Technical Report

UCAM-CL-TR-139
ISSN 1476-2986

Number 139

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Petri net theory: a survey

Paul R. Manson

June 1988

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

ISSN 1476-2986

Contents

Chapter 1

Introduction

The intense interest in concurrent (or "parallel") computation over the past decade has given rise to a large number of languages for concurrent programming (e.g. Ada[56], CCS[38], CSP[23], MultiLisp[22], etc) representing many conflicting views of concurrency.

The discovery that concurrent programming is a significantly more complex task than sequential programming has prompted considerable research into determining a tractable and flexible *theory* of concurrency, with the aim of making concurrent processing more accessible, and indeed the wide variety of concurrent languages merely reflects the many different *models* of concurrency which have also been developed.

One of the most important models of concurrency, the so-called *interleaving model* views a system's behaviour over a period of time as a partially-ordered sequence of "events" [33]. Intuitively, this means that one observer of a system may see some events occur in a different order to that which another observer sees. Specifically, if event α occurs in some system, followed by events β and γ (which occur very close together), after which an event δ occurs, then an observer of the system could report either of the two event sequences $\alpha\beta\gamma\delta$ or $\alpha\gamma\beta\delta$, depending simply on which of the events β and γ they noticed first. Notwithstanding this uncertainty, it is clear that event α occurred *first* in the sequence, and that event δ occurred *last*; all that we can say about β and γ is that they occurred "at about the same time", or in effect, that they occurred *concurrently*, insofar as an observer of the system could see them as occurring in either order [38].

If events β and γ above, did, in fact, occur at *precisely* the same instant, then the interleaving model has flattened what was a concurrent (or "composite") event $(\beta \parallel \gamma)$ into one of the sequential interleaved sequences $\beta\gamma$ or $\gamma\beta$. Although this simplifies much of the analysis of "pseudo-concurrent" systems, it seems unfortunate to have to dispense with real concurrency of behaviour. One model which does allow truly concurrent events is the *Petri Net* [42,47], and it is the aim of this report to present a summary of the varying research which has been performed on the Petri Net model.

The remainder of the report is structured as follows:

- Chapter 2 introduces Petri Nets and discusses their behaviour and interpretation. The relationship of Petri Nets to other models of concurrency is also considered briefly.

- Chapter 3 defines and discusses several restrictions and extensions of the Petri Net model, shows how they relate to basic Petri Nets and explains why they have been of importance historically.

- Chapter 4 presents a survey of the analysis methods applied to Petri Nets in general, and also in more detail for some of the net models introduced in Chapter 3.
- Chapter 5 concludes the discussion.

The Notation used throughout this report, including that pertaining to multisets, multirelations and partially ordered sets, is defined in the Appendices. Although care has been taken to acknowledge all those who have contributed to the field of Petri Net Theory, apologies are extended to any whose work has not been adequately acknowledged, or whose contribution has been misinterpreted by the author.

Chapter 2

Petri Nets

Petri Nets were designed for modeling and understanding systems in which concurrency is present, and although they have often been restricted to modeling merely an interleaving semantics (see for example [41]), they are fully capable of elegantly expressing truly concurrent event occurrences.

As abstract machines, Petri Nets lie somewhere between Finite Automata and Context-Sensitive/Turing machines, and thus represent a compromise between the simplicity and well-understood behaviour of an NFA and the power and complexity of a Turing machine [26]. The Petri Net model has itself been extended and restricted at various times in order to make the model more expressive or more manageable.

2.1 Basic Definitions

For the purposes of formal mathematical analysis, Petri Nets are best viewed as abstract mathematical structures, but an interpretation of the structure also permits powerful intuition as to the behaviour of a particular net. The abstract model to be presented below will later be provided with such an interpretation.

Definition: An *Unmarked Petri Net*[1] is a 4–tuple $(P, T, pre, post)$ where

P is a nonempty set of *places*,
T is a nonempty set of *transitions*, with $T \cap P = \phi$,
pre and *post* are multirelations from T to P, called the *pre* and *post* condition maps, respectively,

which satisfies the restrictions

$$\forall t \in T.\ \exists p \in P. \qquad pre(p,t) > 0 \ \text{ or } \ post(p,t) > 0$$
$$\forall p \in P.\ \exists t \in T. \qquad pre(p,t) > 0 \ \text{ or } \ post(p,t) > 0$$

□

The axioms require merely that no place or transition be isolated in the net. It should be noted that this is a rather weaker condition than is often used.

[1]The basic Petri Nets as presented here are sometimes called *Place/Transition* (or P/T) *Nets* in the literature.

Definition: If $N = (P, T, pre, post)$ is an Unmarked Petri Net, let $X_N = (P \cup T)$ denote the set of *elements* of N. Say that N is a *finite* net iff X_N is a finite set. □

This report makes considerable use of multisets and multirelations over the sets P and T of places and transitions of Petri Nets. In particular, the following notation is used extensively throughout.

