] \cap [M^2] \neq \emptyset$. So we shall show that there is a marking which is both in $[M^1]$ and $[M^2]$. To this end let w be a node and $M \in [M^1]$ and $M' \in [M^2]$ two w-extremal markings such that $({}^\bullet w \cap M_H) = ({}^\bullet w \cap M'_H)$. Clearly $\tilde{M}$ and $\tilde{M}'$ are two w-extremal markings for the digraph $\tilde{BG}$. Since $[\tilde{M}^1] = [\tilde{M}^2]$, we have through theorem 1.8 that $\tilde{M} = \tilde{M}'$. Let $a \in M_H \cup M_L$. If $a \in {}^\bullet w$, then by construction, $a \in M_H$ iff $a \in M'_H$. If $a \notin {}^\bullet w$, then by theorem 4.7 $a \in M_H$ iff Z(a) is a &-node iff $a \in M'_H$. Thus indeed $a \in M_H \langle M_L \rangle$ iff $a \in M'_H \langle M'_L \rangle$. In other words M = M', hence $[M^1] \cap [M^2] \neq \emptyset$ and we are done. ∎

The last result of this section states that a well behaved bp scheme shows a unique 'steady state' behaviour.

<u>Theorem 4.11</u> Let BP be a well behaved bp scheme. Then for every $M, M' \in [M^0]$, $[M\rangle \cap [M'\rangle \neq \emptyset$.

<u>Proof</u> By theorem 4.9, the underlying bp graph is well behaved at M and M'. Let w be a node of BP. Then we find two w-extremal markings $M^1 \in [M\rangle$ and $M^2 \in [M'\rangle$ such that ${}^\bullet w \in M^1_H = {}^\bullet w \in M^2_H$. As shown in the proof of theorem 4.10, $M^1 = M^2$ and thus indeed $[M\rangle \cap [M'\rangle \neq \emptyset$. ∎

To finish this section we state two conjectures closely related to the results of this section. The first one is concerned with the question of how long it may take a well behaved bp scheme to reach its steady state behaviour. We believe that the following is true: Let BP be a well behaved bp scheme and let $M^0[\sigma\rangle M^1$ where for all $w \in V$, $|\sigma|_w > 0$ (each node fires at least once). Then M^1 is 'reproducable', i.e. there is $M^2 \in [M^1\rangle$, $M^2 \neq M^1$, such that $M^1 \in [M^2\rangle$.

Our second conjecture is an all-or-none property of the second kind. Let BP be a well behaved bp scheme and $\bar{M}$ be an arbitrary live and safe marking of the underlying digraph. Then there exists a marking M of BG with $\tilde{M} = \bar{M}$ such that BG is well behaved at M.

Once the synthesis procedure has been presented, it will be easier to see why the two conjectures stated above are important.

5. THE SYNTHESIS PROCEDURE

We shall now present a technique for systematically constructing well behaved schemes. The idea, as one would expect, is to start with "small" bp schemes which are trivially well behaved and then repeatedly apply a set of transformation rules to obtain more complex well behaved schemes. The seed schemes are called elementary schemes and they can be of two types.

Definition 5.1 1) A ∇-elementary bp scheme is a scheme $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$ which satisfies:

1a) $|V_\nabla|=1$ and $V_\&=\emptyset$; hence $A=M_H^0 \cup M_L^0$.

1b) $|M_H^0|=1$.

2) A &-elementary bp scheme is a scheme $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$ which satisfies:

2a) $V_\nabla=\emptyset$ and $|V_\&|=1$; hence $A=M_H^0 \cup M_L^0$.

2b) $M_H^0=A$.

3) A bp scheme is called elementary iff it is either ∇-elementary or &-elementary. ∎

Fig. 5.1 shows an example of a ∇-elementary and an &-elementary scheme.

Fig. 5.1

We note that every elementary scheme is well behaved.

5.1 The Transformation Rules

In what follows we let $S, \hat{S}, S_0, S_1$ etc. denote bp schemes. We shall present our transformation rule in a (hopefully) precise but pictorial form accompanied by explanations and remarks. A textual formulation will not contribute significantly to understanding the rules (quite the opposite, we

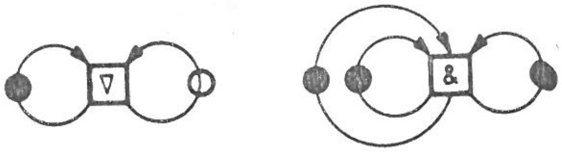


Fig. 5.1

believe) and hence we shall forego the pleasures of doing so.

The general format will be to apply the transformation rule T_i to the source scheme $S = (V_\nabla, V_\&, A; Q, Z, M)$; and obtain the target scheme $\hat{S} = (\hat{V}_\nabla, \hat{V}_\&, \hat{A}; \hat{Q}, \hat{Z}, \hat{M})$. The specification of a rule will consist of four parts.

1) The restrictions on M, the marking of S at which the rule is to be applied.

2) The restrictions that the structure of S (often w.r.t. M) must satisfy in order for the rule to become applicable.

3) The change effected in the structure of S, to yield the structure of $\hat{S}$.

4) The specification of $\hat{M}$, the marking of $\hat{S}$.

There are eight rules on the whole and the first seven rules will have parts 1) and 4) in common. So we shall state them first.

For the rules T_1 through T_7, M, the marking of the source scheme must satisfy: If an arc appears in S but not in $\hat{S}$ then this arc must not be marked at M.

The force of the above restriction is that in going from S to $\hat{S}$ we do not want to lose tokens. Indeed often what causes headache in transforming S to $\hat{S}$ is that we must ensure that $\hat{S}$ is a bp scheme; the underlying marked graph of $\hat{S}$ should be live and safe. Part 4) for the first seven rules reads:

The marking $\hat{M}$ of $\hat{S}$ is obtained as follows. If an arc appears both in S and $\hat{S}$ then it is marked in the same way at $\hat{M}$ as it was at M. If an arc appears in $\hat{S}$ but not in S then such an arc is left unmarked at $\hat{M}$.

Thus in stating the (first seven) rules we need to mostly deal with just the structural restrictions and the changes in the structure.

$\underline{T_1}$ (Arc Refinement)

Fig. 5.2

There are no restrictions. An arc a can always be extended by introducing a new 1-in 1-out node w (which can be either a ∇-node or an &-node) and a new arc a' as shown.

$\underline{T_2}$ (Node refinement)

Fig. 5.3

A node w of type x is split into two nodes w_1 and w_2 of the same type; a non-empty set of new arcs A_{12} is created each of which originates from w_1 and terminates at w_2; $^\bullet w$, the input arcs of w in S is partitioned into $(^\bullet w)_1$ and $(^\bullet w)_2$ so that $(^\bullet w)_1$ becomes the set of input arcs of w_1 and $(^\bullet w)_2 \cup A_{12}$ becomes the set of input arcs of w_2 in $\hat{S}$; $w^\bullet$, the set of output arcs of w in S is partitioned into $(w^\bullet)_1$ and $(w^\bullet)_2$ so that $(w^\bullet)_1 \cup A_{12}$ becomes the set of output arcs of w_1 and $(w^\bullet)_2$ becomes the set of output arcs of w_2 in $\hat{S}$. The crucial restriction this rule must satisfy is:

The partitioning of $^\bullet w$ - into $(^\bullet w)_1$ and $(^\bullet w)_2$ - and the partitioning of $w^\bullet$ - into $(w^\bullet)_1$ and $(w^\bullet)_2$ - must be such that the underlying marked graph of $\hat{S}$ is live and safe.

This rule is unsatisfactory in that its applicability is not a "local" condition. For each instance of its application, we must check globally that $\hat{S}$, the target scheme is - viewed as a marked graph - live and safe. Matters are improved somewhat by observing that in going from S to $\hat{S}$ liveness is automatically preserved. What can get destroyed is safety. It is known to us that the safety of the underlying marked graph of $\hat{S}$ can be checked in time bounded by $O(|A|^3)$ [ ]. And we might be able to do even better by keeping around and updating some information concerning the basic circuits of $\tilde{S}$, the underlying marked graph of the source scheme S. After all, to ensure that the underlying marked graph of $\hat{S}$ is safe, just two things must be guaranteed. Firstly, in $\tilde{S}$ there must be at least one basic circuit passing through some arc in $(^\bullet w)_1$ and some arc in $(^\bullet w)_2$. This provides safety for each of the new arcs (A_{12}). Secondly if in $\tilde{S}$ and arc

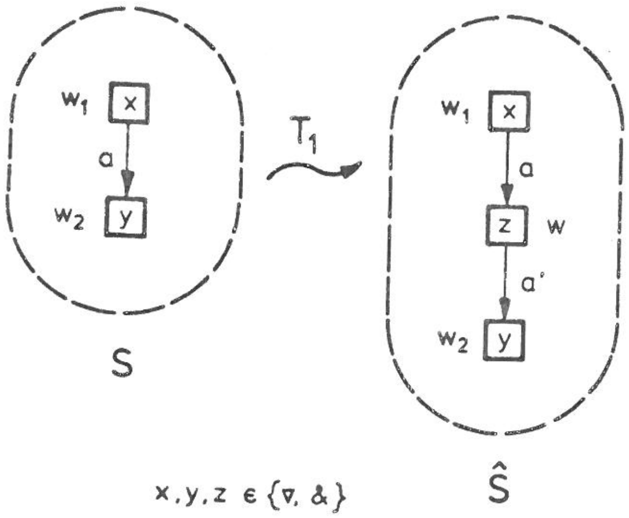


Fig. 5.2

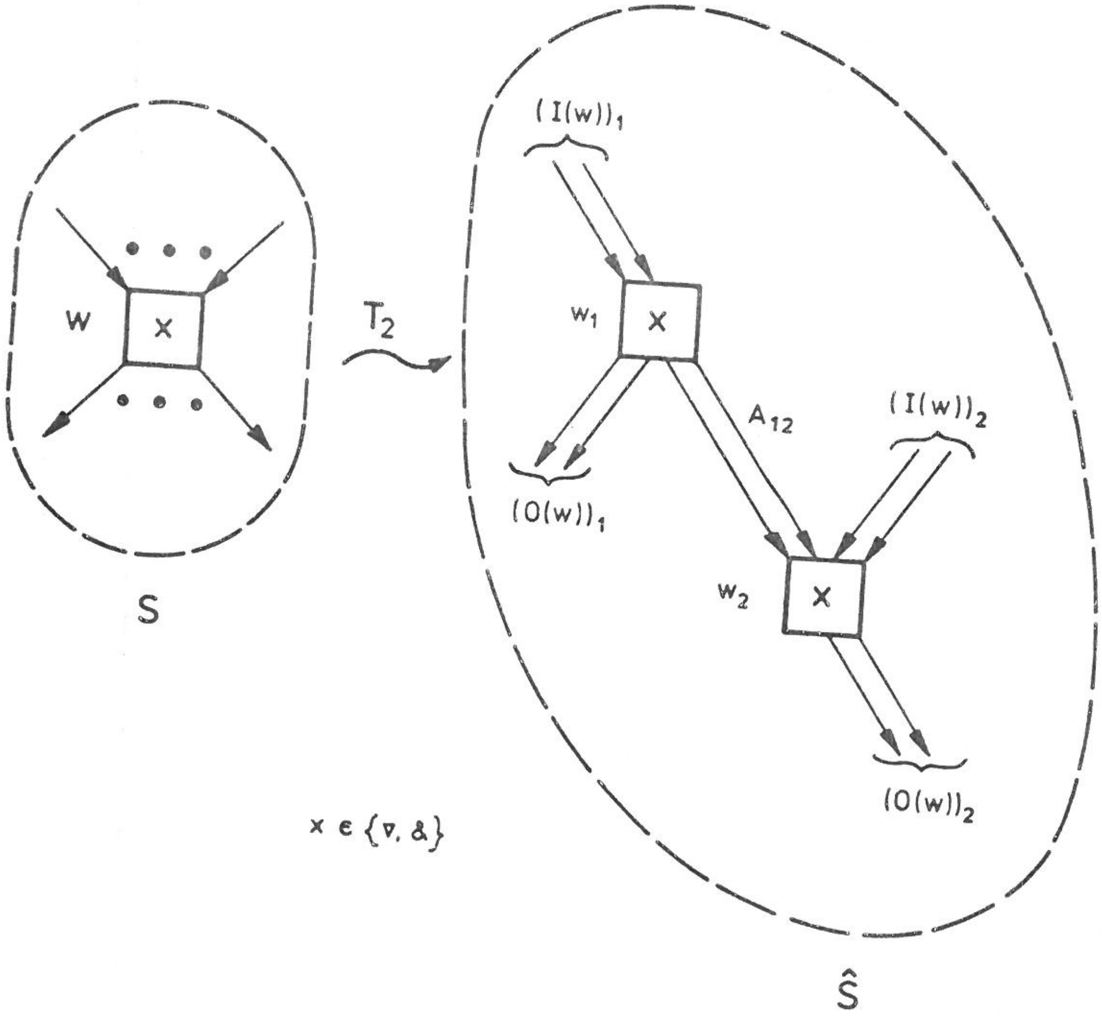


Fig. 5.3

b is contained in a basic circuit passing through some arc in $({}^\bullet w)_2$ and some arc in $(w^\bullet)_1$ - this basic circuit will be destroyed by the application of T_2 - then there must be a basic circuit in S containing b which does not pass through any arc in $({}^\bullet w)_2$ <u>or</u> any arc in $(w^\bullet)_1$. This guarantees that no "essential" basic circuits are destroyed in going over to $\hat{S}$.

<u>T_3 (∇-diamond Transformation)</u>

Fig. 5.4

The arc a is replaced by two arcs a_1 and a_2 as shown. The restrictions that must be satisfied are:

M, the marking of S in addition to the general restriction stated at the beginning must be such that no arc in $\{b, b_1, b_2\}$ is marked at M. As for the structure, in S, the environment of the ∇-node v is no more than what is shown. In other words, in S, ${}^\bullet v = \{b\}$ and $v^\bullet = \{b_1, b_2\}$.

The effect of this rule is to distribute the fork (&) operation over the split (∇) operation. Stated differently, this rule can be equivalently, and for our purposes less conveniently, presented as shown in fig. 5.5

Fig. 5.5

<u>T_4 (&-diamond Transformation)</u>

Fig. 5.6

Once again, the arc a is replaced by the two new arcs a_1 and a_2 as shown. As in the previous rule the restrictions to be met are:

At M, no arc in $\{b, b_1, b_2\}$ should be marked. Moreover, the environment of the &-node u is no more than what is shown; ${}^\bullet u = \{b_1, b_2\}$ and $u^\bullet = \{b\}$.

One of the pleasing aspects of the theory of well behaved bp schemes is that the ∇ and & operations have some strong duality relationships. Most of our theorems about well behaved schemes remain valid if we interchange ∇ and & and reverse the direction of all the arcs. And this duality will be strongly reflected in our synthesis procedure. T_3 and T_4 are reverse duals of each other. As can be easily verified, each of the remaining ones (except the last one) is its own reverse-dual.

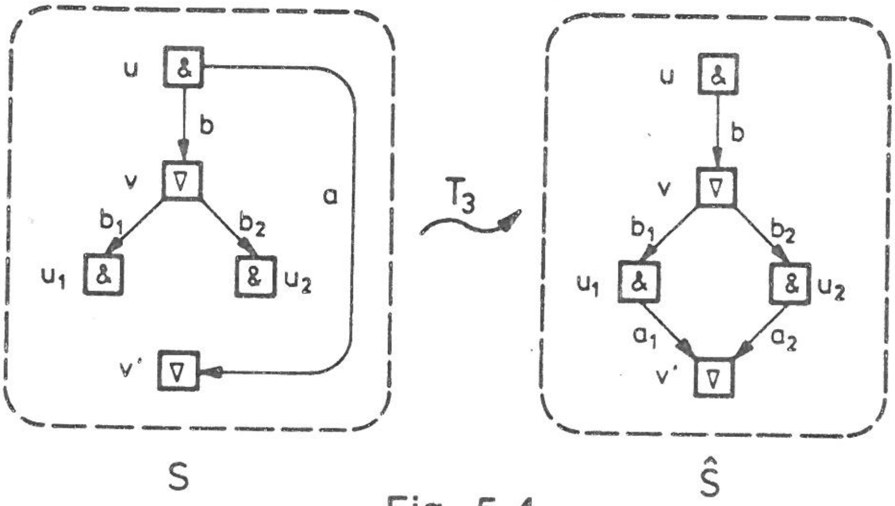


Fig. 5.4

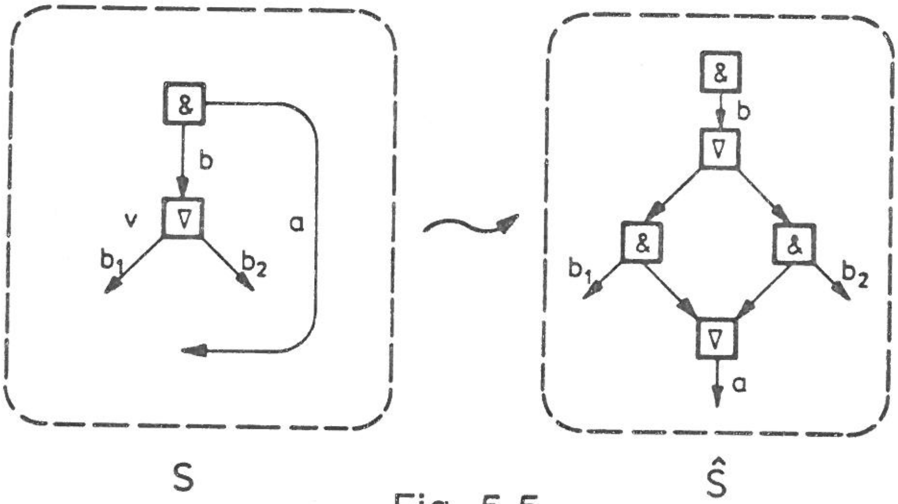


Fig. 5.5

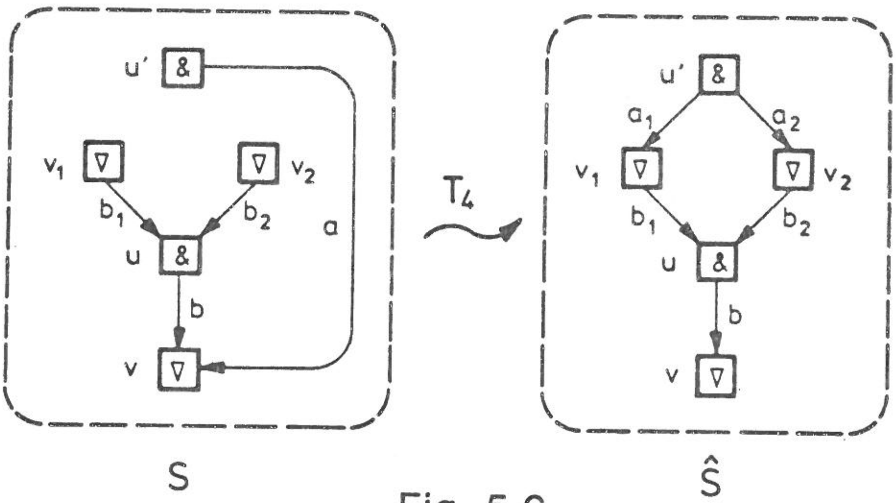


Fig. 5.6

$\underline{T_5}$ (⊠ Transformation)

Fig. 5.7

The two &-nodes u_1 and u_2 are replaced by a single ∇-node v. The two ∇-nodes v_1 and v_2 are replaced by a single &-node u. The four arcs c_{11}, c_{12}, c_{21}, c_{22} are replaced by a single arc c. The restriction to be satisfied is:

The environments of each of the nodes in $\{u_1,u_2,v_1,v_2\}$ is no more than what is shown in fig. 5.7.

The effect of generalising the three previous rules is obtained by introducing two additional rules.

$\underline{T_6}$ (Arc Reduction)

Fig. 5.8

There are no structural restrictions except the obvious one that w must be 1-in 1-out. The arc sequence aa' can be shortened to a through the elimination of the 1-in 1-out node w and the arc a'. This rule is the "reverse" of T_1.

$\underline{T_7}$ (Node Reduction)

Fig. 5.9

Two nodes w_1 and w_2 of the <u>same type</u> such that $\emptyset \neq A_{12} = w_1^{\bullet} \cap {}^{\bullet}w_2$, can be collapsed together, after eliminating A_{12}, to one node w of the same type as w_1 and w_2.
In $\hat{S}$, ${}^{\bullet}w = {}^{\bullet}w_1 \cup ({}^{\bullet}w_2 - A_{12})$ and $w^{\bullet} = (w_1^{\bullet} - A_{12}) \cup w_2^{\bullet}$. The condition that must be satisfied is:

In S at M, there is no token-free path of length greater than one from w_1 to w_2.

As opposed to T_2, in going from S to $\hat{S}$ through T_7, safety will be preserved. What might get lost is liveness. The applicability condition stated above is designed to ensure that the underlying marked graph of $\hat{S}$ is live. This rule is also non-local, but as before, the applicability condition can be checked quite efficiently.

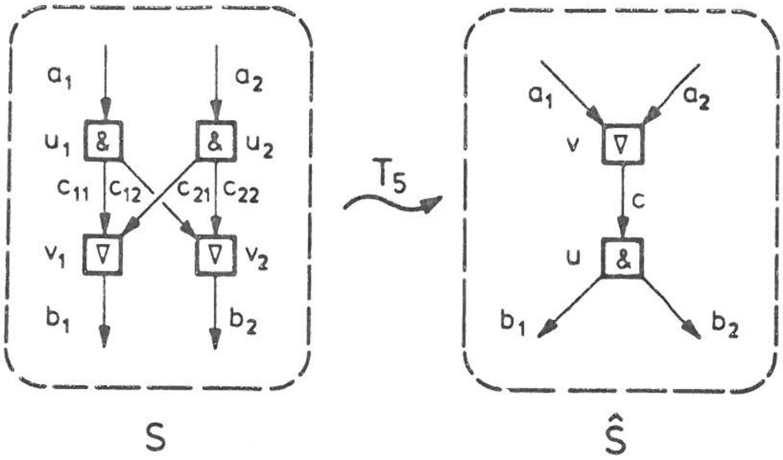


Fig. 5.7

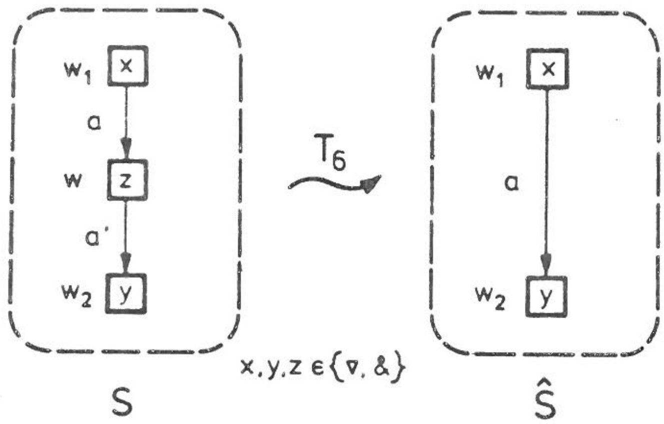


Fig. 5.8

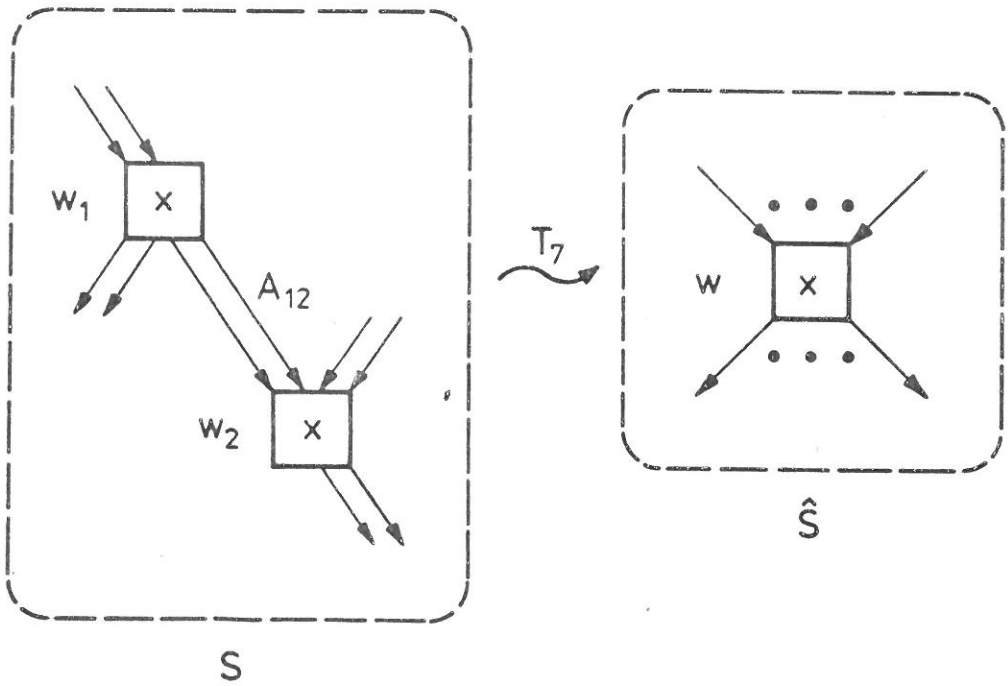


Fig. 5.9

$\underline{T_8}$ (Marking Transformation)

The effect of applying this rule to $S = (V_\nabla, V_\&, A; Q, Z, M)$ is to yield the scheme $\hat{S} = (V_\nabla, V_\&, A; Q, Z, \hat{M})$ where $\hat{M} \in [M]$.