Notation: The expressions $\bullet A$ and $A\bullet$, where A is a multiset of transitions (i.e. $A \in \mu T$) denote the multisets of places corresponding to the multiset sums

$$\sum_{t \in T} A_t \cdot (\bullet t) \quad \text{and} \quad \sum_{t \in T} A_t \cdot (t\bullet) \ ,$$

respectively, where the expressions $\bullet t$ and $t\bullet$, for $t \in T$, denote the multisets of places in the *pre* and *post* relations for t, i.e.

$$\bullet t = \sum_{p \in P} pre(p,t) \cdot \hat{p} \quad \text{and} \quad t\bullet = \sum_{p \in P} post(p,t) \cdot \hat{p} \ ,$$

respectively, where $\hat{p}$ is the singleton multiset consisting of one p-component only.

The expressions $\bullet A$ and $A\bullet$ are defined similarly when A is a multiset of places (i.e. $A \in \mu P$). □

As an aid to human understanding, Petri Nets are often represented as bipartite graphs of places (circles) and transitions (boxes) with directed arcs between those nodes in the *pre* and *post* multirelations. The places of a Petri Net correspond, intuitively, to the potential (distributed) state(s) of the Net, and such state(s) may be changed by the transitions of the Net, which correspond to the possible events which may occur (perhaps concurrently).

Example: The Unmarked Petri Net $N = (P, T, pre, post)$ where the sets of places and transitions are $P = \{a, b, c\}$, $T = \{\alpha, \beta\}$, and the only nonzero components of *pre* and *post* are:

$$\begin{array}{ll} pre(a,\alpha) = 1 & post(a,\alpha) = 1 \\ pre(c,\beta) = 1 & post(b,\alpha) = 2 \\ pre(b,\beta) = 1 & post(c,\beta) = 1 \end{array}$$

may be represented graphically as shown in figure 2.1. As an example of the $\bullet A, A\bullet$ notation, the following expressions

$$\begin{array}{lllll} A = \{\alpha\} & \Longrightarrow & \bullet A = \{a\}, & A\bullet = \{a,b,b\}, & \text{and} \\ A = \{\alpha,\beta\} & \Longrightarrow & \bullet A = \{a,b,c\}, & A\bullet = \{a,b,b,c\} & \end{array}$$

are valid for this net. □

The Unmarked Net of figure 2.1 is an example of an *uninterpreted net*, as no interpretation has been imposed on its nodes or arcs. As a net is an abstract mathematical object, any such interpretation is unnecessary for the purposes of mathematically analysing the net, but interpretation is very important for human interaction with nets. Thus it is often desirable to informally associate an interpretation with a net, purely for human convenience, as shown in figure 2.2, which interprets the net of figure 2.1 as a variation on the classical *producer-consumer* system.

The transitions α and β are interpreted as *produce* and *consume* events, respectively; the place b acts as a *buffer* between the producer and consumer processes, and the places a and c represent the state of each process when it is *producing* or *consuming*, respectively. The dynamics of this particular net will be be explained in section 2.3.

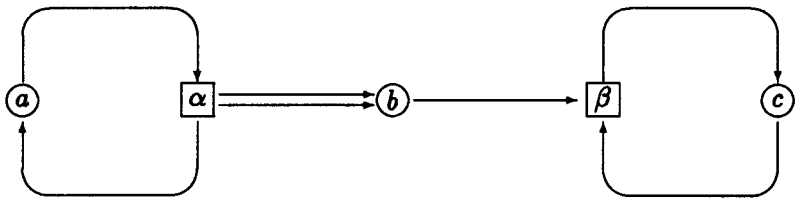

Figure 2.1: An Unmarked Petri Net

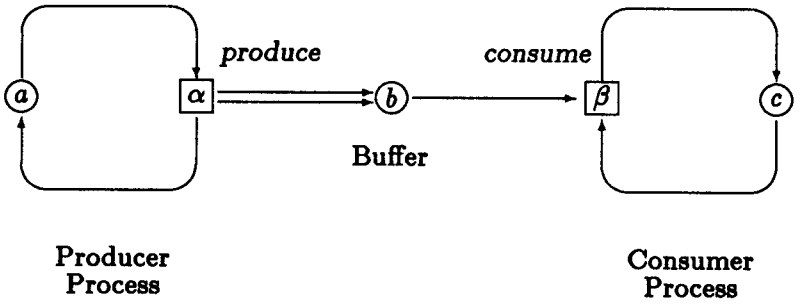

Figure 2.2: An interpreted net

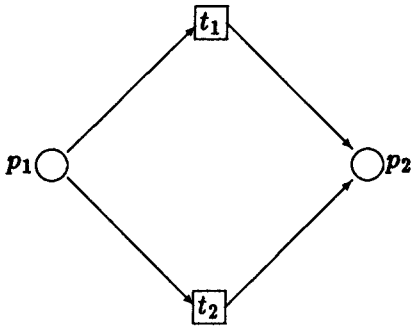

Figure 2.3: A net which is not simple

2.2 Some Structural Subclasses of Unmarked Petri Nets

Several simple subclasses of the basic Petri Net Model have proven to be of interest in the past, for various reasons. Some analysis techniques are easier to apply if the Petri Net model is restricted slightly, or if only a certain form of Net is considered. Several of the more historically significant of these subclasses are presented in this section.

Definition: An Unmarked Petri Net $N = (P, T, pre, post)$ is said to be *pure* if and only if $\forall x \in X_N$. $^\bullet x$ and $x^\bullet$ have no nonzero components in common (i.e. $^\bullet x \cap x^\bullet = \mathbf{0}$, the null multiset). □

In other words, a net is pure if no connected pair of elements (a place and a transition) of the net form a self-loop, i.e. there are no place/transition pairs of the form ○⇄□.

Definition: An Unmarked Petri Net $N = (P, T, pre, post)$ is called *simple* iff

$$\forall x, y \in X_N.\ (^\bullet x = {}^\bullet y \ \text{ and } \ x^\bullet = y^\bullet) \Longrightarrow x = y.$$

□

A Net is thus simple if there are no *equivalent* nodes, where two nodes are equivalent in the sense that they are connected to exactly the same set of other nodes, and in exactly the same way.

Example: The two transitions t_1 and t_2 of the net of figure 2.3 are equivalent, and thus the net is not simple. The producer-and-consumer net of figure 2.1 is, however, simple. □

Definition: An Unmarked Petri Net $N' = (P', T', pre', post')$ is a *subnet* of another Unmarked Net $N = (P, T, pre, post)$, denoted $N' \subseteq_{Net} N$, iff

$$\begin{aligned} P' &\subseteq P, \\ T' &\subseteq T \end{aligned}$$

and

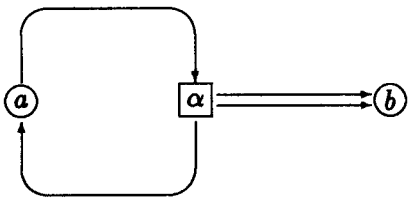

Figure 2.4: A Subnet of the producer-consumer net

$pre' : T' \rightarrow_\mu P'$ is the projection $\Pi^{T,P}_{T',P'}$ *pre* which assigns elements of *pre* to *pre'* whenever $(p,t) \in P' \times T'$.
$post' : T' \rightarrow_\mu P'$ is the projection $\Pi^{T,P}_{T',P'}$ *post* which assigns elements of *post* to *post'* whenever $(p,t) \in P' \times T'$.

□

Essentially, a subnet is a subset of elements of another net, with all arcs between those nodes in the subset intact and no other arcs present. The notion of a subnet is an important one when "building" a large net out of smaller component nets.

Example: The net of figure 2.4 is a subnet of the producer-consumer net (figure 2.1), which retains the places $P = \{a, b\}$, transitions $T = \{\alpha\}$. The only nonzero components of *pre* and *post* are $pre(a,\alpha) = 1$, $post(a,\alpha) = 1$, and $post(b,\alpha) = 2$. □

Definition: The *dual* of an Unmarked Petri Net $N = (P,T,pre,post)$ is the 4-tuple $\hat{N} = (\hat{P}, \hat{T}, \hat{pre}, \hat{post})$ where $\hat{P} = T$, $\hat{T} = P$, $\hat{pre} = post$, and $\hat{post} = pre$. □

Intuitively, $\hat{N}$ is N with its places changed to transitions and its transitions to places. Some basic results concerning the dual of a net are captured in the following.

Lemma 1 *Let $N = (P,T,pre,post)$ be an Unmarked Petri Net. Then*

1. *$\hat{N}$, the dual of N, is also an Unmarked Petri Net,*
2. *The dual of $\hat{N}$ is N, and*
3. *If N' is another Unmarked Net, with $N' = (P',T',pre',post')$, then $N \subseteq_{Net} N'$ iff $\hat{N} \subseteq_{Net} \hat{N}'$.*

Proof:

1. From the definition of dual, $\hat{N} = (\hat{P},\hat{T},\hat{pre},\hat{post})$ is obviously a 4-tuple with the correct structure. It remains merely to check the two restrictions for Unmarked Nets. Since N was an Unmarked Net, neither T nor P contain any unconnected elements, so neither do $\hat{T} = P$ or $\hat{P} = T$, and hence $\hat{N}$ is an Unmarked Net.

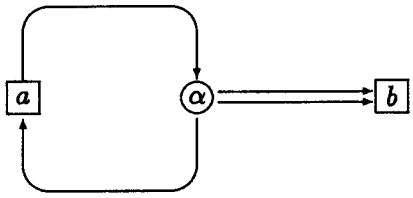

Figure 2.5: The dual of a net

2. Obvious, by repeated application of the definition of dual.
3. $\Longrightarrow$. If $N \subseteq_{Net} N'$ then by the definition of a subnet, $P \subseteq P', T \subseteq T'$ and *pre* & *post* are pre' & $post'$ projected onto $T' \times P'$. The definition of dual means that

   $\hat{P} \subseteq \hat{P'}$ [since $\hat{P} = T$ and $\hat{P'} = T'$],
   $\hat{T} \subseteq \hat{T'}$ [since $\hat{T} = P$ and $\hat{T'} = P'$],
   $\hat{pre}$ is a projection of $\hat{pre'}$ [since $\hat{pre} = post$ and $\hat{pre'} = post'$],
   $\hat{post}$ is a projection of $\hat{post'}$ [since $\hat{post} = pre$ and $\hat{post'} = pre'$],

   and thus $\hat{N}$ is a subnet of $\hat{N'}$.
   $\Longleftarrow$. Since the dual of $\hat{N}$ is N (from 2, above), the above argument with $\hat{N}$ for N gives the reverse implication.