In other words the rule consists of firing the nodes of S forwards or backwards a finite number of times.

This completes our presentation of the transformation rules. The synthesis procedure is outlined in the next definition.

5.2 Well Formed bp Schemes

Definition 5.2 The class of well formed bp schemes is denoted as *WF* and is the smallest class of schemes given by:

1) Every elementary scheme is well formed.

2) If S is well formed and $\hat{S}$ is obtained by applying one of the eight transformation rules to S then $\hat{S}$ is also well formed. ∎

In fig. 5.10 we show the generation of a well formed bp scheme which differs from the scheme of fig. 2.5.a just in the initial marking. The resulting scheme is well behaved and we have also demonstrated through this example that the scheme of fig. 2.5.a is well formed.

Fig. 5.10

An interesting sub-class of *WF* is what we call strongly well formed bp schemes. Intuitively, they are schemes in which the ∇ and & operations are properly "nested". Formally:

Definition 5.3 The class of strongly well formed bp schemes is denoted as *SWF* and is the smallest class of schemes given by

1) Every elementary scheme is strongly well formed.

2) If S is strongly well formed and $\hat{S}$ is obtained by applying T_1 (arc refinement) or T_2 (node refinement) to S then $\hat{S}$ is also strongly well formed. ∎

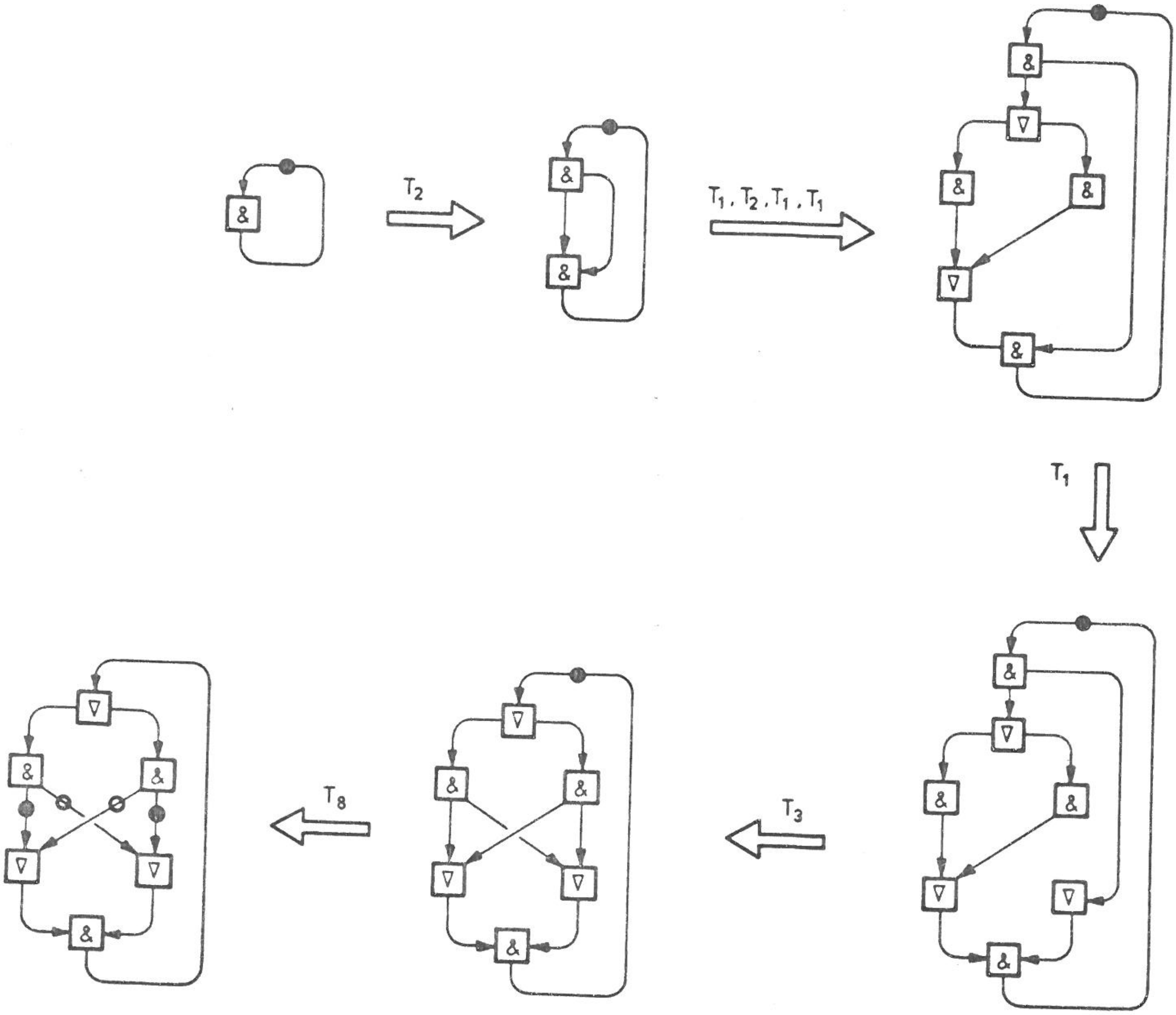


Fig. 5.10

Clearly every strongly well formed scheme is also well formed. That the converse is not true will become clear once we obtain - in the next section - behavioural characterisations of *WF* and *SWF*. It will then be easy to demonstrate that the scheme of fig. 2.5.a is not strongly well formed.

5.3 The Consistency of the Synthesis Procedure

We now wish to prove that our synthesis procedure yields only well behaved schemes. To do so, we shall first argue in detail that T_2, the node refinement rule preserves well behavedness. The main idea is that S, the source scheme, can simulate the behaviour of $\hat{S}$ with a bounded amount of "delay". The remaining parts of the proof are quite straightforward.

Lemma 5.1 Let $S = (V_\nabla; V_\&, A; Q, Z, M)$ be a well behaved bp scheme and $\hat{S} = (\hat{V}_\nabla, \hat{V}_\&, \hat{A}; \hat{Q}, \hat{Z}, \hat{M})$ be obtained by applying T_2 to S. Then $\hat{S}$ is also a well behaved scheme.

Proof We assume that the situation is as shown in fig. 5.3. The node w has been split into w_1 and w_2 with $A_{12} = w_1^\bullet \cap {}^\bullet w_2 = \hat{A} - A$. The key observation concerning the behaviour of $\hat{S}$ is:

Let $\hat{M}^1 \in [\hat{M}>$. Then $\forall x \in \hat{V}_\nabla \cup \hat{V}_\&: w_1 <_{\hat{M}^1} x \Rightarrow x \not<_{\hat{M}^1} w_2$.

This is true because if π_1 is a token free path from w_1 to x and π_2 is a token-free path from x to w_2 at $\hat{M}^1$ then in S, $\pi_1\pi_2$ would be a circuit (passing through w) which is token-free at every marking in [M]. And this is ruled out because $\tilde{S}$ is a live and safe marked graph.

To proceed with the main proof we note that $M_H \neq \emptyset$ and hence $\hat{M}_H \neq \emptyset$. By assuming that $\hat{S}$ is not well behaved we will derive the contradiction that S is also not well behaved. To this end suppose that σ is a firing sequence at $\hat{M}$ with $\hat{M}[\sigma>\hat{M}'$ and u is a node which is in deadlock at $\hat{M}'$. The underlying marked graph of $\hat{S}$ is live and safe. Moreover no arc in A_{12} is marked at $\hat{M}$. Hence in any firing sequence in $\hat{S}$ starting from $\hat{M}$, the nodes w_1 and w_2 would fire alternatingly with w_1 firing first. The proof is by induction on $k = |\sigma|_{w_2}$.

1) <u>k=0</u> Then $|\sigma|_{w_1} \leqq 1$. If $|\sigma|_{w_1}=0$ then σ is also a firing sequence of S at M. Moreover we can find a marking M' with $M[\sigma>M'$ such that either the node u is in deadlock at M' in case $u \notin \{w_1, w_2\}$ <u>or</u> w is in deadlock at M'. In either case we have the contradiction we are looking for.

So suppose that $\sigma = \sigma_1 w_1 \sigma_2$. Let $\sigma_2 = y_1, y_2, \ldots, y_n$ with $y_i \in \hat{V}$ $(= \hat{V}_\nabla \cup \hat{V}_\&)$ and let $\hat{M}[\sigma_1 > \hat{M}^1$ be such that $\hat{M}^1[w_1\sigma_2 > \hat{M}'$. We shall assume that for $1 \leqq i \leqq n$, $w_1 <_{\hat{M}^1} y_i$. To see that this does not involve any loss of generality:

Let i be the least integer in $\{1,2,\ldots,n\}$ such that $w_1 \nless_{\hat{M}^1} y_i$. Then y_i is firable at $\hat{M}^1$ and we can obtain $\hat{M}^1[y_i w_1 y_1 \ldots y_{i-1} y_{i+1} \ldots y_n > \hat{M}'$.By repeatedly applying this transformation we can arrive at a firing sequence which satisfies our requirements. Which, to recall, are: $\hat{M}[\sigma_1 w_1 \sigma_2 > \hat{M}'$; some node u is in deadlock at $\hat{M}'$; $\hat{M}[\sigma_1 > \hat{M}^1 [w_1 \sigma_2 > \hat{M}'$; with $\sigma_2 = y_1 y_2 \ldots y_n$, it is the case that for each y_i, $w_1 <_{\hat{M}^1} y_i$.

Now no arc in A_{12} is marked at $\hat{M}^1$ and hence by firing σ_1 at M (in S) in the same way in which it was fired in $\hat{S}$, we can obtain $M[\sigma_1 > \hat{M}^1$. Let σ' be a minimal w-enabling sequence at $\hat{M}^1$ (in S). If σ' does not exist then S is not well behaved and we are done. Assuming that σ' indeed exists, we claim that for $1 \leqq i \leqq n$, $|\sigma'|_{y_i}=0$. This is because w_1 is firable at $\hat{M}^1$ (in $\hat{S}$) so that σ' is a minimal w_2-enabling sequence at $\hat{M}^1$ in $\hat{S}$. We know that $w_1 <_{\hat{M}^1} y_i$ for each y_i and this at once implies that $y_i \nless_{\hat{M}^1} w_2$ as we observed right at the beginning. From theorem 1.6 it follows at once that $|\sigma'|_{y_i}=0$.

It is now easy to check that in S we can find a marking M' where $M[\sigma_1 \sigma' w \sigma_2 > M'$ such that some node is in deadlock at M'.

2) <u>k>0</u> Let $\sigma = \sigma_1 w_1 \sigma_{12} w_2 \sigma_2$ with $|\sigma_1|_{w_1} = 0 = |\sigma_{12}|_{w_1}$. Let $\hat{M}^1$ be such that $\hat{M}[\sigma_1 > \hat{M}^1$ and $\hat{M}^1[w_1 \sigma_{12} w_2 \sigma_2 > \hat{M}'$. Set $\sigma_{12} = y_1 y_2 \ldots y_n$ with $y_i \in \hat{V}$. As observed in the proof of the basis step we can assume without loss of generality that $w_1 <_{\hat{M}^1} y_i$ for $1 \leqq i \leqq n$. Which lets us conclude that $y_i \nless_{\hat{M}^1} w_2$. As a result, $\sigma' = \sigma_1 w_1 w_2 \sigma_{12} \sigma_2$ is also a firing sequence at $\hat{M}$ with $\hat{M}[\sigma' > \hat{M}'$.

Finally let $\hat{M}[\sigma_1 w_1 w_2 > \hat{M}^2 [\sigma_{12} \sigma_2 > \hat{M}'$. Then it is straightforward to verify that $M[\sigma_1 w > \hat{M}^2$ in S also. The required contradiction can now be obtained by applying the induction hypothesis at $\hat{M}^2$. ∎

<u>Theorem 5.2</u> Every well formed bp scheme is well behaved.

<u>Proof</u> Every elementary scheme is well behaved. Hence we merely need to prove that the transformation rules preserve well behavedness. To do so, let S be a well behaved scheme and $\hat{S}$ be obtained by applying the rule T_i to S for some $i \in \{1,2,\ldots,8\}$.

<u>i=1,6</u> $\hat{S}$ is obviously well behaved.

<u>i=2</u> By lemma 5.1, $\hat{S}$ is well behaved.

<u>i=3,4</u> That $\hat{S}$ is a bp scheme (i.e. the underlying marked graph of $\hat{S}$ is live and safe) is easy to establish. The proof that $\hat{S}$ is well behaved is very similar to - and a shade messier than - the proof of lemma 5.1 and hence we shall omit it.

<u>i=5,7</u> Again the fact that $\hat{S}$ is a bp scheme is easy to prove. To show well behavedness we follow the proof idea of lemma 5.1 (along much smoother paths this time).

<u>i=8</u> Theorem 4.9 tells us that $\hat{S}$ is a well behaved scheme. ∎

6. THE COMPLETENESS OF THE SYNTEHSIS PROCEDURE

As the title of the section suggests, the aim here is to show that the synthesis technique introduced in the previous section yields all well behaved schemes. The proof is somewhat involved and we shall do our best to chop it up into digestable pieces. We will start with a set of reduction rules using which one can "parse" bp schemes.

6.1 The Reduction Rules

As in the presentation of the synthesis rules we specify the reduction rules only graphically. There are six rules on the whole. We denote the source scheme to which the reduction rule R_i is to be applied by $S=(V_\nabla,V_\&,A;Q,Z,M)$ and resulting target scheme by $\hat{S} = (\hat{V}_\nabla,\hat{V}_\&,\hat{A};\hat{Q},\hat{Z},\hat{M})$. As before, the specification of each rule will consist of:

1) The restrictions on M, the marking of S at which the rule is to be applied.

2) The restrictions on the structure of S that must be met for the rule to be applicable.

3) The changes in the structure effected by the application of the rule.

4) The specification of $\hat{M}$, the marking of $\hat{S}$.

The first and fourth parts of the specification is common to the rules R_1 through R_5. So we shall put them down first.

The marking M must satisfy: If an arc appears in S but not in $\hat{S}$ then such an arc should not be marked at .

The new marking $\hat{M}$ of $\hat{S}$ is obtained as follows: If an arc appears in S and $\hat{S}$ then it is marked in the same way at $\hat{S}$ as it was marked at M. If an arc appears in $\hat{S}$ but is not in S, then it is left unmarked at $\hat{M}$. The first two reduction rules are renamed versions of two transformation rules.

$\underline{R_1}$ (Arc Reduction) Same as the transformation rule T_6.

$\underline{R_2}$ (Node Reduction) Same as the transformation rule T_7.

$\underline{R_3}$ (X Reduction)

Fig. 6.1

This is the "reverse" of a generalised version T_5. The restrictions are:

The environments of v and u in S are no more than what is shown in fig. 6.1. In other words $^{\bullet}v = \{a_1,a_2,\ldots,a_m\}$, $v^{\bullet} = \{c\} = {}^{\bullet}u$ and $u^{\bullet} = \{b_1,b_2,\ldots,b_n\}$.

$\underline{R_4}$ (∇-diamond Reduction)

Fig. 6.2

The ∇-node v_1 is split into two ∇-nodes v_1' and v_1'' ; a new &-node u is inserted in between through the addition of the arcs b and b' as shown. The key part of the reduction is to replace the arcs $a_1,a_2,\ldots,a_n$ by a single arc a. The restrictions are:

At the marking M in addition to the general restriction stated at the beginning, no arc in $\{a_1,b_1,a_2,b_2,\ldots,a_n,b_n\}$ should be marked. And as indicated in the diagram, for $1 \leqq i \leqq n$, $|u_i^{\bullet}| \geqq 2$. Finally, the environment of the ∇-node v_1 is no more than what is shown in fig. 6.2.

This rule is the "reverse" of a generalised version of T_3.

$\underline{R_5}$ (&-diamond Reduction)

Fig. 6.3

This is the reverse dual of R_4. The restrictions are:

At M, in addition to the common restriction, no arc in $\{a_1,b_1,a_2,b_2,\ldots,a_n,b_n\}$ should be marked. And as indicated, for $1 \leqq i \leqq n$, $|{}^{\bullet}v_i| \geqq 2$. Finally, the environment of the &-node u_1 in S is no more than what is shown in fig. 6.3.

$\underline{R_6}$ (Marking Transformation) Same as the transformation rule T_8.

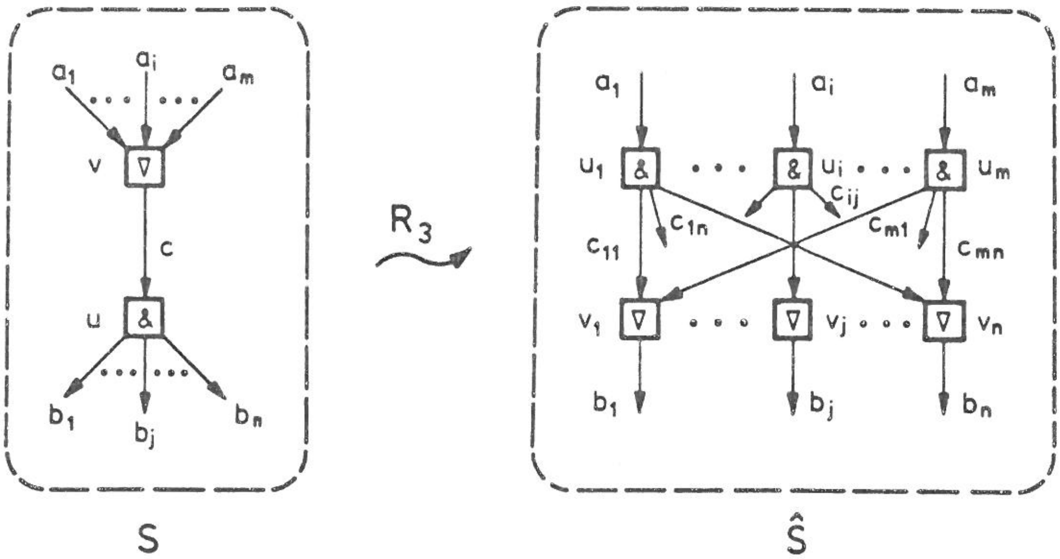

a1
ai
am
v
c
u
b1
bj
bn
S
R3
a1
ai
am
u1
ui
um
cij
c1n
cm1
c11
cmn
v1
vj
vn
b1
bj
bn
Ŝ

Fig. 6.1

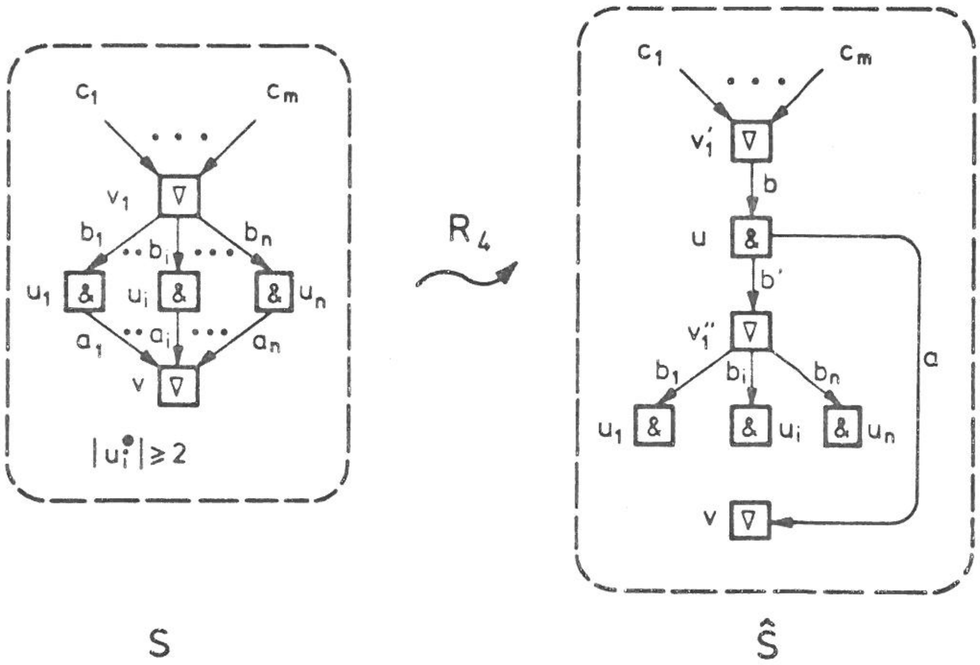

c1
cm
v1
b1
bi
bn
u1
ui
un
a1
ai
an
v
|ui•| ≥ 2
S
R4
c1
cm
v1'
b
u
b'
v1''
a
b1
bi
bn
u1
ui
un
v
Ŝ

Fig. 6.2

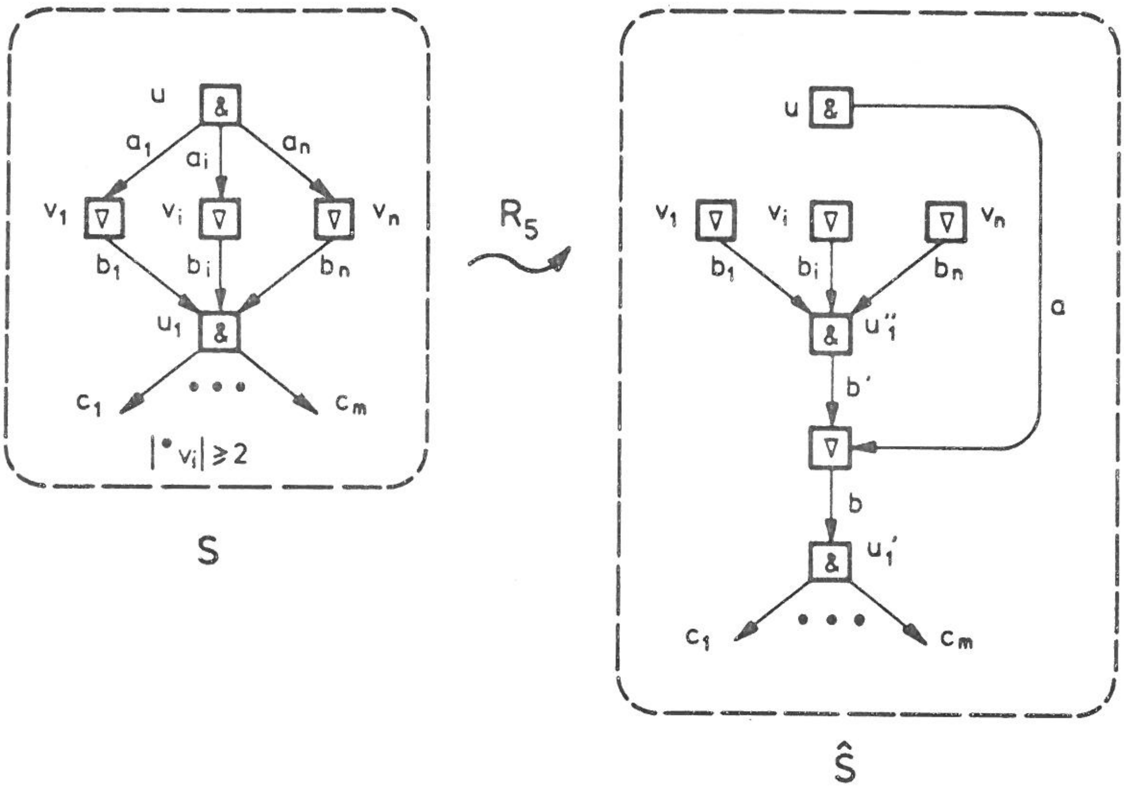

u
&
a_1
a_i
a_n
v_1
v_i
v_n
b_1
b_i
b_n
u_1
c_1
c_m
$|^\bullet v_i| \geq 2$
S
R_5
u
v_1
v_i
v_n
b_1
b_i
b_n
a
u_1''
b'
b
u_1'
c_1
c_m
$\hat{S}$

Fig. 6.3

These are all the reduction rules we need to show the completeness of the synthesis technique. To start with we shall show that each reduction rule transports well behavedness from source to target and well formedness from target to source.

Definition 6.1 Let S be a bp scheme.

a) For $1 \leq i \leq 5$, S is said to be R_i-reducible iff the reduction rule R_i can be applied to S.

b) S is said to be reducible iff for some i with $1 \leq i \leq 5$, S is R_i-reducible.

c) S is irreducible iff S is not reducible.

d) $\hat{S}$ is a reduction of S iff $\hat{S}$ can be obtained by applying a reduction rule to S. ∎

Theorem 6.1 Let $\hat{S}$ be a reduction of the well behaved bp scheme S. Then $\hat{S}$ is also a well behaved bp scheme.

Proof By definition, $\hat{S}$ is the result of applying the reduction rule R_i to S for some i in $\{1,2,\ldots,6\}$.

i=1,2,6 We have already dealt with these cases in theorem 5.1.

i=4,5 That $\hat{S}$ is indeed a bp scheme is not difficult to prove. To show that $\hat{S}$ is well behaved we can borrow the proof technique of lemma 5.1 to simulate $\hat{S}$ with S; in the present two cases without any "delay" even. The required result then follows easily.

i=3 If $\hat{S}$ is a bp scheme then once again through some straightforward simulation we can show that $\hat{S}$ is well behaved because S is well behaved. The part that requires an argument is that $\hat{S}$ is a bp scheme, i.e. the underlying marked graph of $\hat{S}$ is live and safe.