□

Example: The dual of the (sub)net shown in figure 2.4 is illustrated in figure 2.5, and corresponds to the net with place $P = \{\alpha\}$, transitions $T = \{a, b\}$, and whose only nonzero components of *pre* and *post* are $pre(\alpha, a) = 1$, $pre(\alpha, b) = 2$ and $post(a, \alpha) = 1$. □

2.3 The Dynamic Behaviour of Nets

Unmarked Petri Nets as presented above comprise merely a *static* structure; there is no facility for modeling any form of dynamic behaviour, let alone concurrency. In this section the notions of marking and Petri Net dynamics will be introduced, along with a characterisation of the reachable "states" of a net.

Definition: Let $N = (P, T, pre, post)$ be an unmarked Petri Net. A *marking* M of N is a nonnull multiset of places (ie. $M \in \mu P$). □

A marking thus associates a nonnegative integer with each place of a Petri Net; this marking determines its distributed *state*, in the sense of "state" as applied to finite automata. The fact that a particular component M_p of a marking $M \in \mu P$ is nonzero ($M_p > 0$) is interpreted as meaning that M_p *tokens* of information are currently resident in place $p \in P$ of the net.

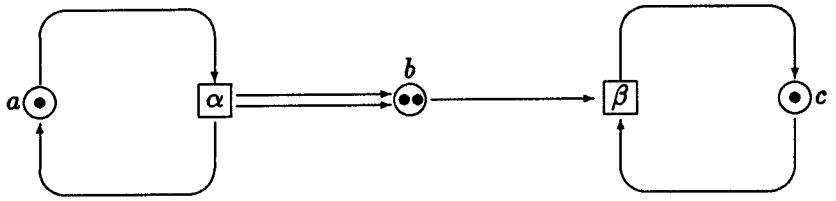

Figure 2.6: A Marked Petri Net

Graphically, these tokens are represented as dots • on the circle representing each place in the net, with M_p dots on each place $p \in P$. (If M_p is too large to conveniently represent as dots, inscribing the *number* M_p on the place suffices)[2].

Example: Figure 2.6 illustrates the *Producer-and-Consumer* Petri Net of figure 2.2, decorated with tokens according to the marking $M \in \mu\{a, b, c\}$, where $M_a = 1, M_b = 2$, and $M_c = 1$. The fact that $M_b = 2$ may be interpreted as meaning that the *buffer* in the producer-and-consumer system currently contains two tokens of information, ready to be consumed. □

The definition of Petri Nets (or simply "nets") to be used in the remainder of this report may now be stated:

Definition: A *Petri Net* N is a 5-tuple $N = (P, T, pre, post, M_0)$ where $(P, T, pre, post)$ is an Unmarked Petri Net and $M_0 \in \mu P$ is a marking of N called its *initial marking*. □

The subclasses of Unmarked Nets discussed in the previous section may be applied to the underlying (unmarked) net component of a (marked) Petri Net. The notion of a subnet is often extended to marked nets by requiring that those places which remain in the subnet have the same initial marking as they did in the original (super)net. It should be noted that the subnet of a (marked) Petri Net may not be a (marked) Petri Net, however it will always be at least an Unmarked Net.

The dynamic behaviour of a Petri Net arises by considering the transition system it determines, given the following definition of a net's transition relation.

Definition: Let M and M' be markings of a Petri Net $N = (P, T, pre, post, M_0)$. Let $A \in \mu T$ be a finite multiset of transitions. The *transition relation* for N is defined as

$$M \xrightarrow{A} M' \quad \text{iff} \quad {}^\bullet A \leq M \text{ and } M' = M - {}^\bullet A + A^\bullet$$

□

[2]Note that Petri Nets are sometimes defined in the literature to include a *place capacity* function which restricts the number of tokens which may reside in a given place. This could, for example, be used to specify a capacity for the buffer b in the producer-consumer example, thus enabling the modeling of fixed-size data structures via nets. This report will not address nets having this restriction, each place thus being considered as having unbounded capacity.

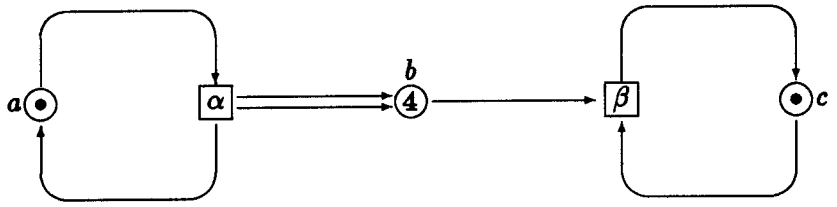

Figure 2.7: The New Marking for the Producers/Consumers Net

Notation: The construct $M \xrightarrow{A} M'$ is called an *event* in this report, and encompasses the notion of $(\beta \parallel \gamma)$ being a composite event as mentioned in the introduction. When it is clear from context that the subject of discussion is a multiset A of transitions, A may also be called an event, independent of the markings M and M'. The composition of two (possibly composite) events $A_1, A_2 \in \mu T$ is denoted $A_1 \parallel A_2$ and is defined to be the multiset sum $A_1 + A_2$. □

The event $M \xrightarrow{A} M'$ denotes that the transitions of A may *occur*[3] (concurrently) when the net is in the state (i.e. marked with) M, and this (multiple) transition occurrence leads N into the state (marking) M'.

Graphically, the interpretation of an event $M \xrightarrow{A} M'$, for $A \in \mu T$, is that the tokens representing marking M are removed from the net and those of marking M' are placed onto the net, with the intuition being that the tokens of each component place in $\bullet A$ have "moved into" the transition, and new tokens have been produced by the transition, one per component place in $A\bullet$.

Example: If, in the net of figure 2.6, the singleton event $A = \hat{\alpha}$ occurred, the resulting net would be marked as in figure 2.7. Intuitively, what happened was that the transition α took one token from the place a as specified in its *pre* condition map, and produced 2 tokens for place b and another for place a. This operation corresponds to a "produce" action in a classical producer-consumer system, where the producer produces two tokens of information each time around the "produce" cycle. The new marking of place b is now 4, as indicated in the diagram. □

Definition: A marking M of a net $N = (P, T, pre, post)$ is a *reachable marking* of N iff

$$M_0 \xrightarrow{A_0} M_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} M_n = M$$

is a sequence of events $A_i \in \mu T$ connecting a sequence of markings M_i of N via the transition relation of N, where M_0 is the initial marking of N. This may also be denoted $M_0 \xrightarrow{A_0 A_1 \cdots A_{n-1}} {}^*M$, or, if the events A_i are not of consequence, simply as $M_0 \longrightarrow^* M$. □

[3] Sometimes "fire" in the literature.

A marking M being "reachable" thus means that from a net's initial state (marking) it is possible for an observer to see some sequence of events occur, after which the net will be in the state M.

Notation: Write $N : M$ to denote that M is a reachable marking of a net N, and $N : M \xrightarrow{A} M'$ to denote that M is a reachable marking of N and $M \xrightarrow{A} M'$ is an event of N. Write simply $M \xrightarrow{A}$ to denote $M \xrightarrow{A} M'$, for some marking $N : M'$, if the identity of this resultant marking M' is not of interest. When $N : M \xrightarrow{A}$, meaning that the event A may occur at a marking M, say that the transitions of A *have concession*[4] at marking M, and also that A *has concession* at M. □

Definition: The *reachability relation* between markings of a net $N = (P, T, pre, post, M_0)$ is defined to be the relation $\rightsquigarrow \subseteq \mu P \times \mu P$ given by

$$M_1 \rightsquigarrow M_2 \text{ iff } \exists A \in \mu T : \; M_1 \xrightarrow{A} M_2.$$

The class of reachable markings of N at a given marking M is the set $\mathcal{R}_N(M) = \{M' \in \mu P \mid M \rightsquigarrow^* M'\}$, called the *reachability class* of N, where $\rightsquigarrow^*$ is the reflexive, transitive closure of $\rightsquigarrow$. Where the initial marking M_0 is understood, $\mathcal{R}_N$ may denote $\mathcal{R}_N(M_0)$; where the net involved is clearly N, $\mathcal{R}(M)$ may denote $\mathcal{R}_N(M)$. □

An inportant class of concurrent systems are those (such as Operating Systems) which continually perform a sequence of operations; this class of Petri Nets are called *cyclic*.

Definition: A net $N = (P, T, pre, post, M_0)$ is called *cyclic*[5] iff $M \rightsquigarrow^* M'$ (or $M' \in \mathcal{R}_N(M)$) for all reachable markings M and M' of N. i.e. all markings are reachable from all other reachable markings, and, in particular, the initial marking is reachable from any other reachable marking of the net. □

By way of an example, the producers-and-consumers net of figure 2.6 is cyclic.

2.4 Basic Net-Dynamics Situations

The following definitions describe several situations which are basic to the dynamic behaviour of Petri Nets; *viz.* those involving sequential, conflicting, concurrent, contacting and confused events. These situations characterise the behaviour of (parts of) Petri Nets, and represent a reasonably broad spectrum of behaviour from purely sequential to fully concurrent.

2.4.1 Sequential Behaviour

Definition: Let $N = (P, T, pre, post, M_0)$ be a Petri Net and let $A_1, A_2 \in \mu T$ be two events of N. If, at a marking $N : M$ of N, it is the case that the event $M \xrightarrow{A_1} M'$ may occur, but the event $M \xrightarrow{A_2}$ may *not* occur, and it is the case that $M' \xrightarrow{A_2}$ may occur, then A_1 and A_2 are said to be *in sequence*. □

Example: In the net of figure 2.8 the events $A_1 = \hat{t}_1$ and $A_2 = \hat{t}_2$ are in sequence at the initial marking $M = \hat{p}_1$. The occurrence of A_1 *must* precede that of A_2. □

---

[4] Or "are firable"

[5] Sometimes "reversible" in the literature.