There is no trouble about verifying liveness. To show safety we must prove that each of the new arcs in $\{c_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$

is contained in a basic circuit. To do so, it is sufficient to prove that in $\tilde{S}$, the underlying marked graph of S, there is a basic circuit containing a_i and b_j for $1 \leq i \leq m$ and $1 \leq j \leq n$.

Let i,j be such that $1 \leq i \leq m$ and $1 \leq j \leq n$. While $\tilde{S}$ is live and safe there is a basic circuit π passing through a_i (in $\tilde{S}$). Since $v^{\bullet} = {}^{\bullet}u = \{c\}$, (see fig. 6.1) π must contain an output arc of u, say b_k where $1 \leq k \leq m$. Similarly let π' be a basic circuit of $\tilde{S}$ containing b_j and which of necessity must pass through some input arc of v, say a_l. Without loss of generality, let us assume that π and π' originate from u. Now π and π' can be expressed as $\pi = \pi_1 \pi_2$, $\pi' = \pi_1' \pi_2'$ such that π_1 and π_1' are node-disjoint except for the initial and terminal nodes (see fig. 6.4).

Fig. 6.4

Suppose that π_1 and π_1' have the ∇-node v as their terminal nodes. In other words π and π' meet at v for the first time after departing from u. Then starting from a marking at which u has just H-fired we can apply Prop. 2.6 to conclude that S is not well behaved which is a contradiction. Hence π and π' must meet earlier than v. This implies that $\pi_1' \pi_2$ is also a basic circuit (recall that $\tilde{S}$ is live) and it passes through both a_i and b_j. ∎

<u>Theorem 6.2</u> Let S be a reduction of the bp scheme S. If $\hat{S}$ is well formed then S is also well formed.

<u>Proof</u> Let $\hat{S}$ be obtained by applying the reduction rule R_i to S where $1 \leq i \leq 6$.

<u>i=1</u> We can apply T_1 to $\hat{S}$ to get back to S.

<u>i=2</u> We can find a suitable application of T_2 to $\hat{S}$ to get S. The required node splitting would be permitted because $\tilde{S}$ is known to be safe.

<u>i=3</u> We have shown in Fig. 6.5 for the concrete case m=3, n=2 how S can be obtained from $\hat{S}$ through a sequence of applications of the transformation rules. An inductive argument based on this idea can be easily constructed.

Fig. 6.5

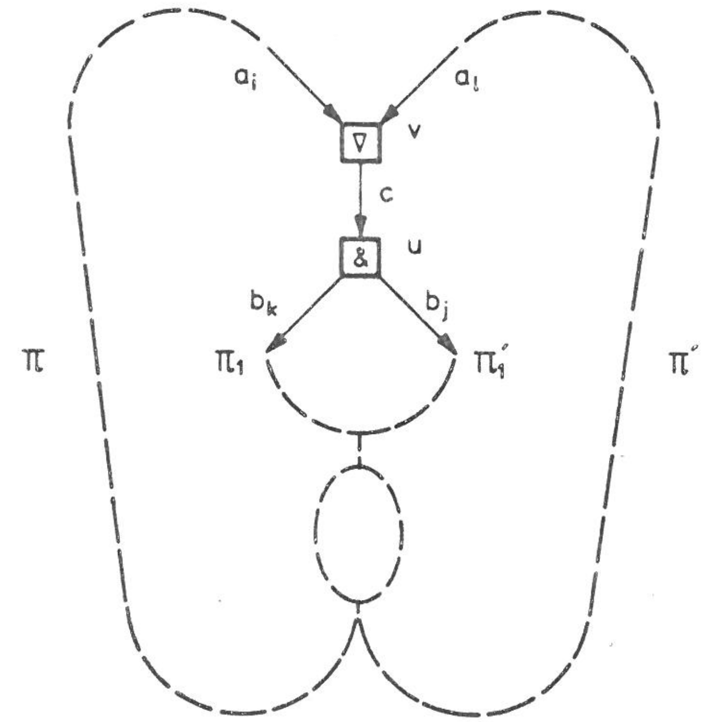


Fig. 6.4

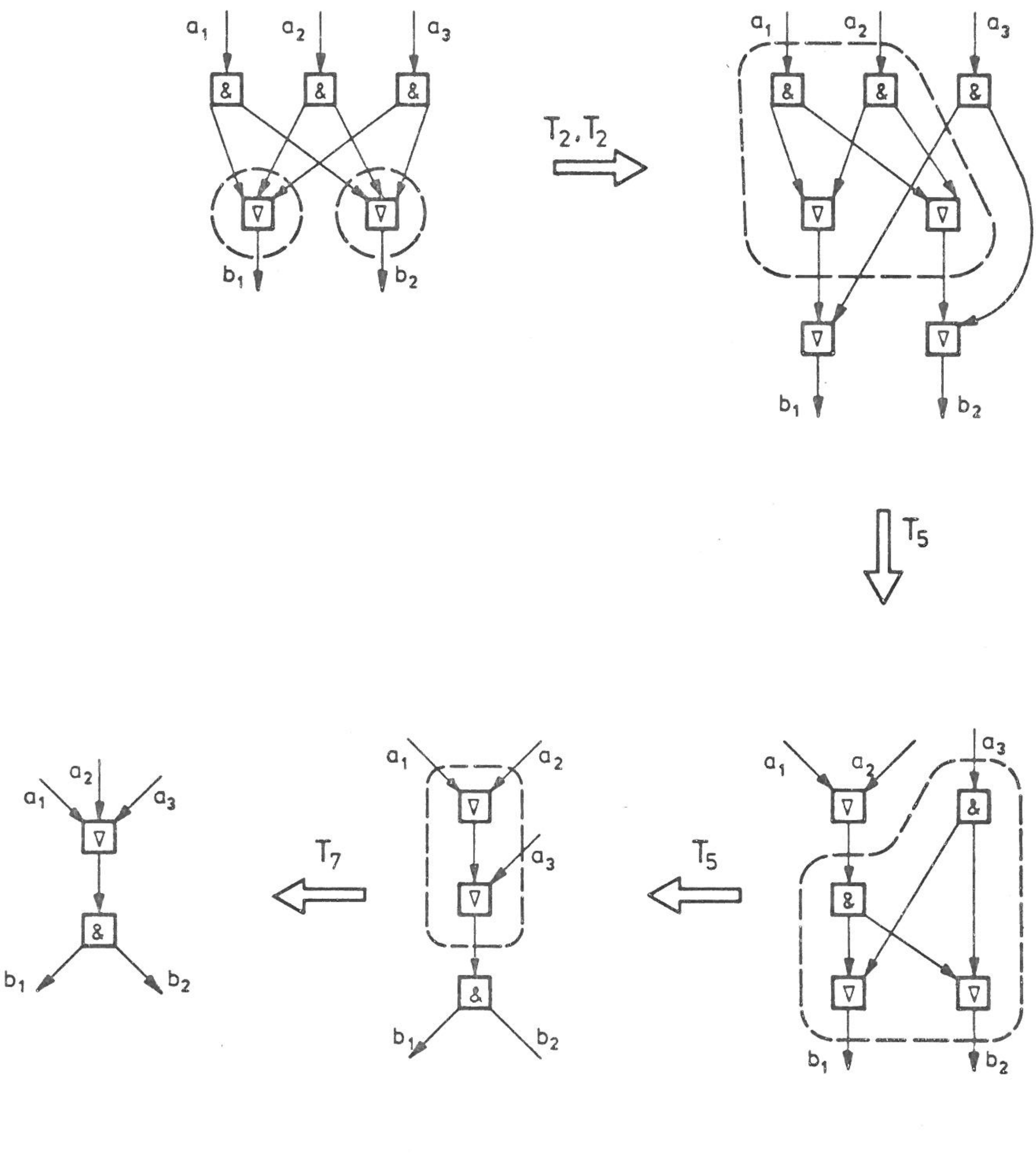


Fig. 6.5

i=4 Once again we have illustrated, in fig. 6.6, the proof idea through a concrete example.

Fig. 6.6

i=5 Similar to that of the previous case.

i=6 Trivial. ∎

The above result suggests a possible way of proving that every well behaved scheme is well formed. It is sufficient to show that every well behaved scheme can be - by repeatedly applying the reduction rules - reduced to an elementary bp scheme. Motivated by this we shall first show that starting from a well behaved scheme, if the reduction process terminates then the resulting scheme is elementary. Afterwards we will show that there is one "standard" way to do the reduction which always terminates.

6.2 Irreducible Well Behaved Schemes are Elementary

The first result is interesting in its own right.

Theorem 6.3 Let S be a bp scheme. If the supporting bp graph of S has one of the three structures shown in fig. 6.7 as a (bp) subgraph (see def. 3.9), then S is not well behaved.

Fig. 6.7

Proof

(A) Assume that fig 6.7.A is a subgraph of S. Let M^1 be a v-extremal marking in the forward marking class of S. If no such marking exists then S is not well behaved and we are done. There are two situations to consider at M^1.

Suppose there is no token-free path of length greater than one from v to u at M^1. If $Q(c)=v$ then by H-firing v at M^1 so that c or b gets an H-token, we can create a deadlock at u. If $Q(c)\neq v$ then starting from M^1, without firing v, we can obtain a marking M^2 at which c carries a token. If $c \in M^2_H$ $\langle M^2_L \rangle$ then by H-firing v at M^2 so that b gets an L-token $\langle$H-token$\rangle$ we can create a deadlock at u.

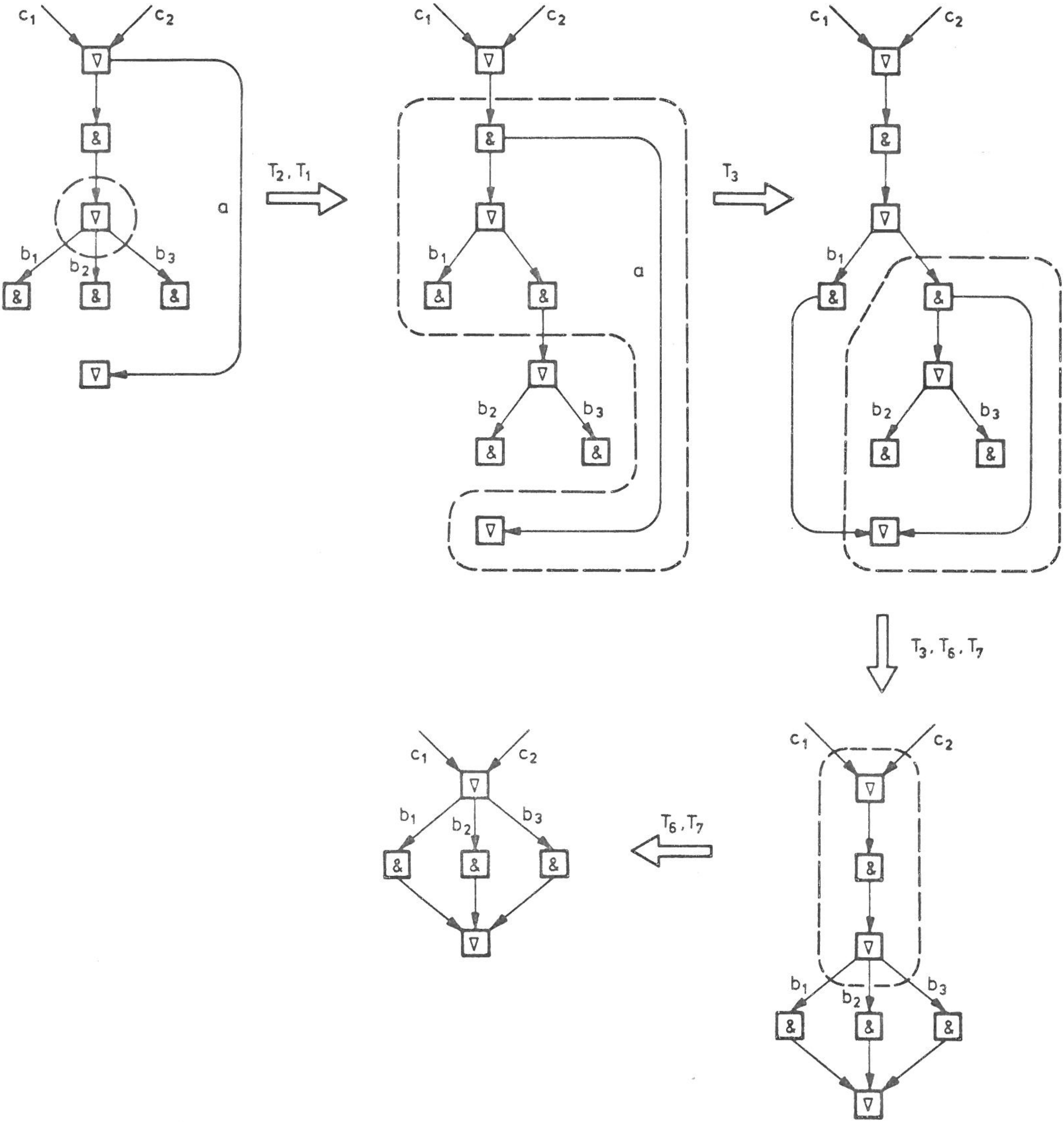


Fig. 6.6

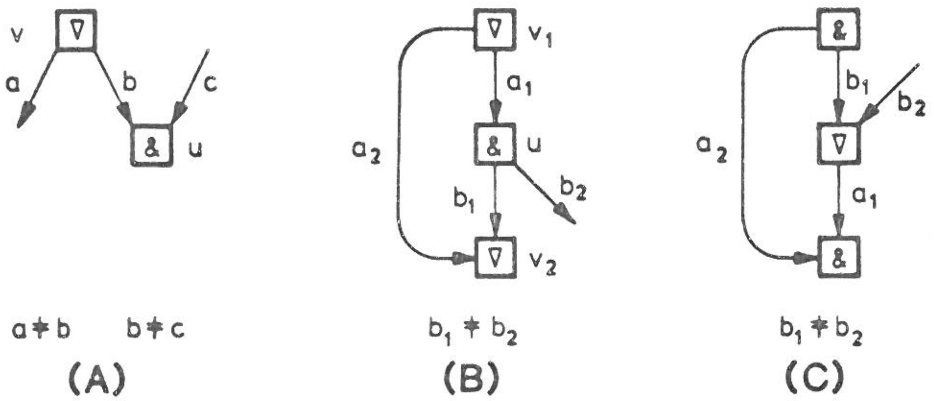


Fig. 6.7

So suppose indeed π is a token-free path at M^1 of length greater than one from v to u. Without loss of generality assume that the first arc of π is a and that the last arc of π is c. Let $M^1[v>M^2$ such that $a \in M^2_H$ and hence $b \in M^2_L$. Starting from M^2, without firing u, we can guide the H-token on a along π to reach a marking M^3 at which $b \in M^3_L$ and $c \in M^3_H$. In other words, u will be in deadlock at M^3.

(B) Assume that fig. 6.7.B is a subgraph of S. If S is to be well behaved then every arc must be contained in a ∇-component of S (theorem 3.9). If the arc b_2 (see fig. 6.7.B) is contained in a ∇-component of the form $S^1 = (V^1_\nabla, V^1_\&, A^1; Q^1, Z^1, M^1)$ then $a_1 \in A^1$. If not $|{}^\bullet u|>1$ and by part A) of this theorem, S is not well behaved. But $Q(a_1) \in V^1_\nabla$ implies that $v_1^\bullet \subseteq A^1$ so that $a_2 \in A^1$ and $Z(a_2) \in V^1_\nabla$. This once again implies that ${}^\bullet v_2 \subseteq A^1$ so that $b_1 \in A^1$. But $b_1, b_2 \in A^1$ so that $|u^\bullet \cap A^1|>1$ contradicts the assumption that S^1 is a ∇-component; a_2 cannot be contained in any ∇-component of S; S cannot be well behaved.

(C) Assume that fig. 6.7.C is a subgraph of S. Then it is easy to verify that the arc b_2 is not contained in any &-component of S. Once again by theorem 3.9 we can conclude that S is not well behaved. ∎

Using the above result we can now demonstrate that if S is a well behaved scheme which is not elementary and to which R_1, R_2 and R_3 cannot be applied, then R_4 or R_5 can be applied to S. First we shall demonstrate that a "partially" irreducible well behaved scheme must contain a particular kind of subgraph.

<u>Lemma 6.4</u> Let S be a well behaved scheme which is not elementary and to which the reduction rules R_1, R_2 and R_3 cannot be applied. Then S has a basic circuit of maximum length of the form $\pi = a_1 a_2 a_3 \ldots a_n$ such that (see fig. 6.8):

Fig. 6.8

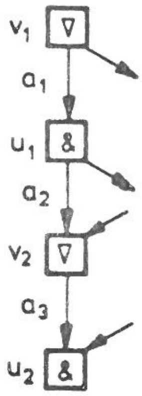


Fig. 6.8

1) $|\pi| \geq 3$

2) $Q(a_1)=v_1,\ Q(a_3)=v_2 \in V_\nabla$;

$Z(a_1)=u_1,\ Z(a_3)=u_2 \in V_\&$.

3) $|v_1^\bullet|,\ |{}^\bullet v_2|,\ |u_1^\bullet|,\ |{}^\bullet u_2|>1;\ |{}^\bullet u_1|=|v_2^\bullet|=1$.

Proof Let π be a basic circuit of maximum length in S. Since R_2 cannot be applied to S, for every arc b contained in π we must have that Q(b) and Z(b) are of different types. Now the required result follows by appealing to the facts that R_1 and R_3 cannot be applied to S; fig. 6.7.A cannot be a subgraph of S; and $\tilde{S}$ is a live and safe marked graph. ∎

It will be convenient to go through one more intermediate step before we get to the result we are actually after.

Lemma 6.5 Let S be a well behaved scheme which is not elementary and to which the reduction rules R_1, R_2 and R_3 cannot be applied. Let the structure shown in fig. 6.8 be a subgraph of S where the arcs a_1,a_2 and a_3 are contained in a basic circuit of maximum length. Let M1 be a v_1-extremal marking. Then,

1) For every $x \in v_1^\bullet$ there is a token-free path at M^1 from v_1 to v_2 which contains x.

or

2) For every $y \in {}^\bullet u_2$, at M^1, there is a token-free path from u_1 to u_2 which contains y.

Proof Suppose that $x \in v_1^\bullet$ and $y \in {}^\bullet u_2$ such that at M^1 there is no token-free path from v_1 to v_2 containing x and there is no token-free path from u_1 to u_2 containing y. We shall derive the contradiction that S is not well behaved.

Let $M^1[v>M^2$ and $M^1[v>M^3$ such that $a_1 \in M_H^2$ - which implies $x \in M_L^2$ - and $x \in M_H^3$, which implies $a \in M_L^3$. The simple but crucial observation to keep in mind is that

$$M_H^2 \cup M_L^2 = M_H^3 \cup M_L^3 .$$

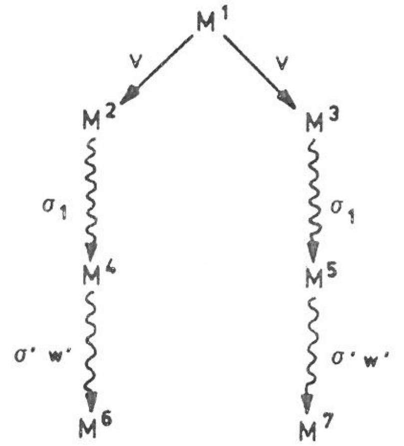


Fig. 6.9

Fig. 6.9

Consider a minimal v_2-enabling sequence σ at M^2. Let $Z(x)=w$. Then because we are assuming that there is no token-free path from v_1 to v_2 which contains x, we can conclude that $|\sigma|_w=0$. At M^1 and hence at M^2 and M^3, there cannot be any token-free path from u_1 to v_2 other than the one provided by the arc a_2, (prop. 2.6). Hence we can assume without loss of generality that u_1 fires right at the end of σ. In other words $\sigma = \sigma_1 u_1$. Let $M^2[\sigma_1 > M^4$ and M^5 be the marking obtained by firing σ_1 at M^3 in the same way in which it was fired at M^2 (see fig. 6.9). This is possible because $\tilde{M}^2=\tilde{M}^3$ and S is well behaved. By construction of M^4 and M^5 we have $\tilde{M}^4=\tilde{M}^5$ $a_1 \in M_H^4$, $x \in M_H^5$ (recall $|\sigma|_w=0$). Indeed the only difference between M^4 and M^5 is the way in which the two arcs a_1 and x are marked. While $\sigma = \sigma_1 u_1$ is a v_2-enabling sequence we know that every input arc of v_2 other than a_2 is marked at M^4 and M^5; moreover they are marked in the same way at M^4 and M^5. We claim that $^\bullet v_2 - \{a_2\} \subseteq M_L^4$ and $^\bullet v_2 - \{a_2\} \subseteq M_L^5$.

To see this we merely need to observe that if some arc in $^\bullet v_2 - \{a_2\}$ carries an H-token at M^5 and hence at M^4, then at the marking obtained by H-firing u_1 at M^4, the ∇-node v_2 will be in deadlock. This proves the claim.

We can now start concentrating on the arc y. Suppose that $y \in M_H^4$. Then $y \in M_H^5$ also. At M^5, we can L-fire u_1 followed by an L-firing of v_2 to create a deadlock at u_2. If $y \in M_L^4$ that at M^4, we can H-fire u_1, followed by an H-firing of v_2 to create a deadlock at u_2.

Hence assume that the arc y is not marked at M^4 and M^5. Let $Q(y)=w'$ and σ' a minimal w'-enabling sequence at M^4. Let $M^4[\sigma'w'>M^6$ and M^7 be the marking obtained by firing $\sigma'w'$ at M^5 in the same way as it was fired in going from M^4 to M^6. Because we are assuming that at M^1 there is no token-free path from u_1 to u_2 containing y, it is also true at M^4 and M^5. Hence $|\sigma'|_{u_1}=0$. Consequently, $|\sigma'|_{v_2}=0$ also and it is easy to verify the following details:

$$a_1 \in M_H^6;\ a_1 \in M_L^7;\ ^\bullet v_2 - \{a_2\} \subseteq M_L^6;\ ^\bullet v_2 - \{a_2\} \subseteq M_L^7 .$$

Since y is marked at M^6 and it is marked in the same way at M^6 and M^7 we need consider two cases. Let $y \in M^6_H$. Then at M^7 we can L-fire u_1 followed by an L-firing of v_2 to produce a deadlock at u_2. If $y \in M^6_L$, then at M^6 we can H-fire u_1 followed by a H-firing of v_2 to create a deadlock at u_2. Hence our original assumption must be false. ∎

Theorem 6.6 Let S be a well behaved irreducible bp scheme. Then S is elementary.

Proof Suppose that S is well behaved and the reduction rules R_1, R_2 and R_3 cannot be applied to S. Assume that S is not elementary. Then we need to show that R_4 or R_5 can be applied to S.

To this end we first note that by lemma 6.4, there is a basic circuit of maximum length of the form $\pi = \{a_1, a_2, a_3 \ldots a_n\}$ which fulfills the conditions laid out by that lemma (see fig. 6.8).

Using lemma 6.5, let us first suppose that for every $x \in v_1^\bullet$ there is a token-free path from v_1 to v_2 containing x at the v_1-extremal marking M^1. Let $x \in v_1^\bullet - \{a_1\}$ and π' be such a token-free path. If $|\pi'|>2$ then π cannot be a basic circuit of maximum length. If $|\pi'|=1$, then fig. 6.7.B would be a subgraph of S which contradicts the well behavedness of S (theorem 6.3). Hence $|\pi'|=2$. Z(x) must be a &-node because R_2 is not applicable to S and π is a basic circuit of maximum length. If $|{}^\bullet Z(x)|>1$ then once again S cannot be well behaved because fig. 6.7.A would be a subgraph S. Hence ${}^\bullet Z(x) = \{x\}$. If $|Z(x)^\bullet|=1$ then R_1 would be applicable to S. Consequently, $|Z(x)^\bullet|>1$. Since this is true of every output arc of v_1, we now have that R_4 is applicable to S.

If on the other hand, at M^1, for every $y \in {}^\bullet u_2$, there is a token-free path from u_1 to u_2 containing y then by a similar set of arguments we can conclude that R_5 is applicable to S. ∎

6.3 A characterization of *SWF*

As a run up to the main result we will now obtain a behavioural characterisation of *SWF*.

Definition 6.2   Let $S = (V_{\nabla}, V_{\&}, A; Q, Z, M)$ be a bp scheme. Then the reverse of S is denoted as S^R and is given by $S^R = (V_{\nabla}, V_{\&}, A; Q^R, Z^R, M)$ where $Q^R = Z$ and $Z^R = Q$. ∎

The reverse of a bp scheme is also a bp scheme. This is because from theorem 1.3 we have that the marked graph $MG = (V, A; Q, Z, M)$ is live and safe iff the marked graph $MG^R = (V, A; Q, Z, M)$ is also live and safe.

Definition 6.3   A bp scheme is said to be strongly well behaved iff both S and its reverse are well behaved. ∎

The bp scheme shown in fig. 2.3 is strongly well behaved but not the one shown in fig. 2.5.a. It turns out *SWF* is completely characterised by this property.

Lemma 6.7   Let S be a strongly well behaved scheme and $\hat{S}$ be obtained by applying T_1 or T_2 to S. Then $\hat{S}$ is also a strongly well behaved scheme.