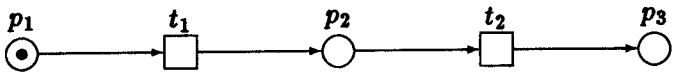

Figure 2.8: A Net illustrating sequence

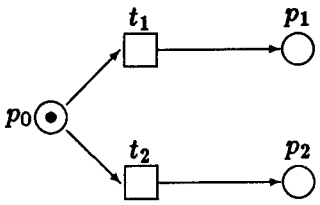

Figure 2.9: A Net illustrating conflict

2.4.2 Conflict

Definition: Let $N = (P, T, pre, post, M_0)$ be a net with $A_1, A_2 \in \mu T$ two events of N. A_1 and A_2 are said to be *in conflict* at a marking M of N if either may occur at M (i.e. $N : M \xrightarrow{A_1}$ and $N : M \xrightarrow{A_2}$), but *both* may *not* occur simultaneously at M (i.e. $N : M \xrightarrow{A_1 \| A_2}$ is *not* an event of N. □

Example: In the net of figure 2.9, the events $A_1 = \hat{t}_1$ and $A_2 = \hat{t}_2$ are in conflict at the indicated marking $M = \hat{p}_0$, since the event $A_1 \parallel A_2 = \{t_1, t_2\}$ is not an event of the net at M. This conflict may be *resolved* in favour of either A_1 or A_2 by a nondeterministic decision to perform the transition(s) specified by one of the two conflicting events. For example, the conflict might be resolved in favour of A_1, in which case the transition t_1 would occur, causing the token in place p_0 to move to place p_1. When a net contains no conflict situations, it is said to be *deterministic.* □

A property related to the absence of conflict in a marked Petri Net is that of *persistence*, which is defined as follows;

Definition: [34] A Petri Net is called *persistent* iff for all events $A_1 = \hat{t}_1, A_2 = \hat{t}_2$ with $t_1, t_2 \in T, t_1 \neq t_2$ and any reachable marking M, it is the case that $M \xrightarrow{A_1}$ and $M \xrightarrow{A_2} \Longrightarrow M \xrightarrow{A_1 \| A_2}$, i.e. if A_1 and A_2 are enabled at a reachable marking then the occurrance of one cannot disable the other. □

Persistent nets have several important properties which will be discussed in Chapter 4. The class of Persistent nets is a proper superset of the class of conflict-free nets.

2.4.3 Concurrent Behaviour

Definition: Let $N = (P, T, pre, post, M_0)$ be a net and let $A_1, A_2 \in \mu T$ be events of N. Say that A_1 and A_2 can occur *concurrently* at a marking $N : M$ iff $N : M \xrightarrow{A_1 \| A_2}$ is an event of N. □

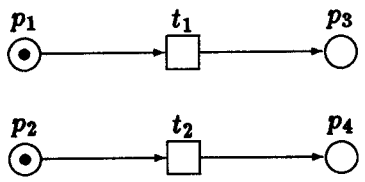

Figure 2.10: A Net illustrating concurrency

Example: The events $A_1 = \hat{t}_1$ and $A_2 = \hat{t}_2$ of the net in figure 2.10 may occur concurrently at the indicated marking $M = \{p_1, p_2\}$. They may occur completely independently of each other, so the event sequences of the net may be either $A_1 \parallel A_2$ (concurrent occurrence), $A_1 A_2$ (A_1 occurred first) or $A_2 A_1$ (A_2 occurred first). □

2.4.4 Contact

Definition: A net $N = (P, T, pre, post, M_0)$ is said to be *contact-free* iff for all reachable markings M of N and for all events $A \in \mu T$ of N, ${}^\bullet A \leq M \implies A^\bullet \cap M = \mathbf{0}$. □

Example: The nets of figures 2.8, 2.9 and 2.10 are all contact-free, but the producers-and-consumers net of figure 2.6 is not (since the event $A = \hat{\alpha}$ is in contact with the marking shown). □

Contact-freedom is a property of *safe* Petri Nets (see section 3.1) and is of mainly historical interest.

2.4.5 Confusion

The final dynamic situation of Petri Nets to be illustrated at this point is that of *confusion*, which is the result of a combination of concurrency and conflict. An example of a confused situation is the net of figure 2.11, where it is not clear whether or not a conflict needed to be resolved in going to the new state (marking) of the net in figure 2.12; that is, in going from marking $M_1 = \{p_1, p_4, p_5\}$ to marking $M_2 = \{p_2, p_6\}$ via some combination of the transitions t_1 and t_5. Two observers could report that either

1. An event $\hat{t}_5$ consisting of the single transition t_5 occurred first, without being in conflict with any other event, and then an event $\hat{t}_1$ consisting of transition t_1 occurred, or
2. The transitions t_1 and t_5 occurred concurrently, as an event $\{t_1, t_5\}$, or
3. An event $\hat{t}_1$ occurred first, and as a result, the events $\hat{t}_1$ and $\hat{t}_5$ got into conflict. This conflict was resolved in favour of $\hat{t}_5$, which then occurred.