Proof   We first observe that if the scheme $\hat{S}$ can be obtained from S through T_1 <T_2> then the reverse of $\hat{S}$ can be obtained from the reverse of S through T_1 <T_2>. This follows once again from the characterisation of live and safe marked graphs and the definitions. The required result can now be derived using lemma 5.1. ∎

Lemma 6.8   Let S be a strongly well behaved scheme and $\hat{S}$ be obtained by applying the reduction rule R_1 or R_2 to S. Then $\hat{S}$ is also strongly well behaved. Moreover if $\hat{S}$ is strongly well formed, then so is S.

Proof   Follows at once from theorem 6.2 and the definitions. ∎

Theorem 6.9   A bp scheme is strongly well formed iff it is strongly well behaved.

Proof Let S be strongly well formed. Since every elementary scheme is strongly well behaved, we can conclude from lemma 6.7 that S is also strongly well behaved.

Assume that S is a strongly well behaved scheme. Let $S_0, S_1, \ldots$ be a sequence of bp schemes of maximum length such that $S=S_0$; for $i \geq 0$, S_{i+1} is obtained by applying R_1 or R_2 to S_i. Now the number of nodes in the target scheme is strictly less than the number of nodes in the source scheme whenever we apply R_1 or R_2 to the source scheme. Consequently the sequence of bp schemes described above must be of finite length. It must terminate with the bp scheme S_n which is strongly well behaved by lemma 6.8. If S_n is elementary then again by lemma 6.8, $S_0=S$ is strongly well formed and we are done.

So assume that S_n is not elementary. Let π be a basic circuit of maximum length in $\tilde{S}_n$. While S_n is not elementary $|\pi|>1$. By construction, neither R_1 nor R_2 can be applied to S_n. We know (theorem 6.3) that neither fig. 6.7.A nor its "reverse" (the one obtained by reversing all arcs) can be a subgraph of the strongly well behaved scheme S_n. Consequently, the situation along π must be one of the two shown in fig. 6.10. In both cases we are led to the contradiction that $\tilde{S}_n$ is not live and safe. Hence S_n must indeed be elementary so that S is strongly well formed. ∎

Fig. 6.10

One of the consequences of this somewhat surprising and pleasing result is that we can check whether a scheme is in *SWF* by firing the nodes systematically forwards and backwards to verify good behaviour; to check whether a scheme is strongly well behaved, we can repeatedly apply R_1 and R_2 and see whether we end up with an elementary scheme. It is easy to see now that the scheme of fig. 2.5.a is not in *SWF*.

What makes it difficult to prove a similar result for the larger class *WF* is that the number of nodes does not decrease strictly for each application of R_3, R_4 and R_5. So there is no guarantee that repeated applications of the reduction rules to a well behaved scheme will ever result in an irreducible scheme. What we need is a more complicated measure and a more sophisticated reduction procedure.

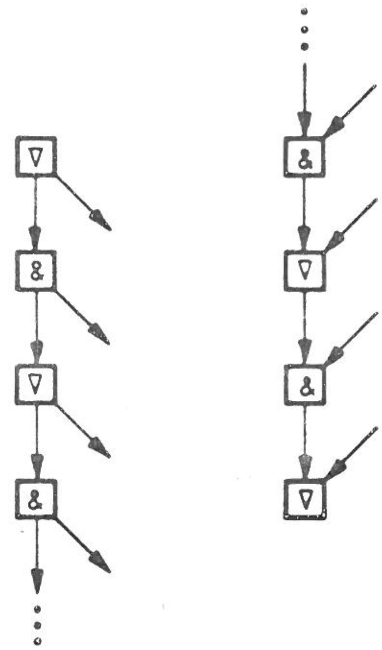


Fig. 6.10

6.4 The Reduction Macros and the Size Factor

We shall first build up some reduction macros by stringing together the reduction rules in appropriate ways. The motivation for constructing these macros will become clear once we define a measure called the size factor. It will turn out that for each application of a reduction macro, the size factor either remains constant or (at least for two of the macros) strictly decreases. This will then enable us to write a reduction procedure which is provably convergent. There are essentially three macros but for the sake of uniformity we shall simply rename two of the reduction rules.

$\underline{RM_1}$ Same as R_1.

$\underline{RM_2}$ Same as R_2.

$\underline{RM_3}$ Apply R_3 to the source scheme S to obtain the target scheme $\hat{S}$. Starting from $\hat{S}$, apply R_2 as often as possible in such a way that in each application (of R_2) at least one node is involved which is in $\hat{S}$ but not in S.

$\underline{RM_4}$ $<RM_5>$ Apply R_4 $<R_5>$ to the source scheme S to obtain the target scheme $\hat{S}$. If possible, apply RM_3 once to $\hat{S}$.

To be precise, we must also include R_6. But from now on, for the sake of convenience, we assume that R_6 comes "free". We shall assume that, whenever necessary, the marking of the scheme that we are working with is replaced by a suitable marking taken from the full marking class of the scheme. Before we proceed to the size factor it is useful to observe:

Lemma 6.10 Let S be a bp scheme and S' be the result of applying RM_i to S where $1 \leq i \leq 5$. If S is well behaved, then S' is also well behaved. If S' is well formed then S is also well formed.

Proof Follows at once from theorems 6.1 and 6.2. ∎

Definition 6.4 Let $S = (V_\nabla, V_\&, A; Q, Z, M)$ be a bp scheme. Then

1) NF(S) denotes the node factor of S and is defined as $NF(S) = |V_\nabla| + |V_\&|$.

2) PF(S) denotes the parity factor of S and is defined as $PF(S) = |\{(v_1, v_2) \in V \mid v_1 \neq v_2, v_1^\bullet \cap {}^\bullet v_2 \neq \emptyset, v_1 \in V_\nabla \Leftrightarrow v_2 \in V_\nabla\}|$.

3) SF(S) denotes the size factor of S and is defined as $SF(S) = NF(S) + PF(S)$. ∎

Roughly speaking, the parity factor counts up the number of pairs of adjacent nodes that are distinct from each other but are of the same type. We will now verify that the size factor decreases monotonically for each application of the reduction macros.

Lemma 6.11 Let S be a bp scheme.
a) Let $\hat{S}$ be obtained by applying RM_1 to S. Then $SF(\hat{S}) \leq SF(S)$.

b) Let $\hat{S}$ be obtained by applying RM_2 to S. Then $SF(\hat{S}) = SF(S)-2$.

Proof Follows easily from the definitions. ∎

Lemma 6.12 Let S be a bp scheme and S' be obtained by applying RM_3 to S. Then $SF(S') \leq SF(S)-2$.

Proof We shall make use of the notations shown in fig. 6.1 to develop the proof. Let $\hat{S}$ be the result of applying R_3 to S. Then,

$$NF(\hat{S}) = NF(S) + m + n - 2.$$

Now consider $a_i \in {}^\bullet v$. If $Q(a_i)$ is a ∇-node in S - and hence in $\hat{S}$ - then the pair $(Q(a_i), v)$ which contributes to SF(S) is not present in $\hat{S}$. If on the other hand $Q(a_i)$ is a &-node then the pair $(Q(a_i),u_i)$ will contribute to $PF(\hat{S})$ whereas this pair does not exist in S. Let

$$m_1 = |\{a \in {}^{\bullet}v \mid Q(a) \in V_{\nabla}\}| \text{ and}$$
$$n_1 = |\{b \in u^{\bullet} \mid Z(b) \in V_{\&}\}|$$

Then it is easy to see that

$$PF(\hat{S}) = PF(S) + (m-m_1) + (n-n_1) - m_1 - n_1.$$

Thus,

$$SF(\hat{S}) = SF(S) + 2(m+n) - 2(m_1+n_1) - 2.$$

Let $a_i \in {}^{\bullet}v$ such that $Q(a_i)$ is a &-node. By construction the only input arc of u_i in $\hat{S}$ is a_i. Hence R_2, involving $Q(a_i)$ and u_i can be applied to $\hat{S}$ (subject to some change in the marking). The node u_i is in $\hat{S}$ but not in S. Hence this is a reduction which is permitted by RM_3. By the previous lemma the size factor will go down by two. But then we can apply R_2 at least $(m-m_1)+(n-n_1)$ times to $\hat{S}$ as part of the execution of RM_3. Consequently,

$$SF(S') \leqq SF(\hat{S}) - 2(m-m_1) - 2(n-n_1).$$

Substituting the expression for $SF(\hat{S})$ that we have derived above (in terms of SF(S)), we get,

$$SF(S') \leqq SF(S) - 2.$$ ∎

<u>Lemma 6.13</u> Let S be a bp scheme and S' be the result of applying RM_4 or RM_5 to S. Then $SF(S') \leqq SF(S)$.

<u>Proof</u> Let S' be obtained by applying RM_4 to S. Once again we shall make use of fig. 6.2 to develop the proof. Let $\hat{S}$ be the result of applying R_4 to S. Then it is easy to verify that $NF(\hat{S}) = NF(S) + 2$ and $PF(\hat{S}) = PF(S)$ so that $SF(\hat{S}) = SF(S) + 2$.

Now an application of RM_3 involving v_1' and u becomes enabled for $\hat{S}$. But then by the previous lemma, $SF(S') \leq SF(\hat{S})-2$.

The proof for RM_5 is similar and we shall omit it. ∎

Thus in a reduction procedure which involves the repeated applications of the reduction macros, RM_2 and RM_3 can be each applied at most a bounded number of times. To ensure that a reduction procedure involving only RM_1, RM_4 and RM_5 will always terminate, we need the following.

<u>Lemma 6.14</u> Let S' be the result of applying RM_4 or RM_5 to the bp scheme S. Then the number of elementary circuits in S' is strictly less than the number of elementary circuits in S.

<u>Proof</u> As usual we will give just one half of the proof. Let S' be the result of applying RM_4 to S. Let $\hat{S}$ be the scheme obtained by applying R_4 (the first step in RM_5) to S. Referring to fig. 6.2, we first note that the path $b_i x$ in S where $x \in u_i - \{a_i\}$ can be simulated by the path $bb'b_i x$ in $\hat{S}$. The paths $b_1 a_1, b_2 a_2 \ldots b_n a_n$ can all be simulated by the single path ba. If a path in S does not touch any of the &-nodes u_1 through u_n then it will also be a path in $\hat{S}$. Using these facts, it is straightforward (but somewhat laborious) to prove that $\hat{S}$ has strictly less elementary circuits than S does.

As noted in the proof of the previous lemma, RM_3 will be applicable to $\hat{S}$. Let S^1 be the result of applying R_3 (as the first step in RM_3) to $\hat{S}$. Then it is easy to verify that S^1 does not have anymore elementary circuits than $\hat{S}$ does. Consider the process of going from S^1 to S' through a number of applications of R_2 (the rest of RM_3). Let w_1 and w_2 be a pair of nodes collapsed together with $w_1^{\bullet} \cap {}^{\bullet}w_2$ eliminated in one such application of R_2. By the specification of RM_3, w_1 or w_2 must be in S^1 but not in $\hat{S}$. From the proof of lemma 6.12 it is clear that if w_1 $\langle w_2 \rangle$ is in S^1 but not in $\hat{S}$, then $|w_1^{\bullet}| \langle |{}^{\bullet}w_2| \rangle = 1$. In either case, through the application of R_2 the number of elementary circuits does not increase. In other words, the number of elementary circuits does not increase in <u>any</u> node reduction performed (though in general, this is certainly not true) in going from $\hat{S}$ to S'. And as we have already seen, $\hat{S}$ has strictly less elementary circuits than S does. ∎

6.5 A Characterisation of *WF*

We can at last establish the completeness of our synthesis procedure. All that is lacking at this stage is a reduction procedure that is guaranteed to terminate. The algorithm given below is one such.

<u>Reduction Algorithm</u>

<u>begin</u>
   <u>input</u> S, a well behaved bp scheme
   $i \leftarrow 0$; $S_i \leftarrow S$.
   <u>do</u> <u>while</u> (S_i is reducible)
      <u>do</u> <u>while</u> (S_i is R_1-reducible)
         let S_{i+1} be the result of applying R_1 to S_i;
         $i \leftarrow i+1$;
      <u>od</u>
      <u>do</u> <u>while</u> (S_i is R_2-reducible or R_3-reducible)
         let S_{i+1} be the result applying RM_2 or RM_3 to S_i;
         $i \leftarrow i+1$;
      <u>od</u>

```
      if S_i is R_4-reducible or R_5-reducible
         then let S_{i+1} be the result of applying RM_4 or RM_5
              to S_i;  i←i+1
         else skip
      fi
   od
   S'←S_i; output S'
end
```

Theorem 6.5 A bp scheme is well behaved iff it is well formed.

Proof Let S be a well behaved bp scheme. To see that S is well formed let us present it as the input to the reduction algorithm given above. Let us assume that the algorithm terminates. Then the output S' is irreducible. It is well behaved by lemma 6.10 and hence is elementary by theorem 6.6. That $S_0=S$ is well formed follows now at once from lemma 6.10. Thus we just need to verify that the reduction algorithm terminates.

First we observe that each application of a reduction macro will certainly terminate. Now consider the first do loop in the algorithm. Each application of RM_1 reduces the number of nodes by one. So that if this loop is entered with S_i having k nodes, then it can execute at most k times. For the second loop, if we enter with S_i having size factor k, then the loop will be executed at most $\lceil k/2 \rceil$ ($\lceil x \rceil$ is the least integer greater than or equal to the rational x) times. This follows from lemmas 6.11 and 6.12. So this loop will also always terminate.

Considering the running of the whole algorithm, the size factor does not increase by the execution of the first loop or by the execution of the if ... fi statement. This follows from the first part of lemma 6.11 and lemma 6.13. As we have already noted, for each iteration of the second loop the size factor goes down by two. Hence if we consider the sequence of schemes generated during the execution of the algorithm $S = S_0, S_1, S_2 \ldots$, then starting with some $j \geq 0$ for every $k \geq j$, S_{k+1} is the result of applying RM_1 or RM_4 or RM_5 to S_k.

The number of elementary circuits does not increase by an application of RM_1 but it strictly decreases by an application of RM_4 or RM_5 (lemma 6.14). Hence if we consider the sequence schemes $S_j, S_{j+1}, \ldots$ (where j is as defined above) then starting with some $l \geq j$ we will have that for every $k \geq l$, S_{k+1} is obtained from S_k by an application of RM_1.

Now the sequence $S_l, S_{l+1}, \ldots$ must certainly terminate because each application of RM_1 as we have already seen reduces the number of nodes by 1. The algorithm indeed terminates.

The second half of the result is theorem 5.2. ∎

Thus our synthesis procedure generates only well behaved schemes and it generates all of them.

7. A COMPUTATIONAL INTERPRETATION

In this section we shall develop a formal interpretation for well behaved schemes. The result will be a flow chart model of a class of distributed computations, or stated differently, a class of concurrent programs.

A program modelled by an interpreted scheme will consist of operations and tests applied to a set of variables. A (distributed) state of a program will have two components: a control state and a value state. A control state is a distribution of commission and omission signals over a set of locations. In other words, we shall view the arcs as locations and refer to the markings of the underlying scheme as control states.

In our model, the tests will be associated with the ∇-nodes and the operations with the &-nodes. An example of an interpreted scheme is shown in fig. 7.1. It will be convenient to postpone explaining what this program does. Actually, as it will turn out, what this program does is not all that interesting. But it does fulfill its role; which is, to serve as a running example to illustrate the various parts of the interpretation that we shall now develop.

Fig. 7.1

In what follows we shall work with the generic well behaved bp scheme $BP = (V_\nabla, V_\&, A; Q, Z, M^0)$. The corresponding interpreted scheme will be denoted by $\overline{BP}$. As mentioned earlier, $[M^0\rangle$ is the set of control states of $\overline{BP}$.

7.1 The Variables and their Allocation to Locations

With $\overline{BP}$ we associate a finite non-empty indexed set of variables

$$X = \{x_1, x_2, \ldots, x_n\}$$

In fig. 7.1, x, y, b_1 and t_1 are some of the variables associated with the program. Each x_i assumes values over the domain D^i. The set of conceivable value states of $\overline{BP}$ is denoted

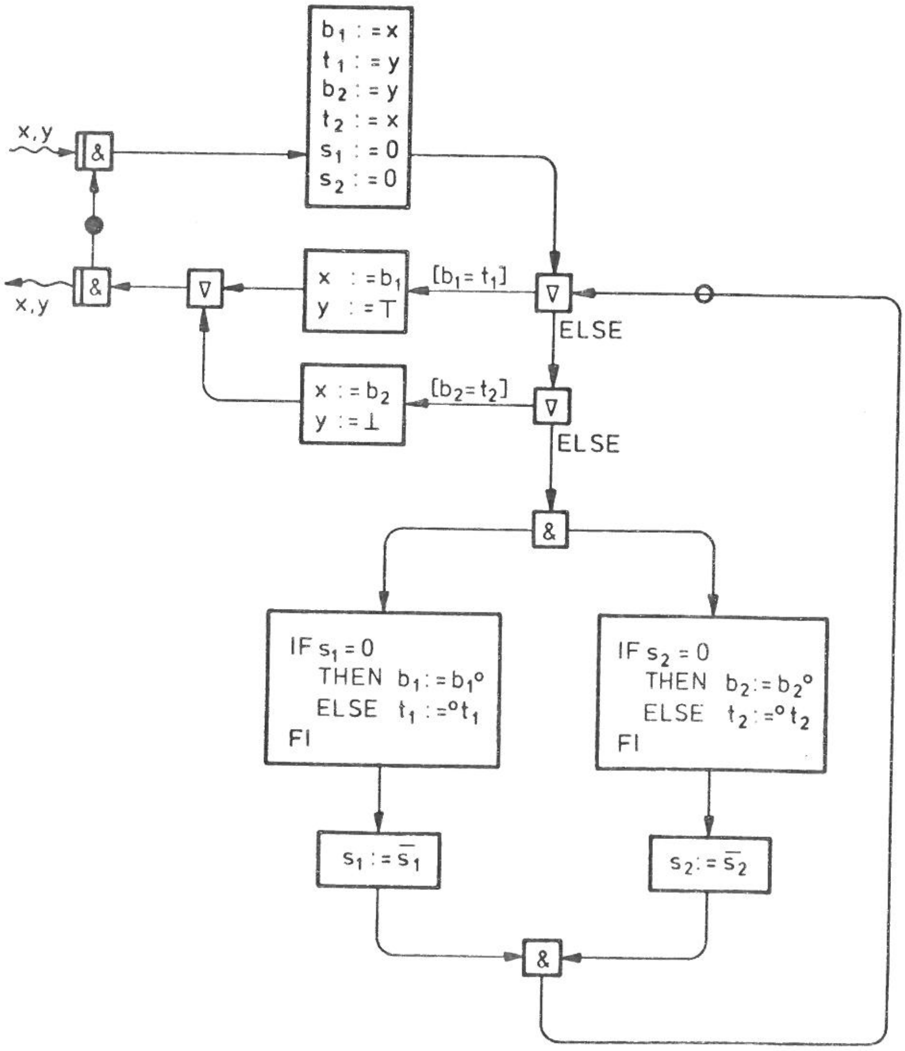

x,y
&
b1 := x
t1 := y
b2 := y
t2 := x
s1 := 0
s2 := 0
x,y
&
∇
x := b1
y := T
[b1 = t1]
∇
ELSE
x := b2
y := ⊥
[b2 = t2]
∇
ELSE
&
IF s1 = 0
THEN b1 := b1°
ELSE t1 := °t1
FI
IF s2 = 0
THEN b2 := b2°
ELSE t2 := °t2
FI
s1 := s̄1
s2 := s̄2
&

Fig. 7.1

as $\mathcal{D}$ and is given by,

$$\mathcal{D} = D^1 \times D^2 \times \ldots \times D^n.$$

Let $\zeta \in \mathcal{D}$ be a value state. Then ζ_i denotes the i'th component of ζ; the value of the variable x_i at this value state. A distinguished value state ζ^0 called the initial value state is assumed to be given.

Combining the control states and the value states we get the states of $\overline{BP}$. More precisely, the set of conceivable states of $\overline{BP}$ is denoted as $\mathcal{K}$ and is given by,

$$\mathcal{K} = [M^0> \times \mathcal{D}$$

The states that $\overline{BP}$ can actually visit during the course of the computation will be called the cases and are denoted as $\mathcal{C}$. As might be expected (M^0,ζ^0) is the state in which the computation starts and is called the initial case. In the next subsection we shall give a formal definition $\mathcal{C}$ after presenting the firing rules for $\overline{BP}$.

We now allocate to each arc $b \in A$ a set of variables $X_b \subseteq X$ through

$$al: X \to \mathcal{P}(A),\ X_b = \{x \in X \mid b \in al(x)\}$$

where $\mathcal{P}(A)$ is the set of subsets of A. Without loss of generality we shall assume that for each variable x, $al(x) \neq \emptyset$. The interpretation of al and X_b are as follows. For a variable x, al(x) is all the locations in which x might find itself during the course of the computation. X_b is all the variables that can ever be present at the location b. And the control state is used to indicate the presence and (explicit absence) of variables at locations. Let $C = (M,\zeta)$ be a case of $\overline{BP}$, $b \in A$ and $x \in X_b$. Then x is accessible at b for the activity associated with Z(b) in the case C iff b carries an H-token at M. In other words, x must be present in b at C.

We demand that al should be such that at each case, every variable is accessible at exactly one location. This is to ensure that the history of each variable is "continuous" (accessible at at least one location) and is "unambiguous" (never accessible at more than one location). Which does not of course rule out the possibility of executing a number of activities concurrently.

Our solution to ensuring this, is sufficient in that not every "behaviourally consistent" $\overline{BP}$ will satisfy our conditions. However our solution is simple and it comes with a number of additional - we think - elegant properties. The idea is to associate each variable with a ∇-component of BP (see def. 3.10).

Let $\{BG^1, BG^2, \ldots, BG^k\}$ be the set of ∇-components of BP where for $1 \leq i \leq k$, $BG^i = (V^i_\nabla, V^i_\&, A^i; Q^i, Z^i)$. We now define,

$$AL: X \to \{BG^1, BG^2, \ldots, BG^k\}$$

and derive al: $X \to \mathcal{P}(A)$ that we are after, as follows:

$$\text{For every } x \in X,\ al(x) = A^i \text{ where } AL(x) = BG^i.$$

Thus our proposal is to allocate a variable to all the arcs of exactly one ∇-component. As an example, consider the scheme of fig. 2.5.a with its two ∇-components shown in fig. 3.6.a. Suppose that $X = \{x_1, x_2, x_3\}$ and AL associates x_1 and x_3 with the left ∇-component and x_2 with the right one.

Then the resulting allocation of variables to arcs will be as shown in fig. 7.2.

Fig. 7.2

That the demand every variable should have a continuous and unambiguous history is now satisfied is brought out in

Theorem 7.1 Let BP be the generic well behaved scheme, $\{BG^1, BG^2, \ldots, BG^k\}$ the set of ∇-components of BP and $AL: X \to \{BG^1, \ldots, BG^k\}$. For $b \in A$, let $X_b = \{x \in X \mid AL(x) = BG^i$ and $b \in A^i$, the arcs of $BG^i\}$. Then for every $M \in [M^0\rangle$ (and for that matter, every $M \in [M^0])$ and every $x \in X$,

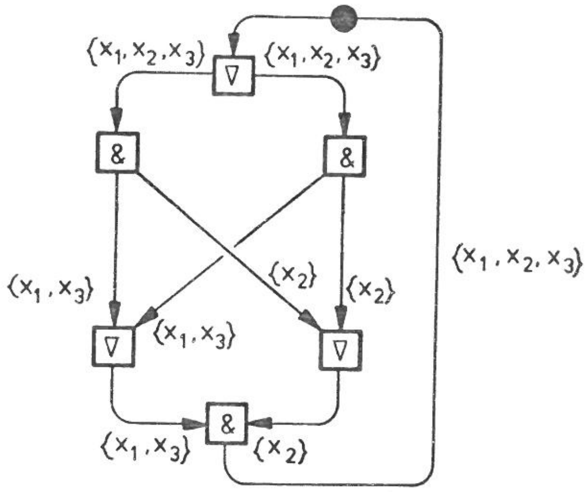


Fig. 7.2

$$|\{b\in A \mid x\in X_b \text{ and } b\in M_H\}| = 1.$$

Proof Follows easily from the definitions and prop. 3.6. ∎

We mention in passing that AL is well defined because the set of ∇-components of a well behaved scheme is non-empty; every arc is contained in a ∇-component according to theorem 3.9. The additional benefits of this allocation method is that it leads to some very regular patterns around the nodes.

Theorem 7.2 Let BP be the generic well behaved scheme and X, AL, X_b be defined as above. Then,

(1) For every ∇-node v, $\forall b,b'\in({}^\bullet v \cup v^\bullet)$; $X_b = X_{b'}$.

(2) For every &-node u,

$$\forall b,b'\in {}^\bullet u: b\neq b' \Rightarrow X_b \cap X_{b'} = \emptyset$$

$$\forall b,b'\in u^\bullet: b\neq b' \Rightarrow X_b \cap X_{b'} = \emptyset.$$

(3) For every node w, $\bigcup_{b\in {}^\bullet w} X_b = \bigcup_{b\in w^\bullet} X_b$.

Proof Follows easily from the definitions. ∎

The first part of the theorem states that the same set of variables is allocated to all the arcs that touch a ∇-node. (2) states that for a &-node, the sets of variables allocated to the input arcs are pair-wise disjoint as also the sets of variables allocated to the output arcs. The last part states that in our schemes variables are neither created nor destroyed.

7.2 Tests, Operations and the Firing Rules

The third part of theorem 7.2 suggests the notion of the variables accessible from a node. Let $w\in V$. Then X_w denotes the set of variables accessible from w and is given by

$$X_w = \bigcup_{b\in {}^\bullet w} X_b.$$

Intuitively X_w is the set of variables the node w has access to during an H-firing. Now given a value state ζ we will often have to work with that sub-vector of ζ which specifies the values of the variables accessible from the node w. For this purpose the following notation will come in handy. Let $w \in V$ and $X_w = \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$ with $i_1 < i_2 \ldots < i_m$ $(m \geq 1)$. Then,

$$\mathcal{D}_{(w)} = D_{i_1} \times D_{i_2} \times \ldots \times D_{i_m} \text{ and}$$

$$\zeta_{(w)} = (\zeta_{i_1}, \zeta_{i_2}, \ldots, \zeta_{i_m}).$$

In what follows we will assume for every node that we encounter that the set of variables accessible from that node is non-empty. Where we wish to drop this assumption, we will do so in a loud and clear manner.

Turning now to the operations in $\overline{BP}$ we associate a function f_u with each &-node u. f_u is defined over the variables accessible from u:

$$f_u : \mathcal{D}_{(u)} \to \mathcal{D}_{(u)}.$$

That for each such function the domain coincides with the range is a technicality forced on us by the third part of theorem 7.2. It is however just that, a technicality, and not a serious limitation. f_u can leave a number of variables in X_u unaffected; u can be used for pure synchronisation by setting f_u to be the identity function.

As for the tests, with each ∇-node v, we associate a family of (test) predicates $\{P_b \mid b \in v^{\bullet}\}$ indexed by the output arcs of v. Each predicate is defined over the set of variables accessible from v.

$$P_b : \mathcal{D}_{(v)} \to \{\underline{true}, \underline{false}\} \quad (b \in v^{\bullet})$$

P_b is said to be an <u>exit condition</u> for the ∇-node v. In an H-firing of v, P_b determines whether the output arc b <u>may</u> get an H-token. In this fashion, the exit conditions can be used to steer the course of the computation. In general more than one exit condition may be true at a case where the ∇-node in question is H-firable; we permit non-deterministic computations. It is also possible that none of the exit conditions of a ∇-node hold at a case even though every input arc carries a token with exactly one of them being an H-token; $\overline{BP}$ can have deadlocks even though the underlying control mechanism, which is what BP is, has been wired to be deadlock free.

We can now deal with the firing rules.

<u>Definition 7.1</u> Let (M^1,ζ^1), $(M^2,\zeta^2)\in[M^0>\times\mathcal{D}$.

1) Let w be a node such that $X_w = \emptyset$. Then $(M^1,\zeta^1)[\![w> (M^2,\zeta^2)$ iff $M^1[w>M^2$ in BP and $\zeta^1 = \zeta^2$.

2) Let v be a ∇-node. Then $(M^1,\zeta^1)[\![v> (M^2,\zeta^2)$ iff

a) $M^1[v>M^2$ in BP

b) $\zeta^1 = \zeta^2$

c) If $b\in M^2_H$ then $P_b(\zeta^1_{(v)}) =$ <u>true</u>.

3) Let u be a &-node. Then $(M^1,\zeta^1)[\![u> (M^2,\zeta^2)$ iff

a) $M^1[u>M^2$ in BP

b) If u may L-fire at M^1 in BP then $\zeta^1 = \zeta^2$

c) If u may H-fire at M^1 then $\zeta^2_{(u)} = f_u(\zeta^1_{(u)})$ and for $x_i \notin X_u$, $\zeta^1_i = \zeta^2_i$. ∎

The relation $[\![\ >$ specifies the transformation of the state effected through a node firing in $\overline{BP}$.

If a ∇-node v fires, the control state is changed as in BP, and the value state remains unchanged. If the outgoing arc $b\in v^\bullet$ is to get an H-token as a result of the H-firing of v, then the exit condition for b must be true. Thus the tests associated with a ∇-node come into play only during H-firings which is how it should be. If a &-node u fires at a state then the control state is once again changed just as it would be in BP. In an L-firing the value state remains unchanged; the operation associated with u is omitted. In an H-firing the values of the variables accessible from u are changed as specified by the function f_u assigned to u. The variables that are not accessible from u are not affected. Consequently, in general, a good many tests and operations can proceed concurrently.

The set of cases now is the set of states that can be reached from the initial case through node firings. More precisely,

Definition 7.2 $\mathcal{C}$, the set of cases of $\overline{BP}$ is the smallest subset of states given by:

1) $c^0 = (M^0, \zeta^0) \in \mathcal{C}$.

2) If $c^1 \in \mathcal{C}$ and for some node w and some state c^2, $c^1 [\![w > c^2$ then $c^2 \in \mathcal{C}$. ∎

7.3 The Input and Output Operations

In the interpreted bp scheme $\overline{BP}$, we will have two special kinds of nodes called R-nodes (R for receive) and S-nodes (S for send). An R-node will have one dangling input arc and an S-node one dangling output arc. They behave like &-nodes except for L-firings in which the dangling arcs do not participate. In diagram 7.3 showing the variable assignment and the firing rules for R- and S-nodes, the dangling arcs are indicated by squiggly lines.

Fig. 7.3

The environment is expected to send only H-tokens along a dangling input arc. It expects to receive only H-tokens along dangling output arcs. In an H-firing, all input arcs of an R-node must carry H-tokens; in an L-firing, the dangling arc is ignored. When an S-node H-fires, all output arcs get an H-token; in an L-firing, the dangling arc does not get any token.

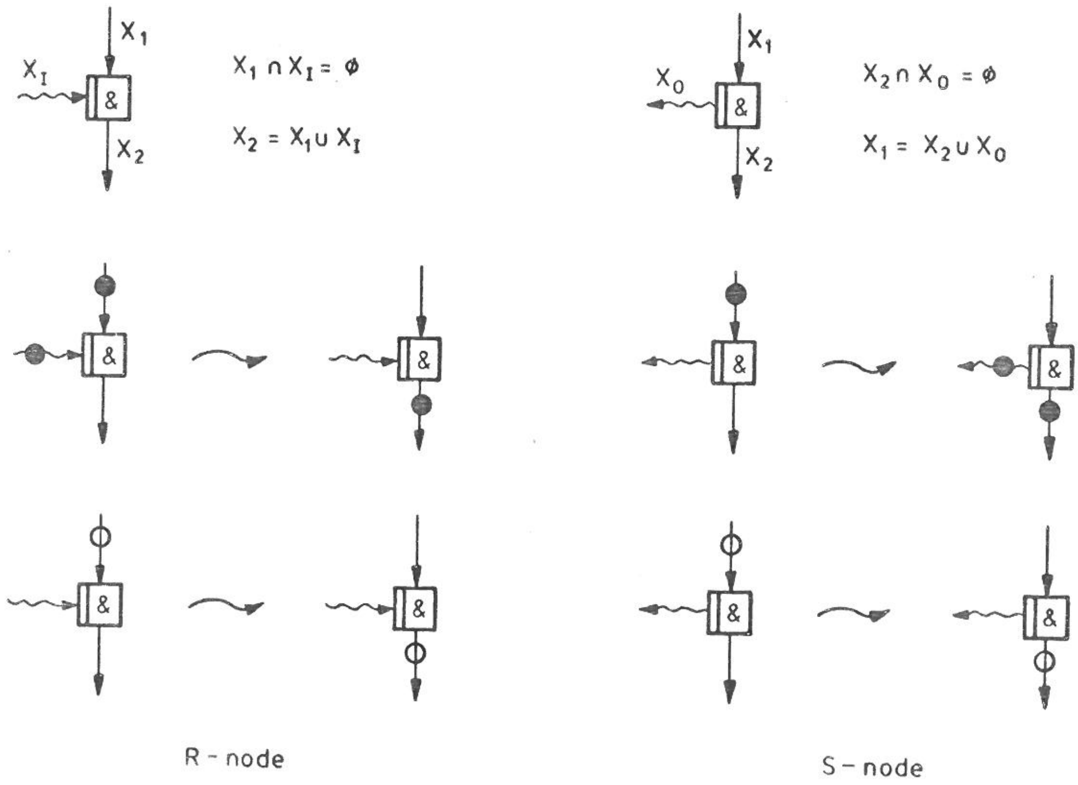

X_1
X_I
&
X_2
$X_1 \cap X_I = \emptyset$
$X_2 = X_1 \cup X_I$
X_1
X_0
&
X_2
$X_2 \cap X_0 = \emptyset$
$X_1 = X_2 \cup X_0$
R - node
S - node

Fig. 7.3

In the interpreted scheme shown in fig. 7.1, there is just one R-node which receives inputs from the environment and just one S-node through which the scheme sends out the final output. However, we may also use a proper combination of S- and R-nodes to introduce a kind of block structure into our schemes.

Intuitively, we consider a block to be an 'arc-like' subscheme, i.e. a subscheme which may replace, or explicate, in an interpreted scheme, a &-node with exactly one input and one output arc. In fig. 7.4 we show two typical patterns that we would like to be able to treat as blocks, an iteration (a) and a feed-back (or delay, or storage) (b). Both are assumed to use an external variable x and internal variable y.

Fig. 7.4

Clearly, neither of the two patterns can safely replace a one-in/one-out &-node. The iteration may receive an H-token at its input but deliver an L-token in return. Additionally, the allocation of the variables cannot be made consistently. The feed-back will be in a deadlock as soon as it receives an L-token. Using a proper combination of R-nodes and S-nodes, however, the two patterns can be encapsulated in such a way that the resulting schemes behave in the desired fashion. This is shown in fig. 7.5.

Fig. 7.5

Thus through a systematic application of the R-nodes and S-nodes, one can construct highly modularised and fairly complex programs.

7.4 The Example and some Remarks

We shall now briefly indicate what the interpreted scheme of fig. 7.1 does. We are given a discrete chain $(X;\leqq)$. For each element x there is an immediate predecessor denoted as ${}^0x$ and an immediate successor denoted as x^0. Hence if $x<y$ then $x^0\leqq y$ and $x\leqq {}^0y$. We assume two special elements $\bot$ and $\top$ which are not in X. If the chain has a least element x then ${}^0x = \bot$ and ${}^0\bot = \bot$ by convention. If the chain has a greatest element y then $y^0 = \top$ and $\top^0 = \top$ by convention.

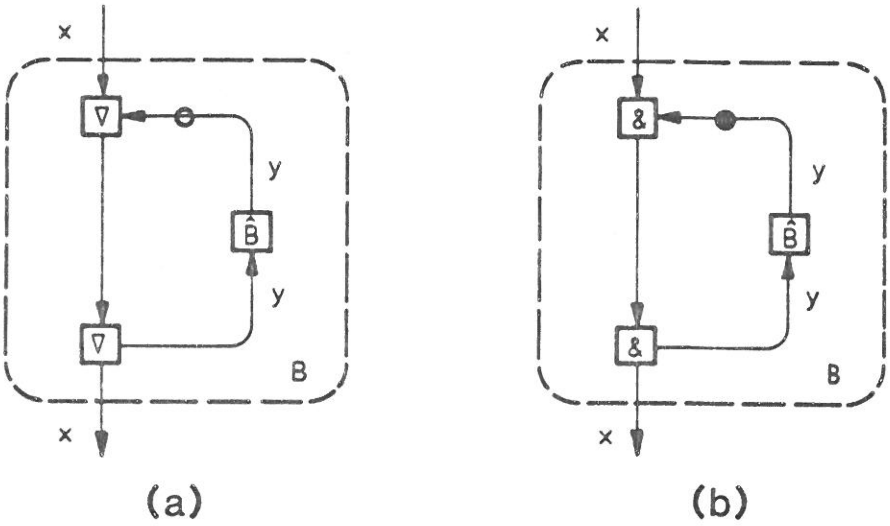


Fig. 7.4

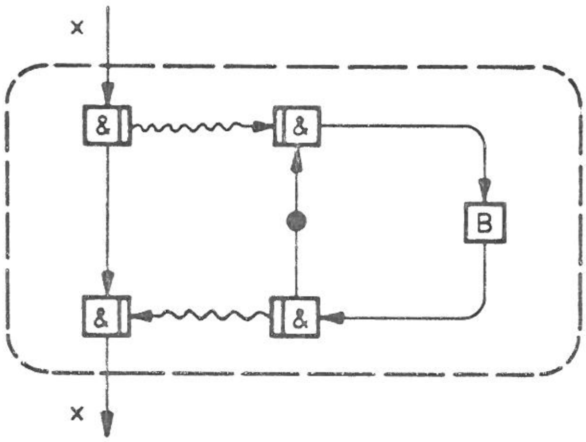


Fig. 7.5

The problem is, given two elements x and y, to determine whether $x \leqq y$ or $y \leqq x$. We assume that for each x we can compute ^{0}x and x^{0}. For reasons that we do not wish to go into here, it is convenient to treat the R-nodes and S-nodes as &-nodes, as far as the variable assignments are concerned. As a consequence, the variables that come into a scheme (input variables) are the same as the variables that leave the scheme (output variables). Once again this is a technicality and not a serious limitation. In our example, the final value of y will indicate the answer. If $x \leqq y$ initially, the final value of y is set to $\top$. If $y < x$ the final value of y is set to $\bot$. As a bonus, the final value of x will be "approximately" half-way between the initial values of x and y.

At the initial case, the scheme waits for the inputs x and y. The computation is finished if one of the predicates $[b_1 = t_1]$ or $[b_2 = t_2]$ becomes true. This will lead through a suitable H-firing of the corresponding ∇-node to the coding of the results onto the (now) output variables x and y. Finally this will cause the H-firing of the S-node indicating that the environment has been presented with the results. The scheme will then wait for the next pair of inputs to arrive.

This example also illustrates some of the things that can be done to make an interpreted scheme look like a flow chart representation of a program. As done in this example, one can suppress a great deal of the information concerning allocation of variables to arcs. The point however is, one could complete the picture in a systematic fashion and check whether a correct allocation function has been chosen. Secondly we have used an IF-THEN-ELSE construct in place of a small ∇-subscheme; once again in order to avoid cluttering up the diagram. By a suitable use of the interface nodes we can also "implement" and use a DO WHILE construct (as already indicated in the previous section). In this fashion, a good deal of the succinctness and readability of sequential textual language can be transported to interpreted schemes.

What we can not do is dynamically create (and destroy) variables. More importantly, at this stage, we cannot have shared data structures that can be accessed in an exclusive mode from different parts of the program. However, by using slightly more complex interface nodes we know how to get around this problem. We however do not wish to go into details here. For a more elaborate presentation of the computational interpretation the interested reader is referred to [7]. There we have also worked out a more interesting example, consisting of a highly asynchronous unbounded stack which can grow and shrink on demand.

8. DISCUSSION

The motivation for the study reported here was to better understand the interplay between choice and independence (of actions) in the context of distributed systems. While net theory offers the tools for clearly distinguishing between the phenomena of causality, independence and choice and while we were reasonably familiar with this theory, we decided to carry out the work under the banner of Petri nets. What was needed was a suitable model. And we decided to construct one based on live and safe marked graphs in order to play it safe (sorry!). That live and safe marked graphs have an extensive theory can be guessed from the selected summary given in section 2. It is also amply documented in the literature, [6, 13,14]. The reader would surely agree that we have thoroughly exploited this theory to develop our results on bp schemes. The point is, the decision to base our model on live and safe marked graphs has paid rich dividends.

The, at first sight bizarre, idea to explicitly represent omissions of actions through L-tokens is crucial for establishing a frutiful connection with marked graphs. We believe that this idea also has some independent merits. It has led us with no ifs and buts to the notion of good behaviour. It has opened up the possibility of establishing strong links between intuition regarding what constitutes "good" structure and formally provable behavioural properties. Holt has made extensive attempts to denote explicitly omissions in modelling systems [9]. His concerns however go much deeper and his game board is much larger. As mentioned at the very beginning of this paper, we decided to carry out our study in a very restricted setting. The importance of omission signals and the roles they play (or could play) in systems in general and organisational systems in particular has been often pointed out by Petri [20].

In developing the theory of bp schemes the focus of attention has been the synthesis problem. The obvious reason is, given our original motivation, it was the best way to gain a deep understanding of the model. In attacking the synthesis problem we have indeed gained some insights and discovered a few facts. The general insight is that in our schemes at least, there is an intimate relationship between choice and independence. And often this relationship can be expressed as a beautiful duality relation. Now we turn to some of the individual results.

In the absence of L-tokens well behaved schemes turn out to be a sub-class of live and safe free choice nets (theorem 3.5). And in fact they lie properly between live and safe free choice nets on the one side and live and safe marked graphs and state machines on the other side. Through these relationships, well behaved schemes inherit the important and elegant structural properties of live and safe free choice nets (theorem 3.9).

We have also established that in a bp scheme one cannot choose between good behaviour and bad behaviour within a marking class (theorem 4.9). We know that a well behaved scheme is essentially determined by its underlying marked graph (theorem 4.10). Indeed if the stronger result we conjecture in this direction (the second conjecture at the end of section 4) is true, then the synthesis procedure can be greatly simplified. One can just generate well formed bp graphs, i.e. those that can be endowed with a good marking. To do so, a much simpler version of T_2, one in which we merely ensure that the target bp graph is strongly connected, can be used. Once the required well formed bp graph has been generated, then viewing it as a digraph we can give it a "gray" marking and get a live and safe marked graph (for details see [6]). Finally, using the idea suggested in the proof of theorem 4.10, we can convert this gray marking to a coloured marking to obtain a well behaved scheme.

The next result of interest is theorem 6.3. It identifies three kinds of local structural defects in ill behaved schemes. Stated differently, it shows three improper ways of combining choice and concurrency. To date, theorem 6.3 is the best structural result we have. We even conjecture that the class of well formed bp graphs can be completely characterised with the help of the three patterns identified in this result. Theorem 6.9 has a grain of (pleasant) surprise to it. Often what constitutes "good" structure is settled through appeals to tradition, good sense and taste. (Not that we are strongly opposed to these things.) For *SWF* though we get some tangible confirmation of the intuition that it is a class of "well structured" objects. They turn out to be precisely the objects which have the behavioural property of being well behaved forwards and backwards.

Turning to the synthesis procedure, our transformation rules are not the only imaginable ones. We have selected them mainly because they are convenient to work with. We are not sure that they constitute a minimal set. Indeed we are yet to identify a well behaved scheme for constructing which we have to make use of T_5. It is just that we do not know how to prove the completeness result for the smaller set of rules. The reverse of T_5 namely R_3 is needed in the proof of lemma 6.4 which is crucial for proving that every well behaved irreducible scheme is elementary. We suspect that if T_5 could be dropped then a more attractive size factor and consequently a less painful proof of completeness can be worked out. We could view our transformations and reductions as simply attempts to restructure a net model while preserving certain properties. In this context, similar work has been carried out for larger classes of nets by Berthelot et al. [2], Andre [1] and for live and safe marked graphs by Murata [18]. The main difference is that our rules provide a solution to the synthesis problem whereas in the papers cited above, the aim is to simplify, where possible, the analysis problem or to provide a partial solution to the synthesis problem.

Section 7 is to be viewed as a first step toward transferring the knowledge that has been obtained about bp schemes to concurrent programs. Here we have merely a model - but we hope an interesting model - of a class of distributed computations. However we have tried our best to make the interpreted schemes look like programs. What we have gained at this stage is that notions like ∇-components and &-components can be readily transported to the interpreted schemes. We have also shown that the theory of bp schemes can be used to develop results on the layer above. We have in mind theorems 7.1 and 7.2 where we have exploited the notion of ∇-components to identify a class of interpreted schemes, that are guaranteed to have some consistency properties. What we would like to do in the future is transfer the insights on the synthesis technique to the level of interpreted schemes, i.e. to develop rules for synthesising and manipulating interpreted schemes in "meaning-preserving" ways. Here the techniques reported by Roucairol in [22] might provide some guidance.

Acknowledgements

Our heartfelt thanks to Karen Møller. Her patience and competence was crucial in the preparation of the manuscript. We would also like to thank Elizabeth Munch for placing her excellent drawing skills at our disposal. This paper was written while the second author was spending a sabbatical semester at Aarhus University. He would like to thank the members of the Computer Science Department for providing a very pleasant working atmosphere.

References

[1] C. Andre, Structural Transformation giving B-Equivalent PT-nets, Proceedings of the third European Workshop on Applications and Theory of Petri Nets, Varenna, Italy (1982).

[2] G. Berthelot, G. Roucairol and R. Valk, Reductions of Nets and Parallel Programs, in: W. Brauer, Ed., Net Theory and Applications, (Springer Verlag, Berlin, Heidelberg, New York 1980) 277-290.

[3] F. Commoner, A.W. Holt, S. Even and A. Pnueli, Marked Directed Grahps, J. of Compt. and System Sciences 5 (1971) 511-523.

[4] H.J. Genrich, K. Lautenbach and P.S. Thiagarajan, Elements of General Net Theory, in: W. Brauer, Ed., Net Theory and Applications (Springer Verlag, Berlin, Heidelberg, New York 1980) 21-163.

[5] H.J. Genrich and K. Lautenbach, System Modelling with High-Level Petri Nets, Theoretical Computer Science 13 (1981) 109-136.

[6] H.J. Genrich and K. Lautenbach, Synchronisationsgraphen, Acta Informatica 2 (1973) 143-161.

[7] H.J. Genrich and P.S. Thiagarajan, Well Formed Flow Charts for Concurrent Programming, Proceedings of the IFIP TC2 Working Conference on "Formal Description of Programming Concepts II", Garmisch-Partenkirchen, W. Germany (1982).

[8] M. Hack, Analysis of Production Schemata by Petri Nets, M.S. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA (1972).

[9] A.W. Holt, Role/Activity Models and Petri Nets, Second Semi-Annual Technical Report, CADD-7503-1411, Massachusetts Computer Associates, Wakefield, Massachusetts, USA (1975).

[10] K. Jensen, Coloured Petri Nets and the Invariant Method, Theoretical Computer Science 14 (1981) 317-336.

[11] M. Jenssen and R. Valk, Formal Properties of Place/Transition-Nets, in: W. Brauer, Ed., Net Theory and Applications (Springer Verlag, Berlin, Heidelberg, New York 1980) 165-210.

[12] N.D. Jodwani and J.R. Jump, Top-Down Design in the Context of Parallel Programs, Information and Control 40 (1979) 241-257.

[13] J.R. Jump and P.S. Thiagarajan, On the Equivalence of Asynchronous Control Structures, SIAM J. on Computing 2 (1973) 67-87.

[14] J.R. Jump and P.S. Thiagarajan, On the Interconnection of Asynchronous Control Structures, J. of ACM 22 (1975) 596-612.

[15] P.E. Lauer, M.W. Shields and E. Best, Formal Theory of the Basic COSY Notation, Technical Report 143, Computing Laboratory, University of Newcastle-upon-Tyne, Newcastle, Great Britain (1979).

[16] A. Mazurkiewicz, Invariants of Concurrent Programs, in: J. Madley, Ed., International Conference on Data Processing, IFIP-INFOPOL 1976 (North-Holland 1977) 353-372.

[17] R. Milner, A Calculus of Communicating Systems, Springer Lecture Notes in Computer Science 92 (Springer Verlag, Berlin, Heidelberg, New York 1980).

[18] T. Murata, Synthesis of Decision-Free Concurrent Systems for Prescribed Resources and Performance, IEEE Transactions on Software Engineering 6 (1980).

[19] C.A. Petri, Introduction to General Net Theory, in: W. Brauer, Ed., Net Theory and Applications (Springer Verlag, Berlin, Heidelberg, New York 1980) 1-20.

[20] C.A. Petri, Private communication.

[21] W. Reisig, Petri Nets with Individual Tokens, Proceedings of the Third European Workshop on Applications and Theory of Petri Nets, Varenna, Italy (1982).

[22] G. Roucairol, Parallelization of Single-Assignment Programs, in: M. Feilmeir, Ed., Parallel Computers - Parallel Mathematics (North Holland, Amsterdam 1977).

[23] S. Skyum, Private communication.

[24] R. Valette, Analysis of Petri Nets by Stepwise Refinements, J. of Compt. and System Sciences 18 (1979) 35-46.