More formally, confusion is defined as follows:

Definition: Let $N = (P, T, pre, post, M_0)$ be a Petri Net with M a reachable marking of N, and let $A \in \mu T$ be an event of N such that $N : M \xrightarrow{A} M'$ for some other $N : M'$. The

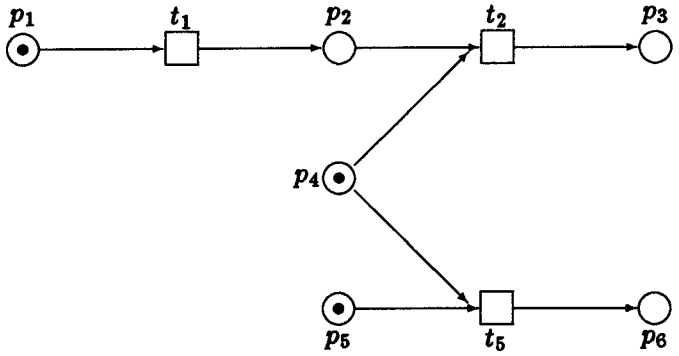

Figure 2.11: A confused situation

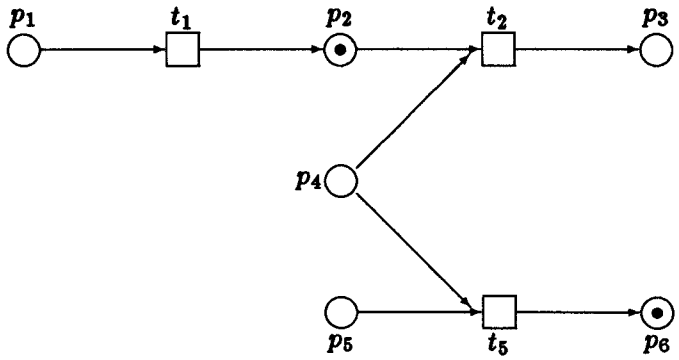

Figure 2.12: The confusion resolved

conflict set of A at M, denoted $cfl(A,M)$, is $\{A' \in \mu T \mid N : M \xrightarrow{A'}$ and not $N : M \xrightarrow{A\|A'}\}$, the set of all events that are in conflict with A at M. □

Definition: Let $N = (P,T,pre,post,M_0)$ be a net with reachable marking M and let $A_1, A_2 \in \mu T$ be two disjoint events of N (i.e. $A_1 \cap A_2 = 0$) such that $N : M \xrightarrow{A_1\|A_2}$. Say that N is *confused* at M iff $cfl(A_1,M) \neq cfl(A_1,M')$ where $N : M \xrightarrow{A_2} M'$. □

In other words, confusion arises if $A_1 \parallel A_2$ is an event at M and the occurrence of A_2 alters the conflict set of A_1. In the above example, the conflict set of the event $\hat{t}_5$ is $cfl(\hat{t}_5,M) = \phi$ at the marking $M = \{p_1,p_4,p_5\}$, but $cfl(\hat{t}_5,M') = \{\hat{t}_2\}$ at the marking $M' = \{p_2,p_4,p_5\}$ which results from the occurrence of the event $\hat{t}_1$. Thus N is confused at M.

Section 3.3, which presents the class of *Free-Choice* Petri Nets, will highlight the ways in which confusion makes analysis of a net's behaviour considerably more complex, and why it is desirable to avoid net-confusion wherever possible. Confusion is analagous to the *glitch* problem in communications protocols, and this well-known problem has been a source of theoretical difficulty for many years.

2.5 Petri Nets in Relation to Other Models of Concurrency

In order that Petri Nets may be related to different models of concurrency (such as Event Structures[52], CCS[38], CSP[23], etc), they have been provided with a notion of *morphism*[6] and cast in a categorical framework, as is briefly described in this section.

Definition: Let $N = (P,T,pre,post,M_0)$ and $N' = (P',T',pre',post',M_0')$ be two Petri Nets. A *morphism* from N to N' is defined to be a pair (η,β) with $\eta : T \to_* T'$ and $\beta : P \to_\mu P'$ such that

$$\beta M_0 = M_0' \quad \text{and} \quad \forall A \in \mu T.\ \ {}^\bullet(\eta A) = \beta({}^\bullet A) \text{ and } (\eta A)^\bullet = \beta(A^\bullet)$$

□

Remark: η is a partial function, linearly extended to multirelations, i.e. the matrix of η satisfies $\eta_{t,t'} \leq 1$, and $(\eta_{t,t'} = 1 \text{ and } \eta_{t,t''} = 1) \Longrightarrow t' = t''$, for transitions t, t' and t''. □

The definition of morphism preserves the dynamic behaviour of nets, in the sense of the following theorem.

Theorem 2 *[54] Let $(\eta,\beta) : N \to N'$ be a morphism of Petri Nets. Then β preserves the initial marking (i.e. $M_0' = \beta(M_0)$) and if $N : M \xrightarrow{A} M'$ then $N' : \beta M \xrightarrow{\eta A} \beta M'$.* □

Example: When a morphism $(\eta,\beta) : N \to N'$ arises from inclusions $\eta : T \subseteq T'$ and $\beta : P \subseteq P'$, N is a subnet of N'. For the subnet (figure 2.4) of the net in figure 2.1, the morphism is given by $\eta : \alpha \longmapsto \alpha$ and $\beta : a \longmapsto a$, $\beta : b \longmapsto b$. □

[6]Note that the definition of morphism used here is that of Winskel[54], as opposed to the older definition of Petri Net morphism as used in, for example, [11], which did not respect the dynamic behaviour of nets.