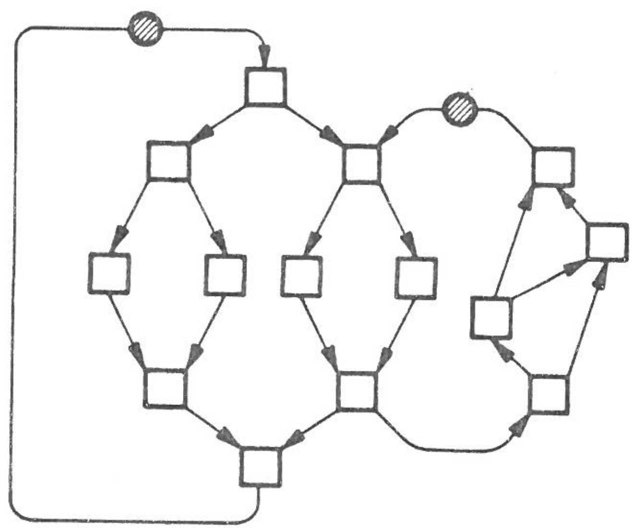


Fig. 1.1

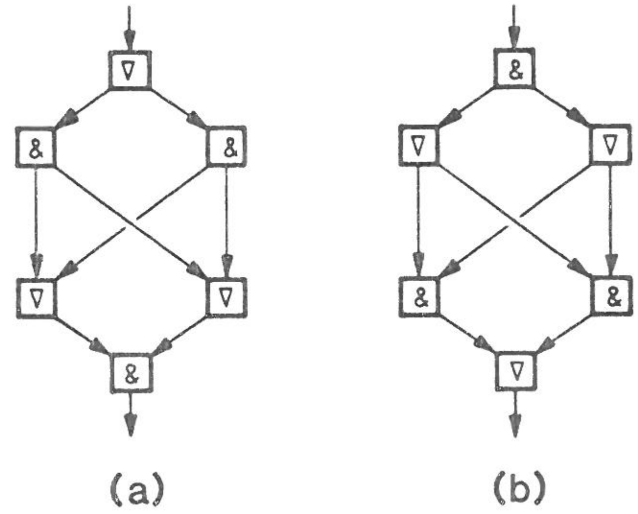
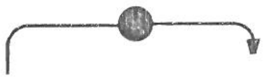


Fig. 2.2



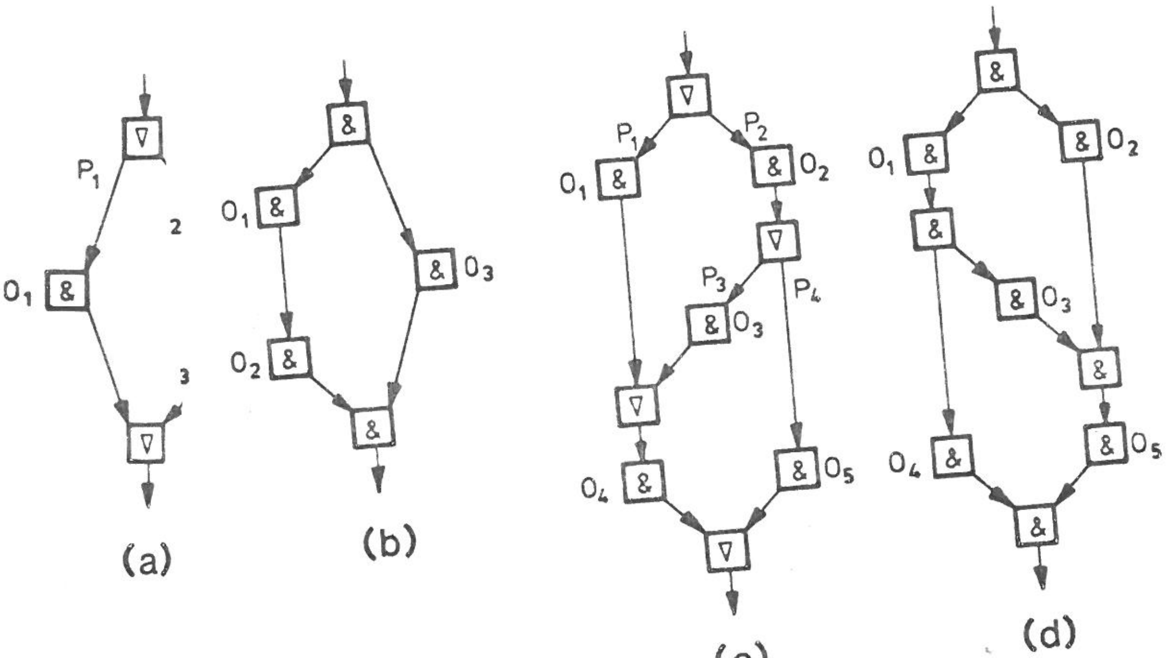


Fig. 2.1

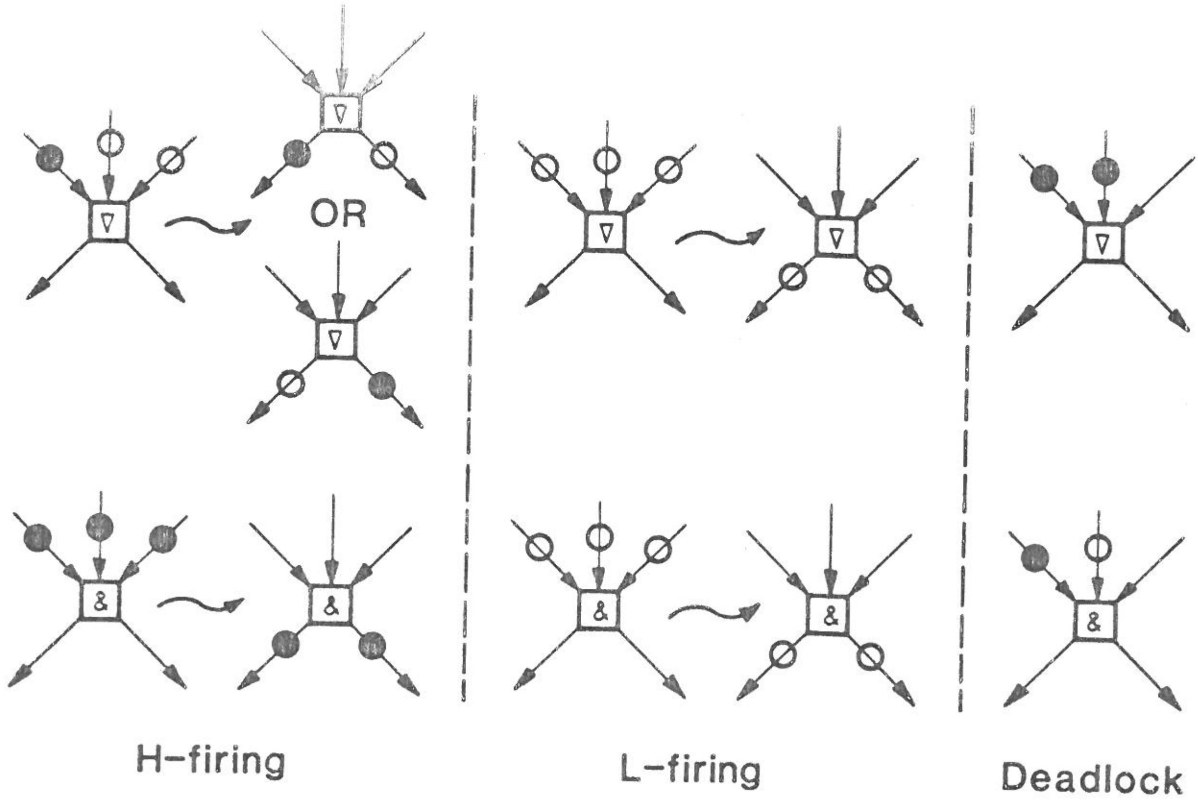


Fig. 2.4

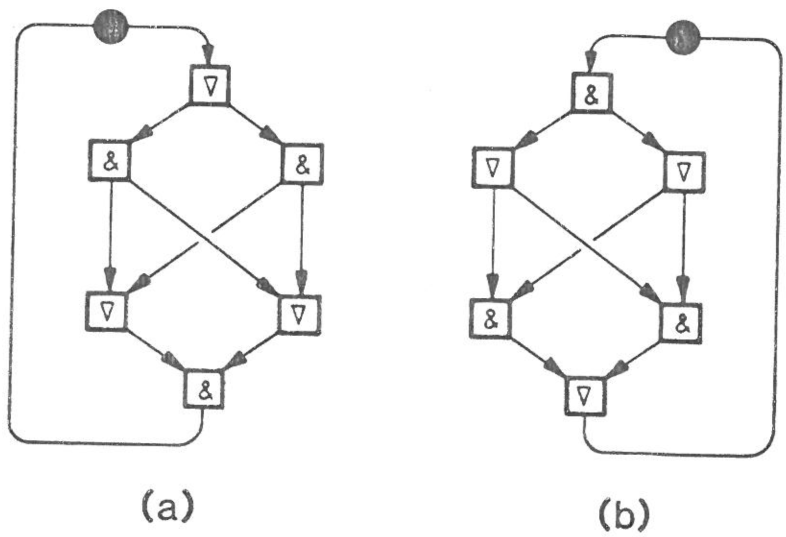


Fig. 2.5

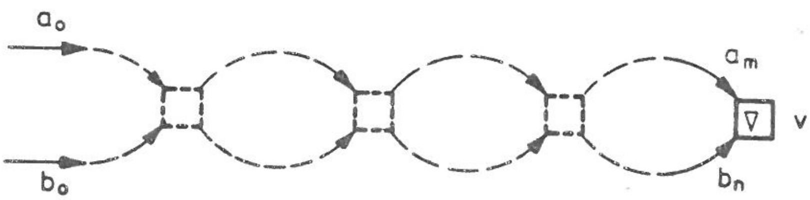


Fig. 2.6

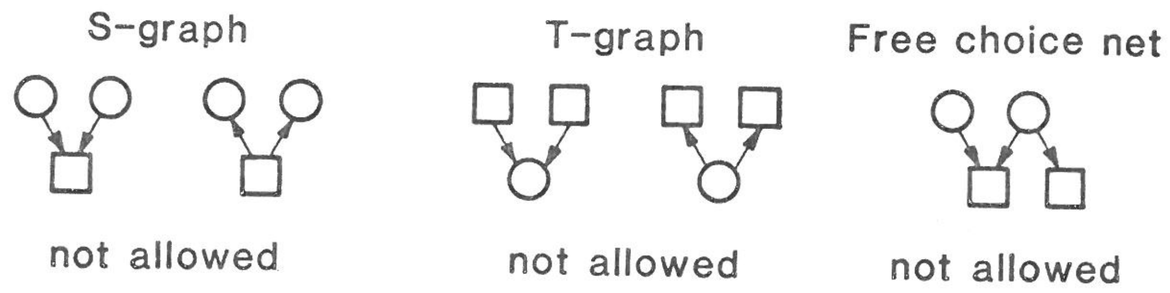

S-graph
T-graph
Free choice net
not allowed
not allowed
not allowed

Fig. 3.1

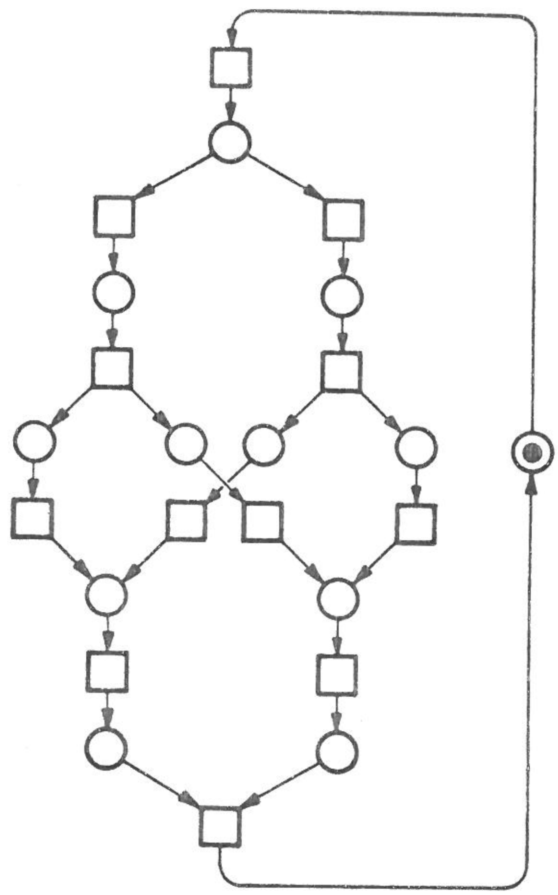


Fig. 3.2

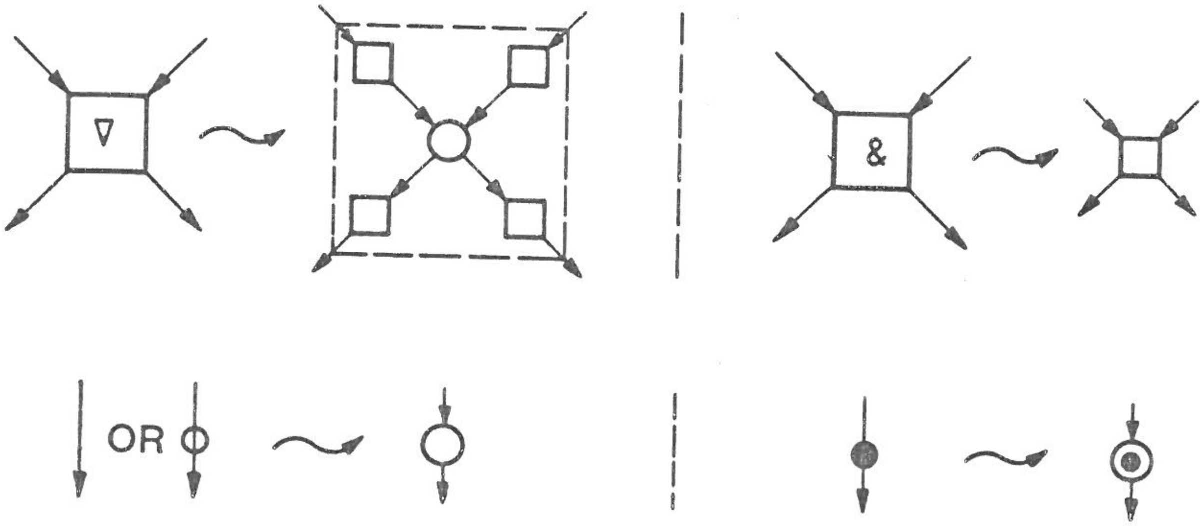


Fig. 3.3

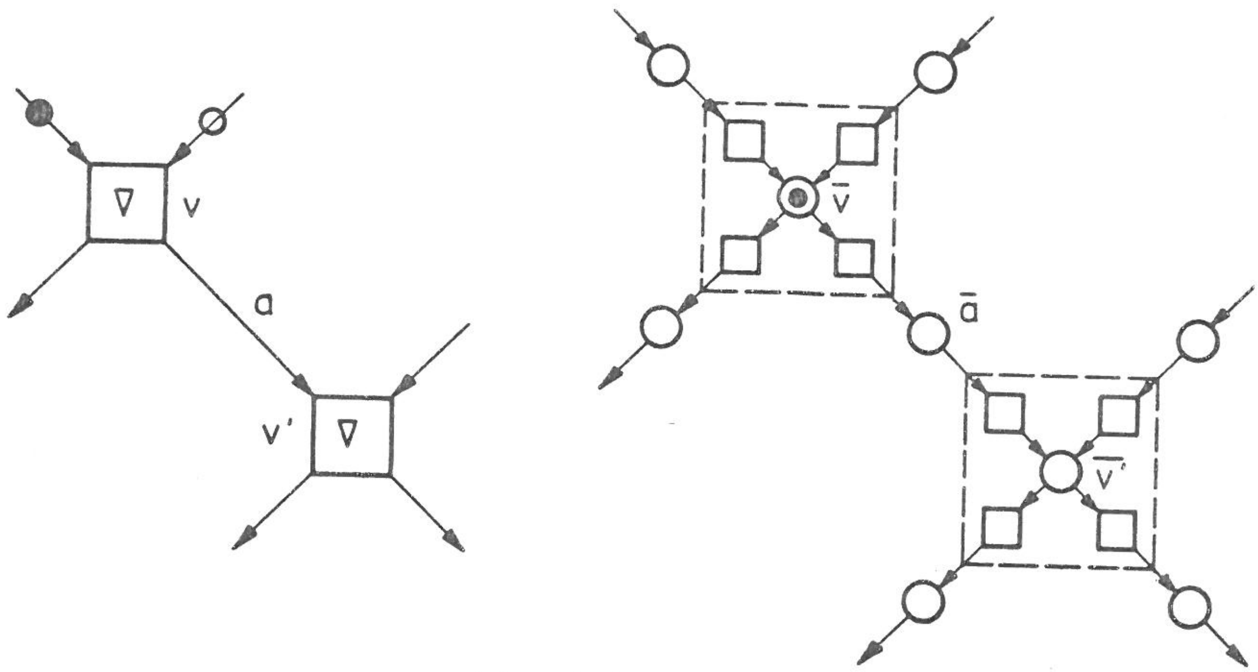


Fig. 3.4

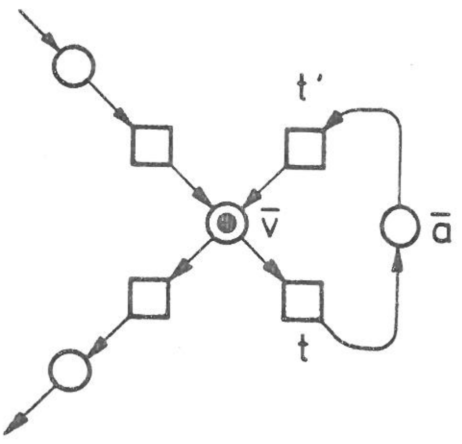


Fig. 3.5

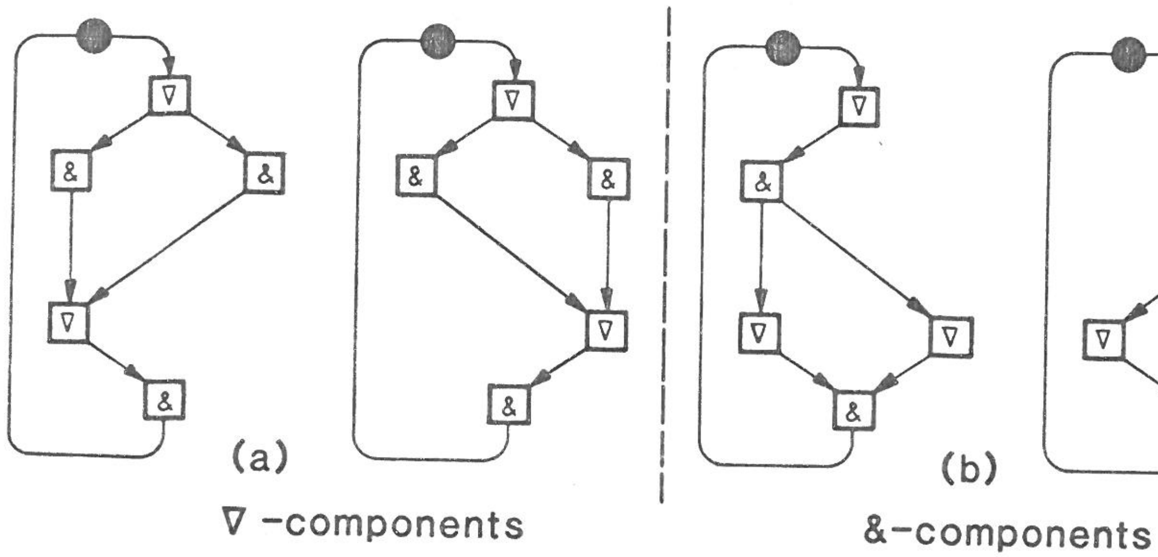


Fig. 3.6

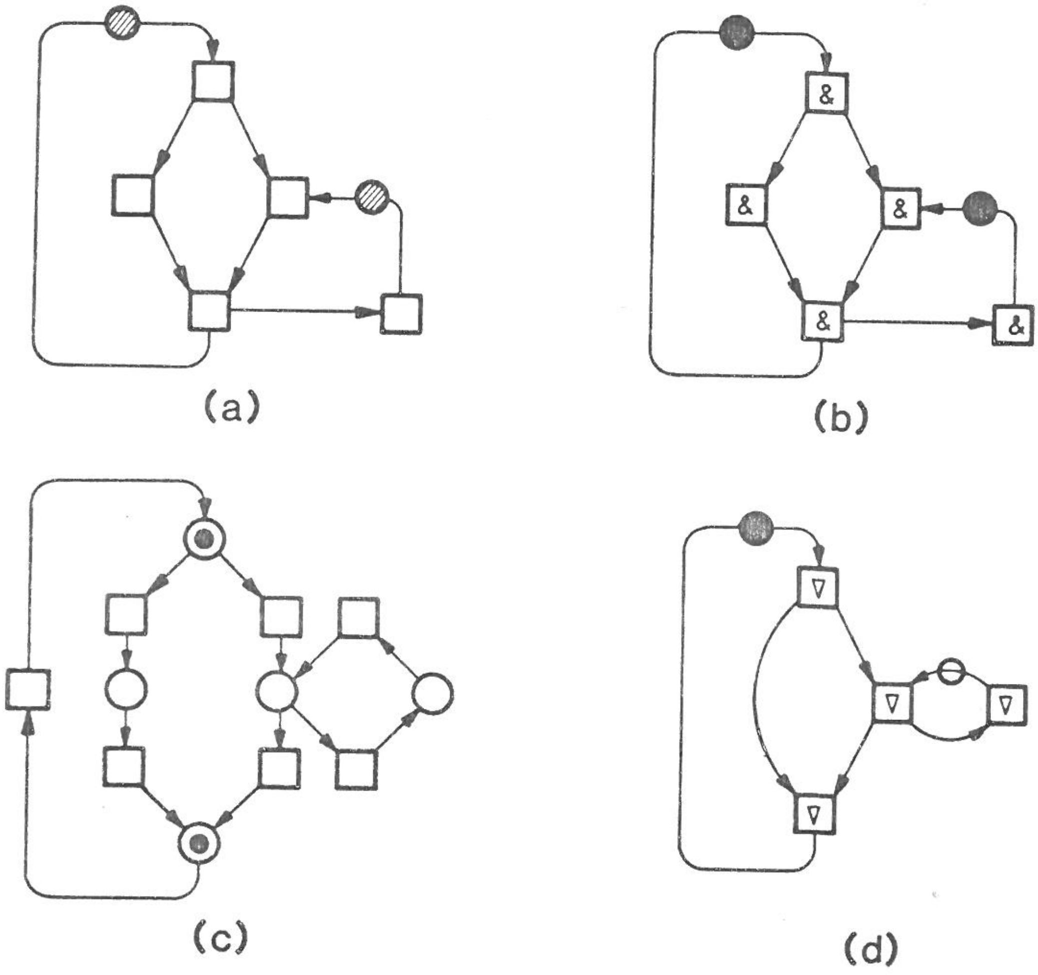


Fig. 3.7

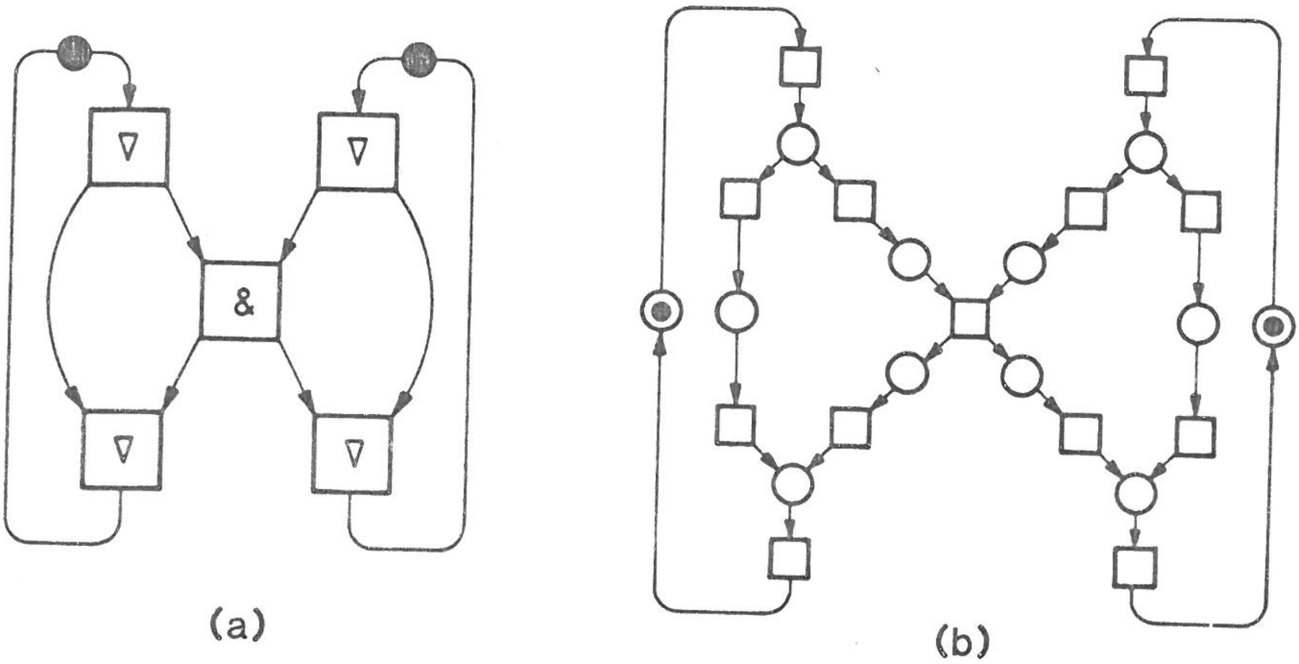

&
(a)
(b)

Fig. 3.8

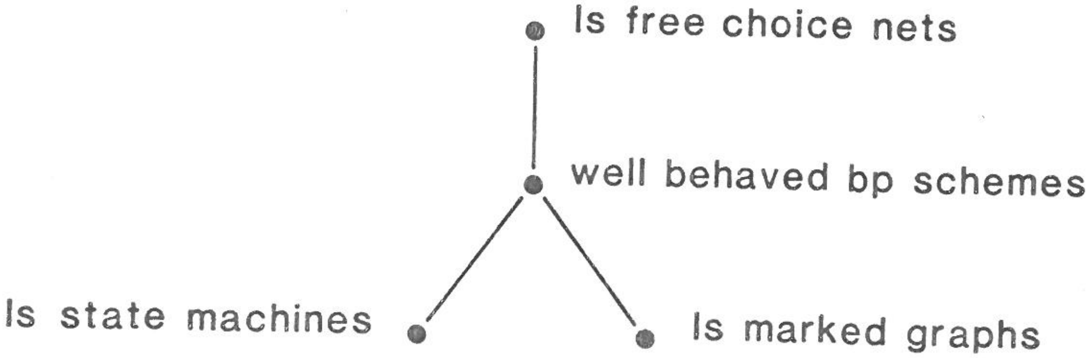

ls free choice nets
well behaved bp schemes
ls state machines
ls marked graphs

Fig. 3.9

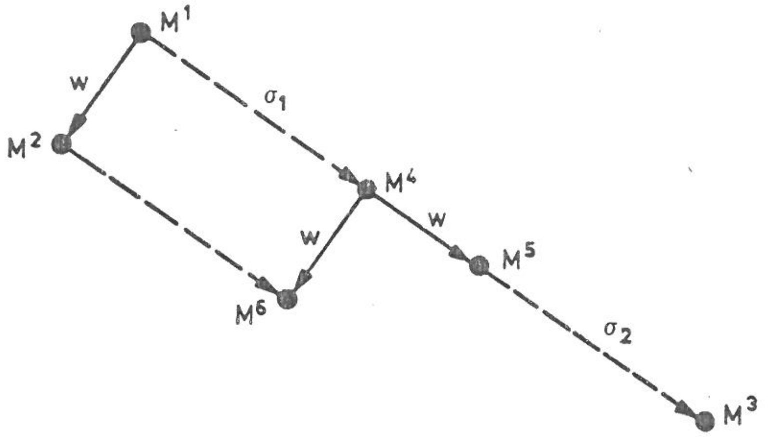


Fig. 4.1

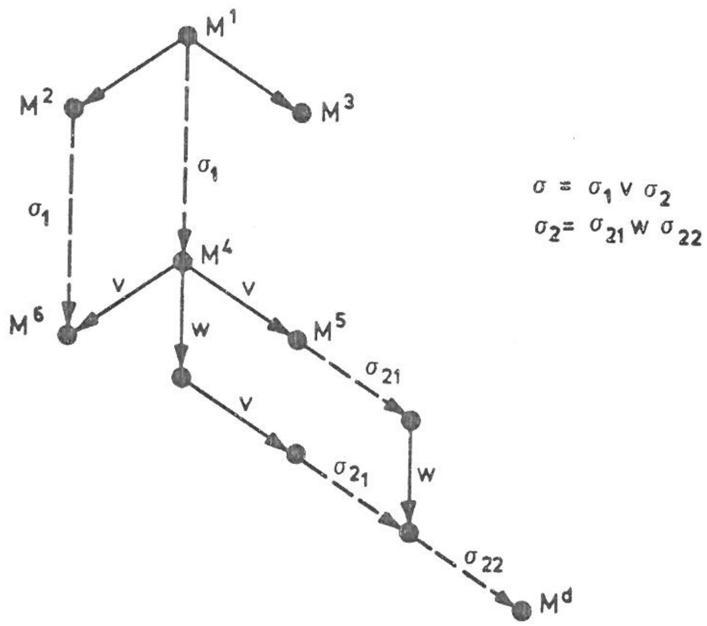


Fig. 4.2

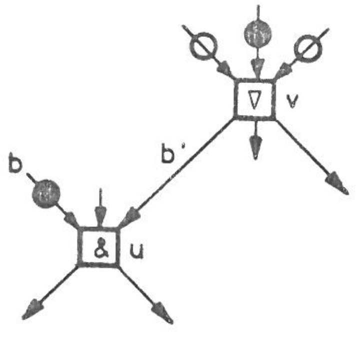


Fig. 4.3

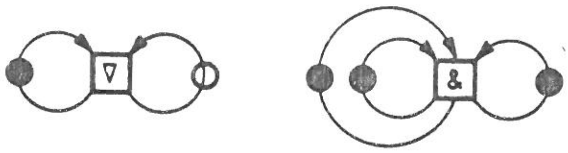


Fig. 5.1

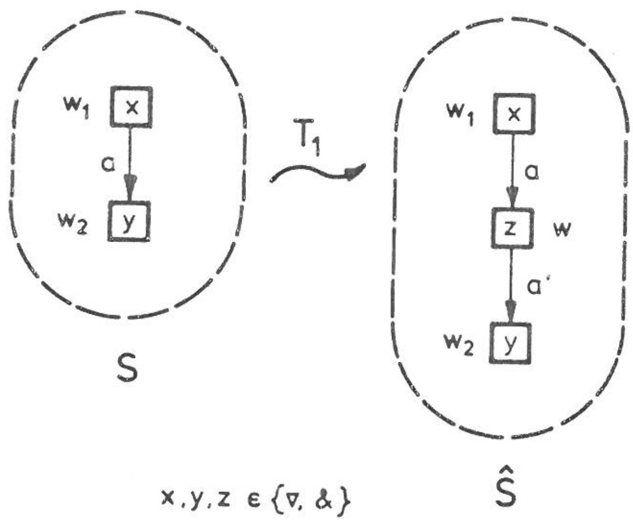


Fig. 5.2

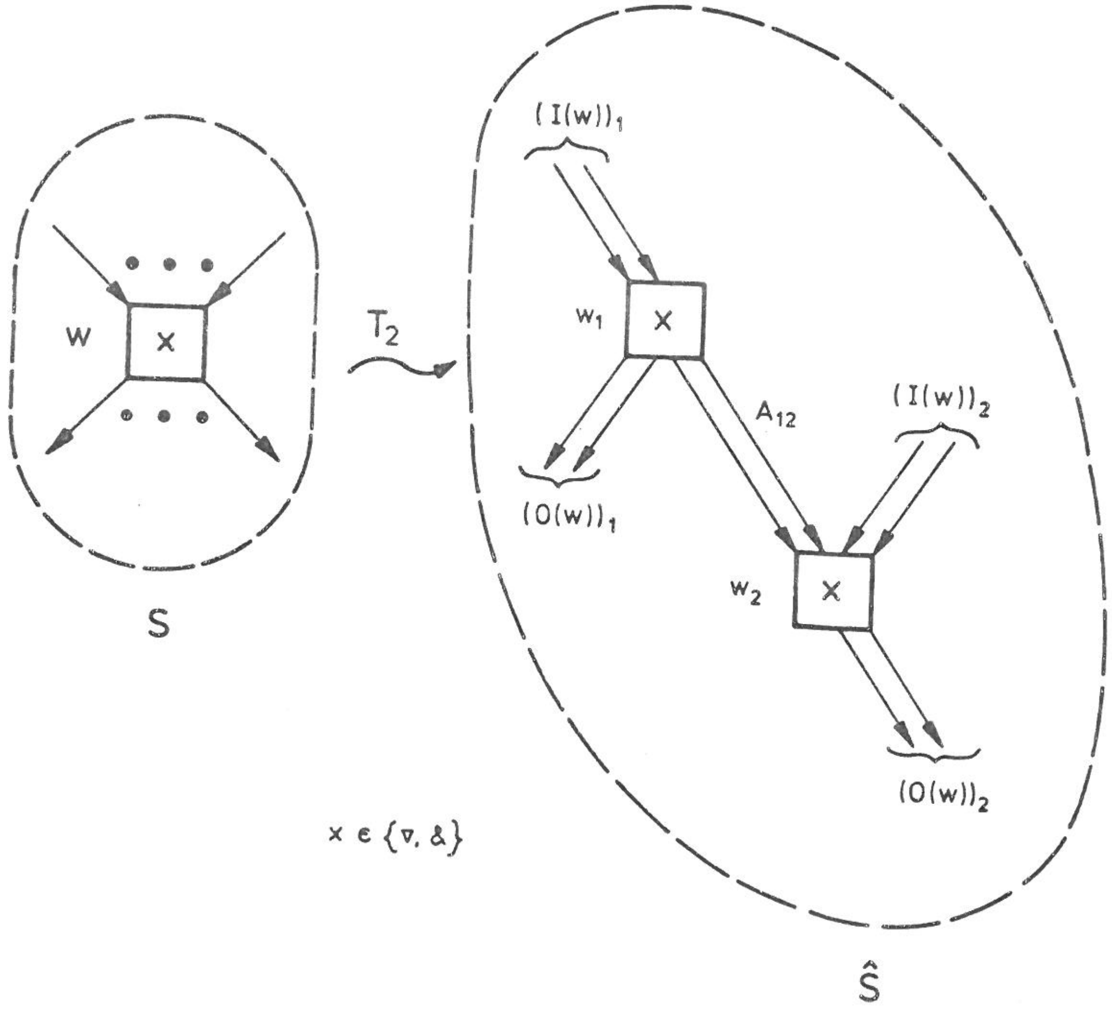


Fig. 5.3

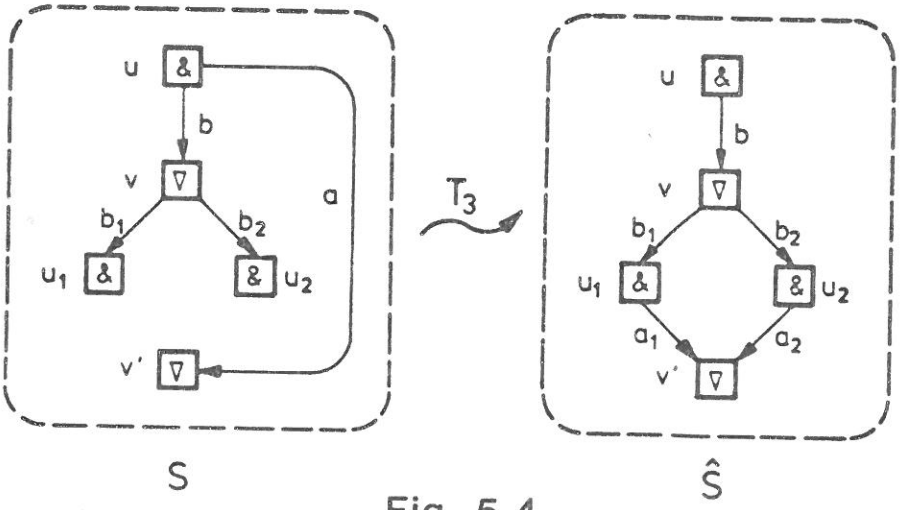


Fig. 5.4

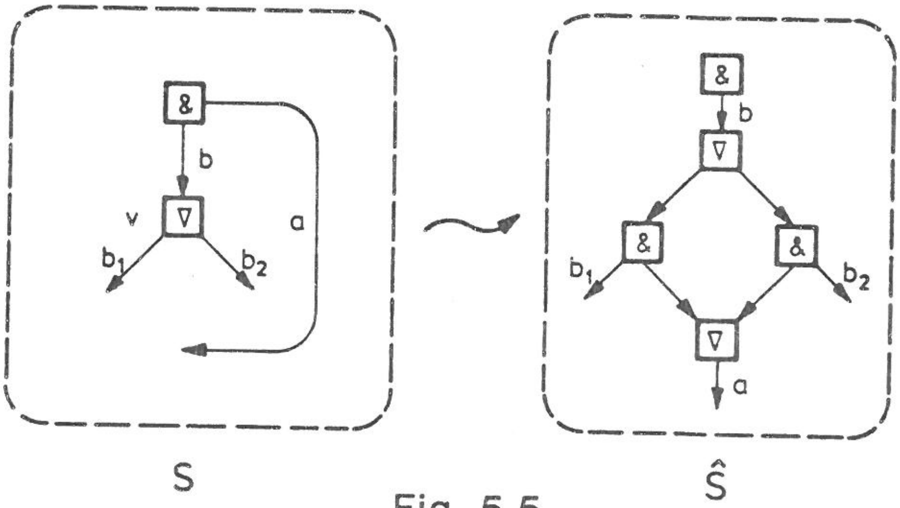


Fig. 5.5

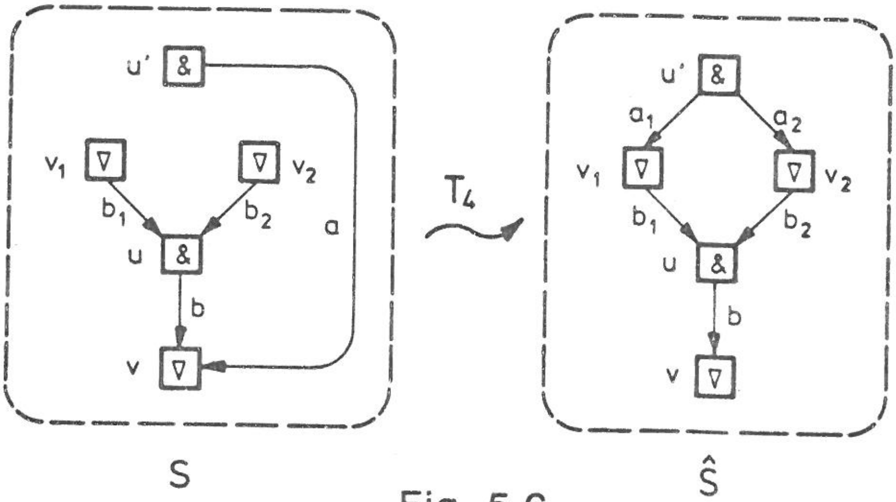


Fig. 5.6

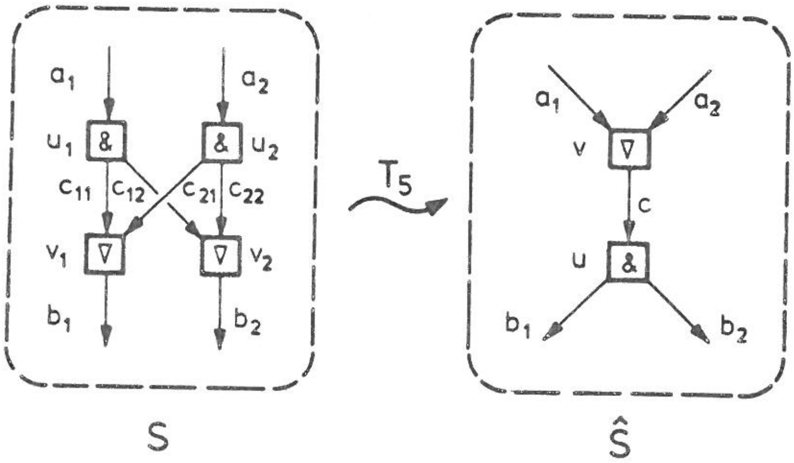


Fig. 5.7

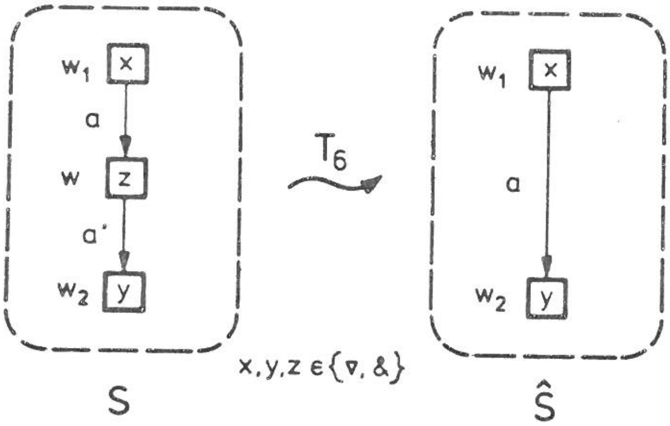


Fig. 5.8

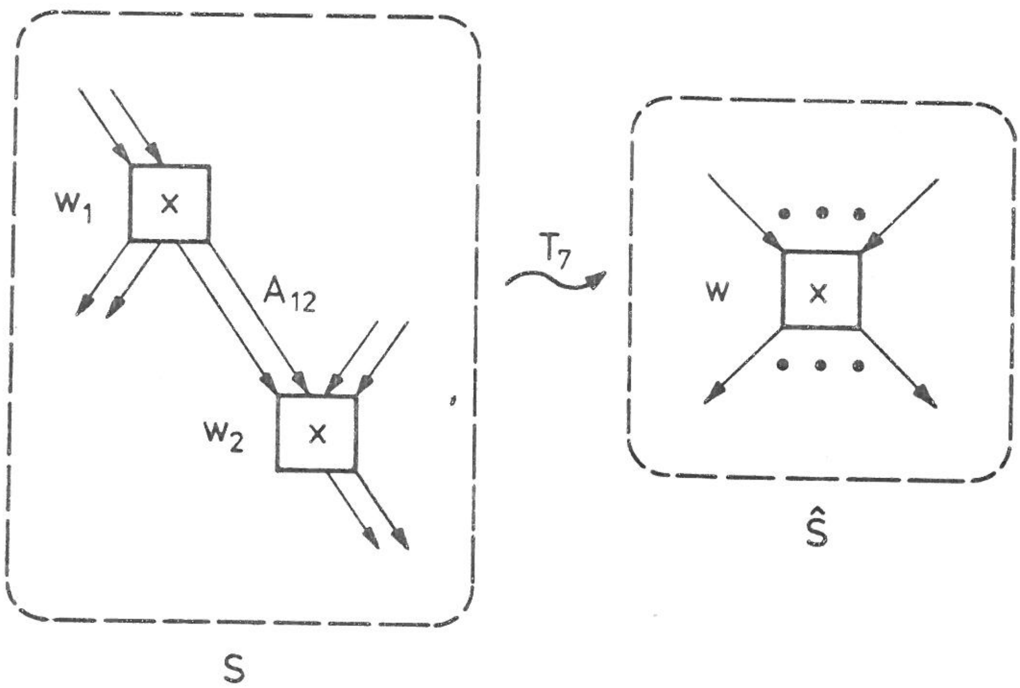


Fig. 5.9

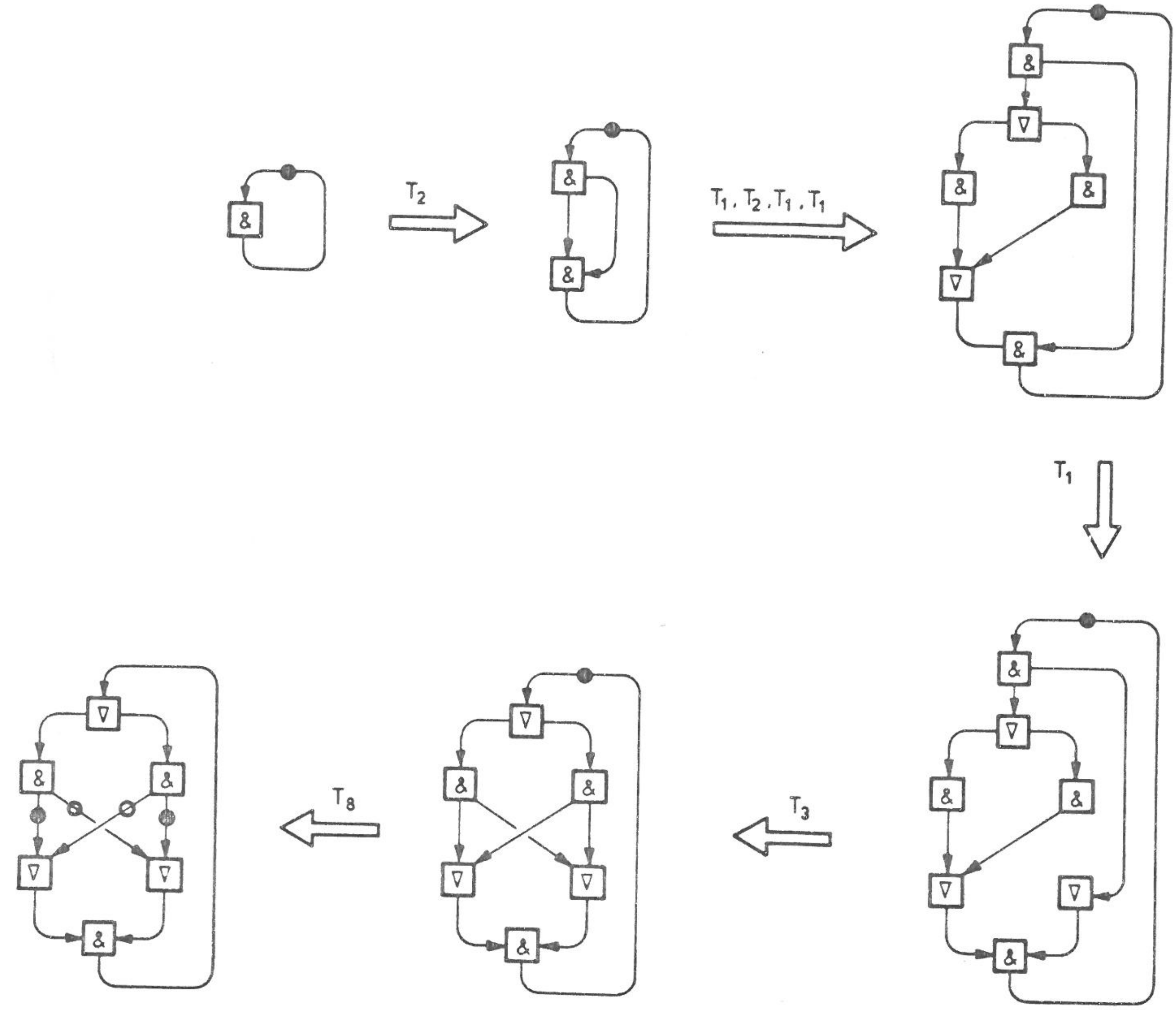
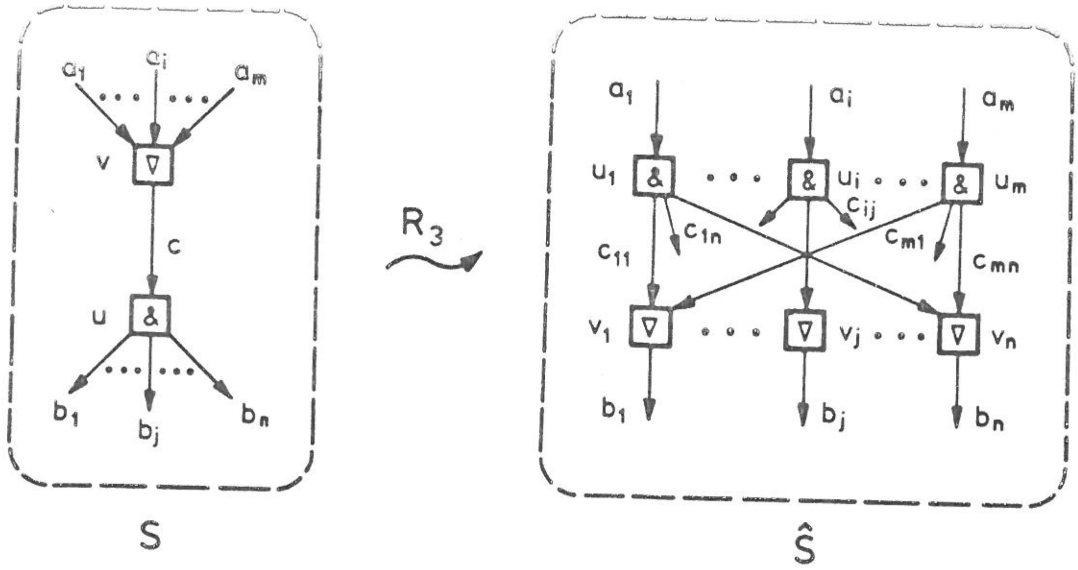
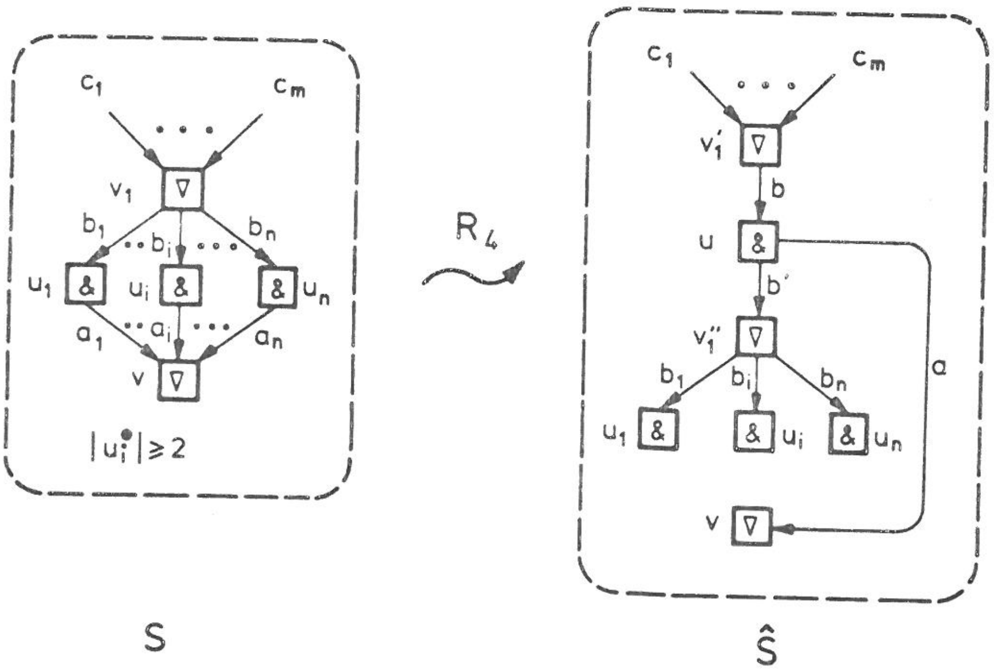


Fig. 5.10



a_1
a_i
a_m
v
c
u
b_1
b_j
b_n
S
R_3
u_1
u_i
u_m
c_{1n}
c_{ij}
c_{m1}
c_{11}
c_{mn}
v_1
v_j
v_n
$\hat{S}$

Fig. 6.1



c_1
c_m
v_1
b_1
b_i
b_n
u_1
u_i
u_n
a_1
a_i
a_n
v
$|u_i^\bullet| \geq 2$
S
R_4
v_1'
b
u
b'
v_1''
a
$\hat{S}$

Fig. 6.2

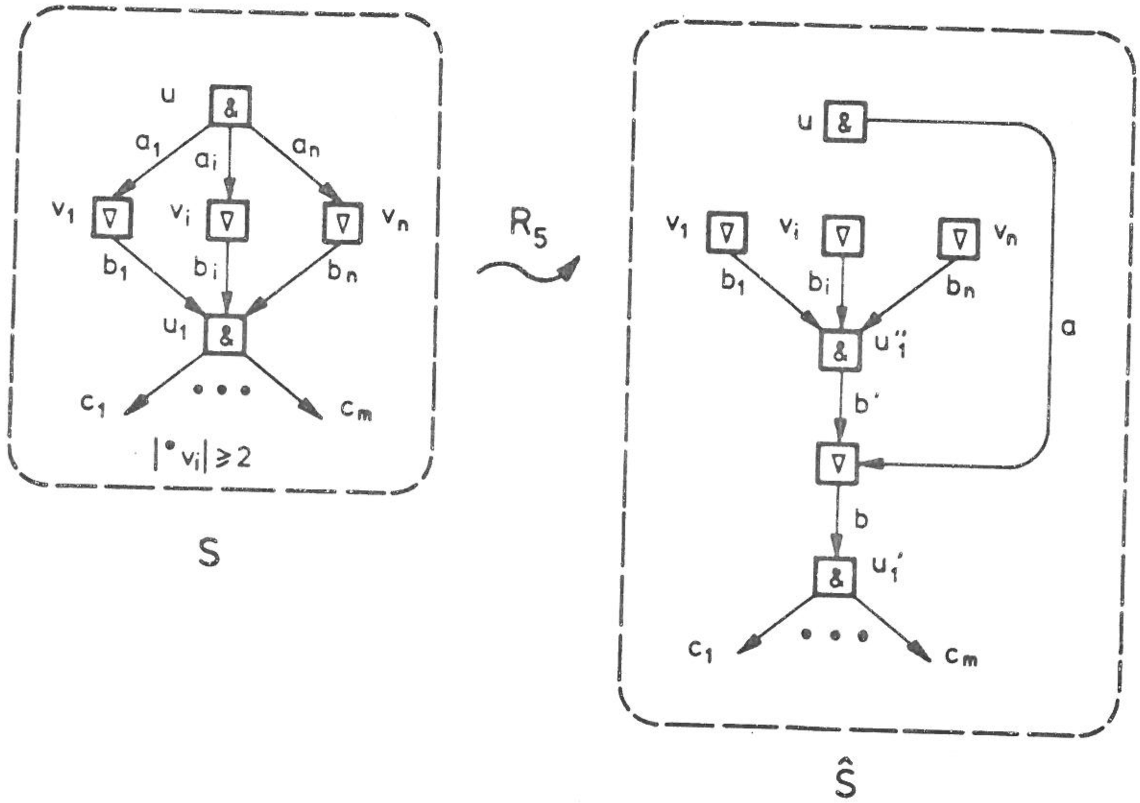

u
&
a1
ai
an
v1
vi
vn
b1
bi
bn
u1
&
c1
cm
|•vi| ≥ 2
S
R5
u
&
v1
vi
vn
b1
bi
bn
&
u1''
a
b'
b
&
u1'
c1
cm
Ŝ

Fig. 6.3

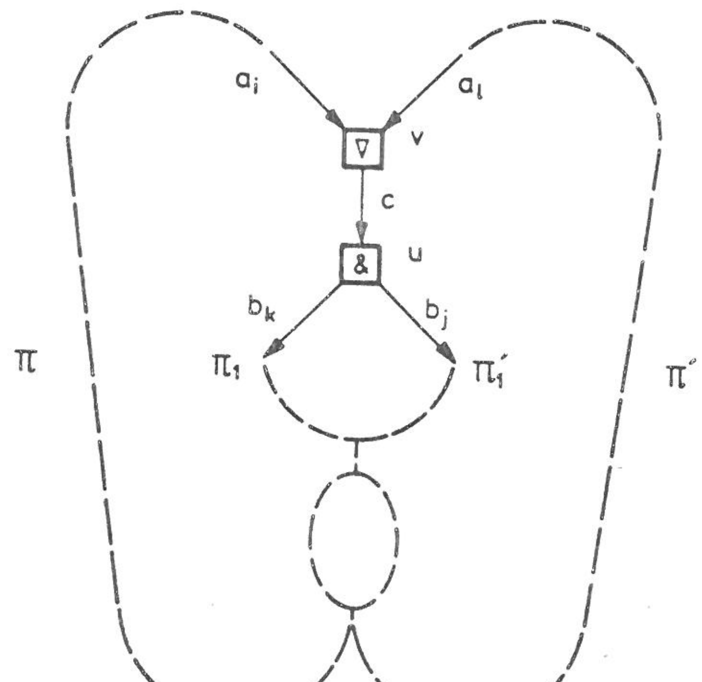

ai
al
v
c
&
u
bk
bj
π
π1
π1'
π'

Fig. 6.4

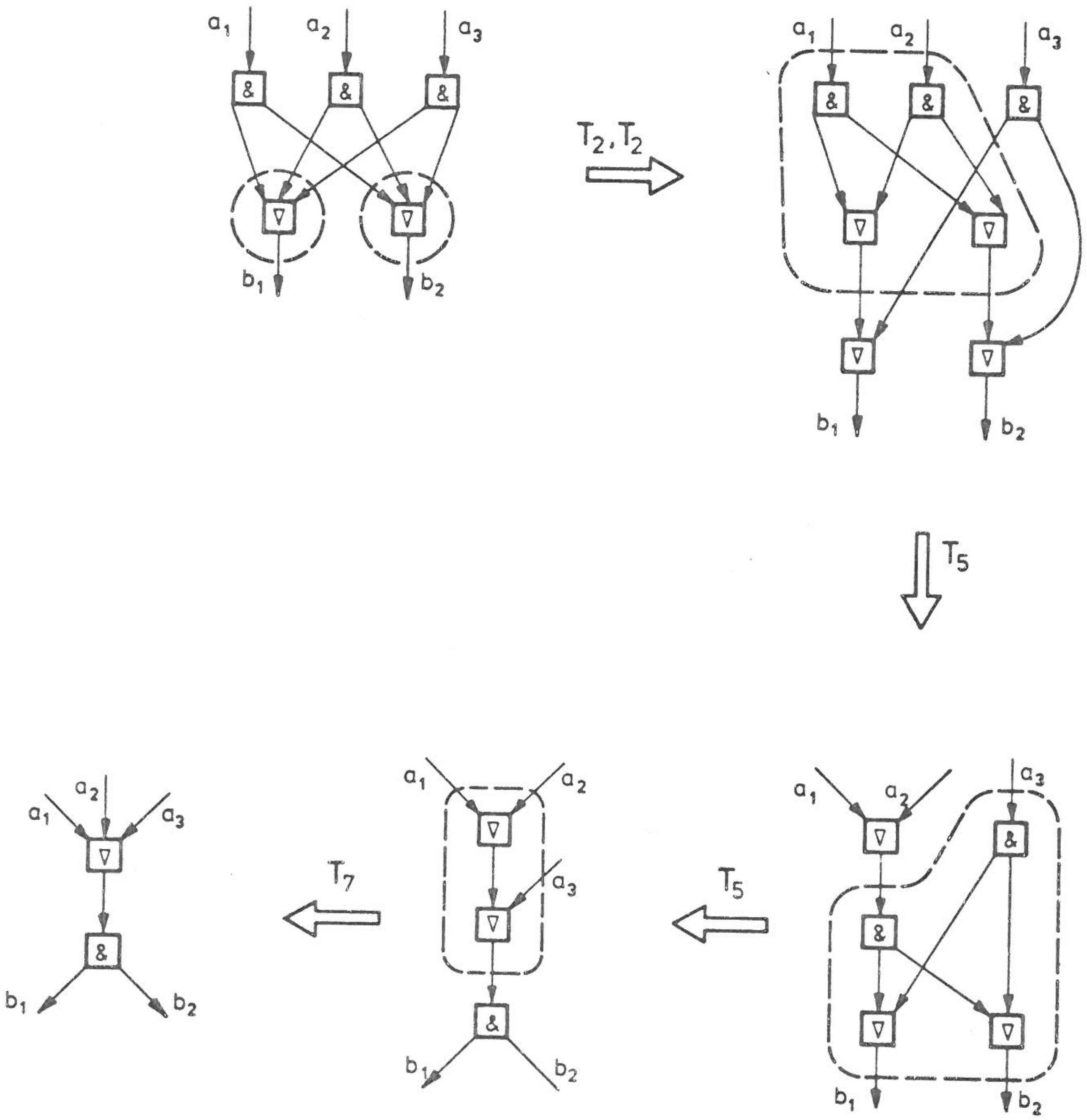


Fig. 6.5

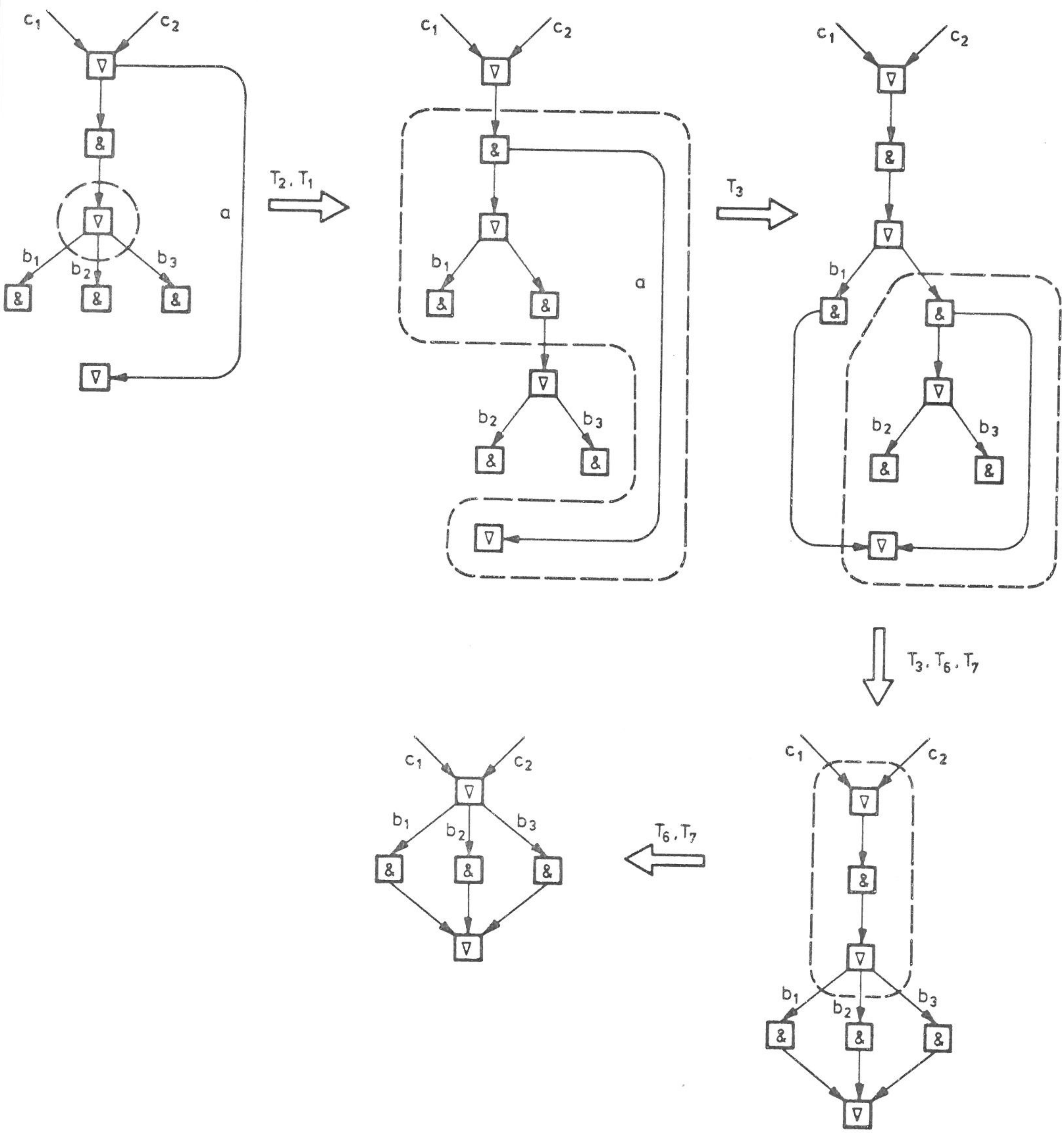


Fig. 6.6

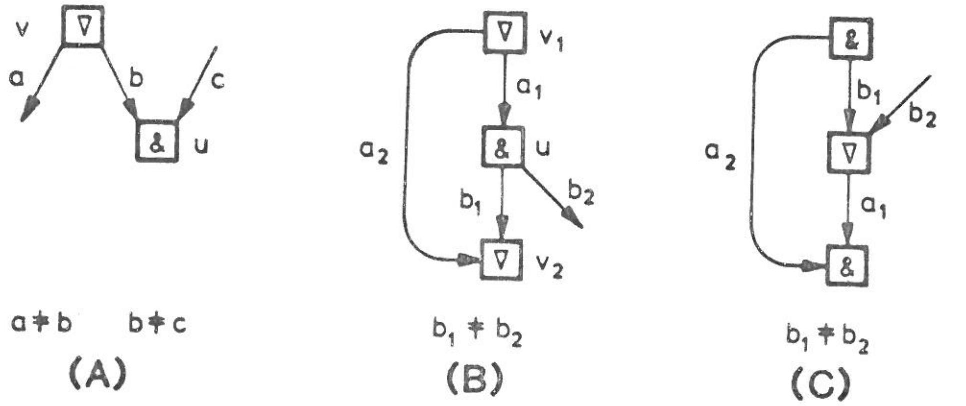


Fig. 6.7

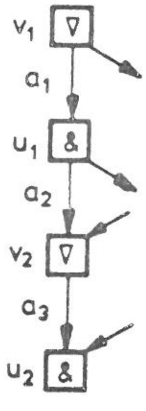


Fig. 6.8

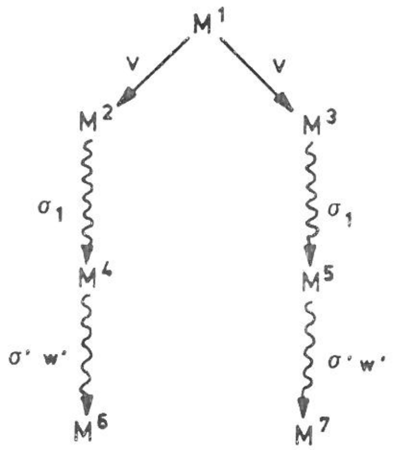


Fig. 6.9

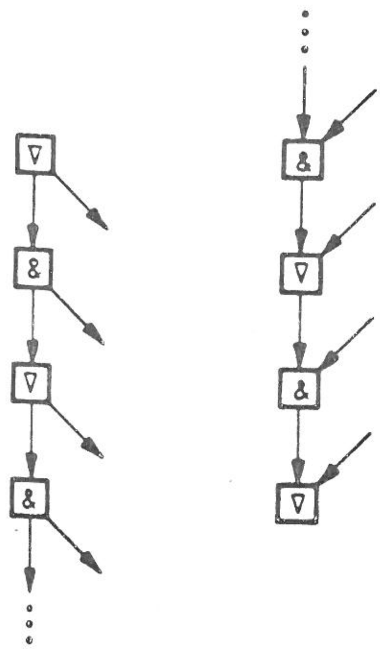


Fig. 6.10

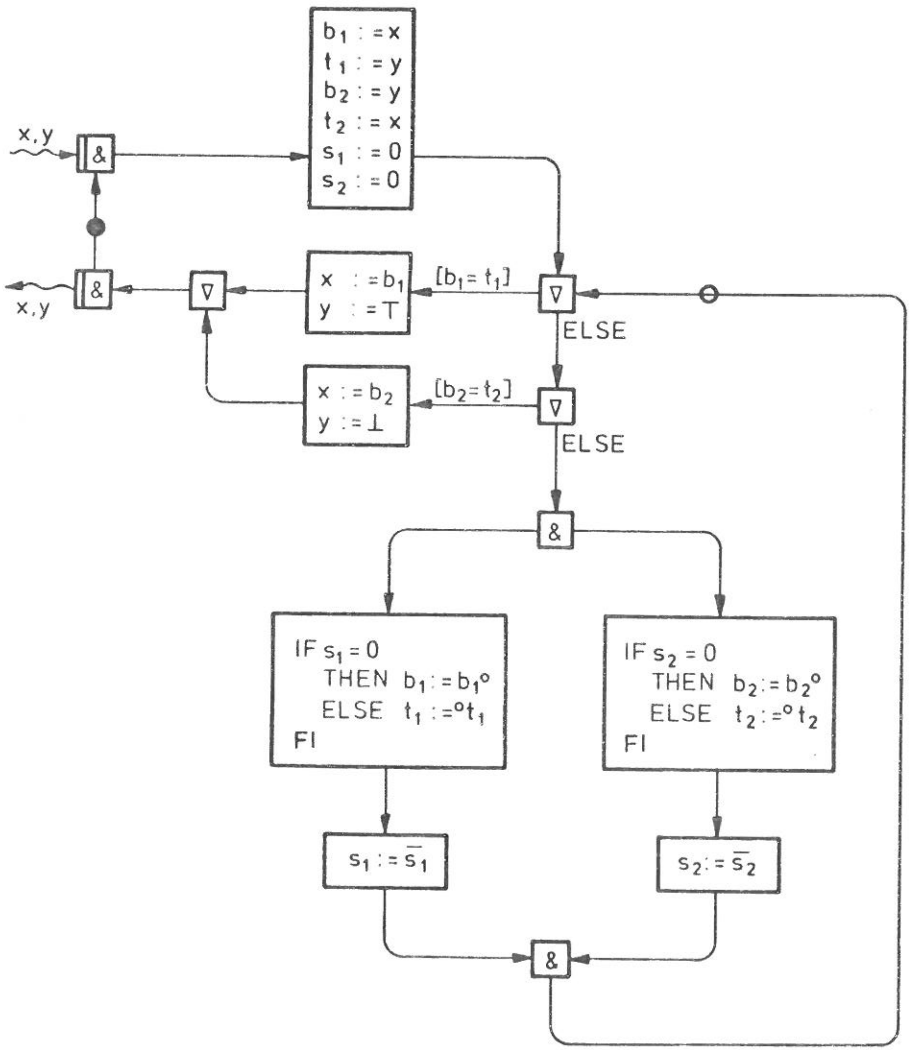

x, y
&
b1 := x
t1 := y
b2 := y
t2 := x
s1 := 0
s2 := 0
x, y
&
∇
x := b1
y := T
[b1 = t1]
∇
ELSE
x := b2
y := ⊥
[b2 = t2]
∇
ELSE
&
IF s1 = 0
THEN b1 := b1°
ELSE t1 := °t1
FI
IF s2 = 0
THEN b2 := b2°
ELSE t2 := °t2
FI
s1 := s̄1
s2 := s̄2
&

Fig. 7.1

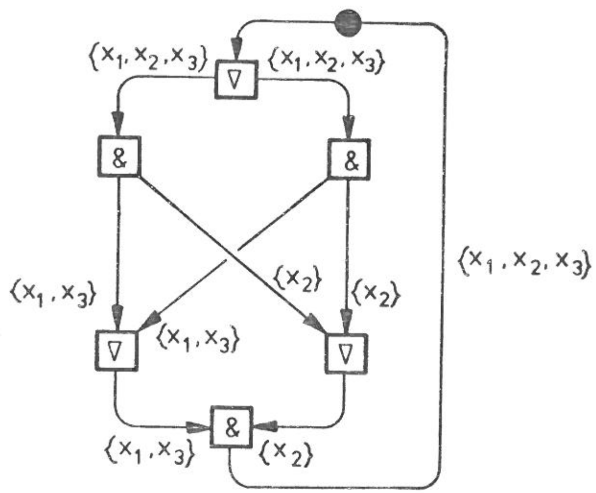


Fig. 7.2

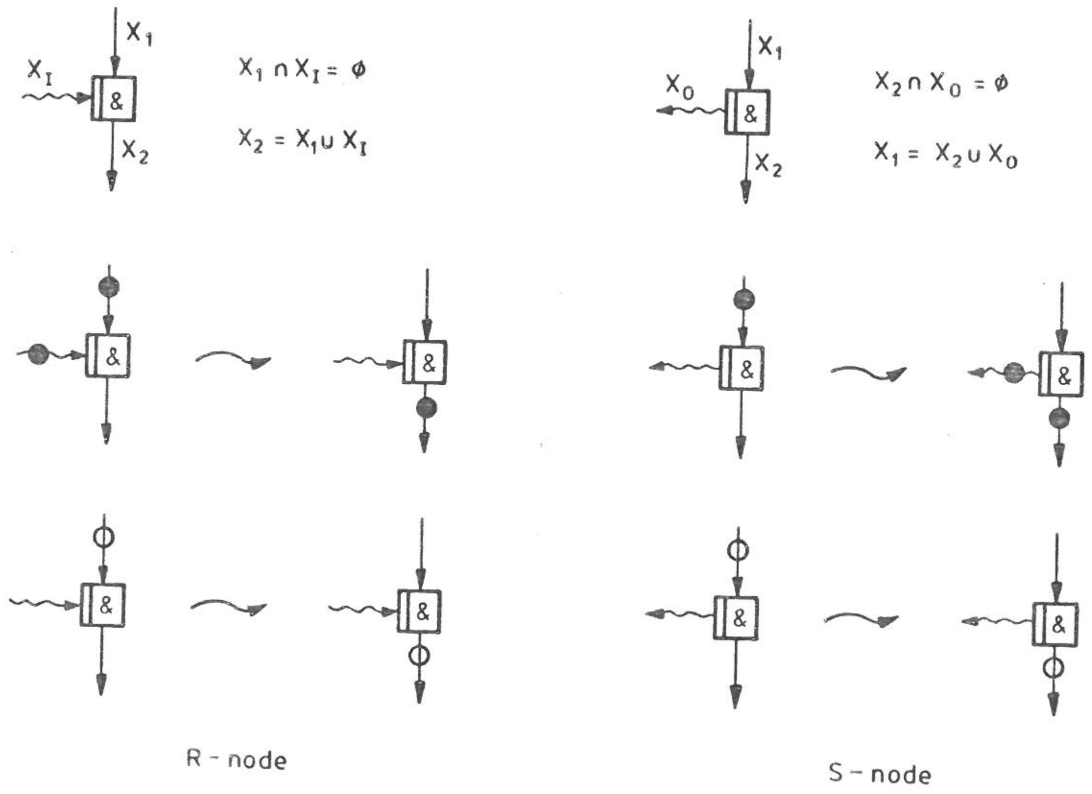
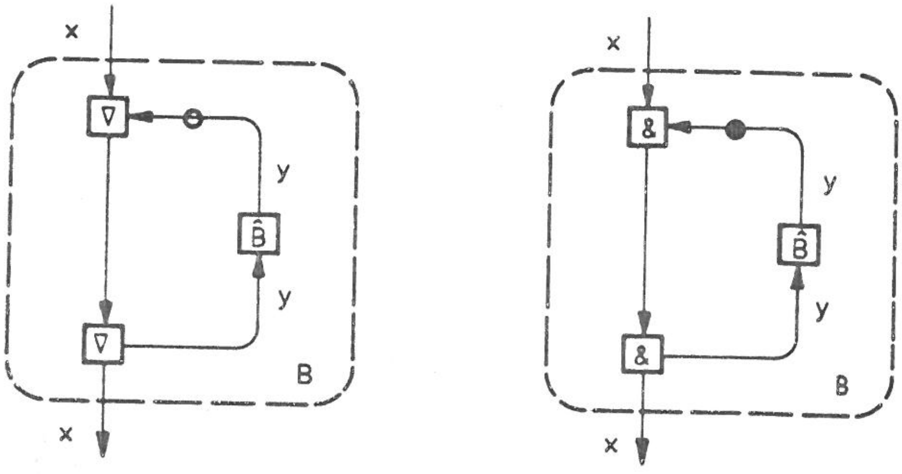


Fig. 7.3



(a) (b)

Fig. 7.4

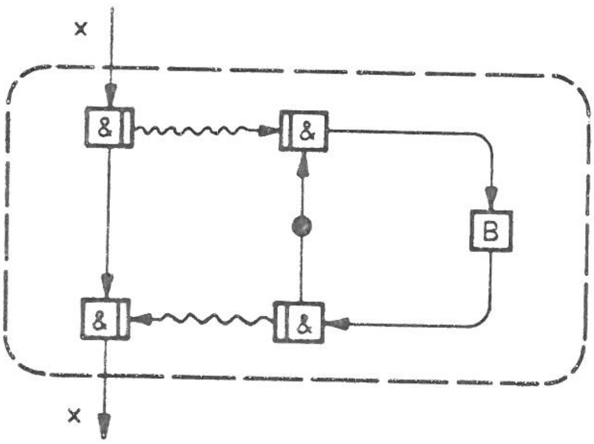


Fig. 7.5