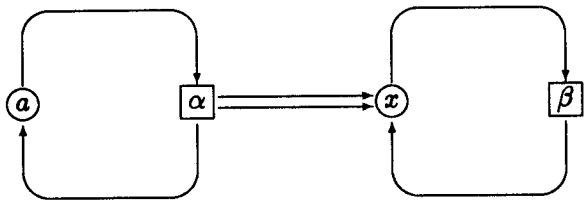

Figure 2.13: A *folding* of the Producers-and-Consumers Net

Example: The net of figure 2.13 is a *folding* of the producers-and-consumers net, where a folding is a morphism which maps no adjacent nodes to the same image. This particular folding represents the morphism

$$\eta : \alpha \longmapsto \alpha \qquad \beta : a \longmapsto a$$
$$\eta : \beta \longmapsto \beta \qquad \beta : b \longmapsto x$$
$$\beta : c \longmapsto x$$

The places b and c have been folded together (about the axis of transition β) into a new place x in the folding. □

Definition: Let **Net** be the *category* of Petri Nets with morphisms as defined above, in which the composition of morphisms $(\eta_0, \beta_0) : N_0 \to N_1$ and $(\eta_1, \beta_1) : N_1 \to N_2$ is defined as $(\eta_0\eta_1, \beta_0\beta_1) : N_0 \to N_2$ and the identity morphism for a net N has the form $(\mathbf{1}_T, \mathbf{1}_P)$ where $\mathbf{1}_T$ and $\mathbf{1}_P$ are the identities on transition-sets and place-multisets, respectively. □

Given this category of nets, the various categorical constructions such as product and coproduct have been defined as in [54,51,55,50], and the relationships between Petri Nets and other models of parallel computation have been formalised via this mechanism.

Within the category **Net** itself, it is desirable to relate Petri Nets and morphisms between them to Milner's notion of *bisimulation*[38], which represents a very useful and well-understood characterisation of *similarity* for concurrent systems.

Definition: Let P be the set of possible concurrent processes in a given programming language and let T be a set of events which may occur in P, where the fact that a process $p \in P$ executes an event $t \in T$ and is transformed into a new process $p' \in P$ is denoted $p \xrightarrow{t} p'$. This notation is extended to composite events A over μT and sequences of (composite) events σ over $(\mu T)^*$.

A relation $\rho \subseteq P \times P$ is called a *bisimulation* iff for all $(p, q) \in \rho$ and $\sigma \in \mu T^*$,

1. Whenever $p \xrightarrow{\sigma} p'$ then for some $q', q \xrightarrow{\sigma} q'$ and $(p', q') \in \rho$.
2. Whenever $q \xrightarrow{\sigma} q'$ then for some $p', p \xrightarrow{\sigma} p'$ and $(p', q') \in \rho$.

□

Processes p and q are said to be *observational equivalent* (written $p \sim q$) iff there exists a bisimulation ρ such that $(p,q) \in \rho$. The above definition is phrased for general concurrent systems, and may be adapted specifically for Petri Nets, as detailed below.

In order to obtain a definition of bisimulation for Petri Nets, it is first necessary to define functions on sequences of transitions. Let $\eta : T \rightarrow_* T'$ be an injective (1-1) partial function from one set of transitions to another, linearly extended to multisets. Then η may be extended to a function $\eta : T^* \rightarrow_* T'^*$ in the canonical way (where T^* is the set of sequences over T, including the empty sequence ϵ_T), i.e.

$$\begin{aligned} \eta(\epsilon_T) &= \epsilon_{T'} \\ \eta(vt) &= \eta(v)\eta(t) \qquad\qquad \text{for } v \in T^*, t \in T. \end{aligned}$$

Furthermore, η^{-1} is a relation in $T' \times T$ and can, in the following way, be extended to a *function* $\eta^{-1} : T'^* \rightarrow T^*$:

$$\begin{aligned} \eta^{-1}(\epsilon_{T'}) &= \epsilon_T, \\ \eta^{-1}(wt) &= \begin{cases} \eta^{-1}(w) & \text{if } t \notin \eta(T) \\ \eta^{-1}(w)\eta^{-1}(t) & \text{if } t \in \eta(T) \end{cases} \end{aligned}$$

for all $w \in T'^*, t \in T'$.

The above definitions allow the following adaptation of bisimulation for Petri Nets; it is a modification of that presented in [3].

Definition: Let $N = (P,T,pre,post,M_0)$ and $N' = (P',T',pre',post',M_0')$ be two Petri Nets. A morphism $(\eta,\beta) : N \rightarrow N'$ is called a Petri Net *bisimulation* relating N and N' iff the following conditions hold:

1. $\beta M_0 = M_0'$,
2. Suppose that $\beta M_1 = M_1'$, for some reachable $N : M_1$ and $N' : M_1'$;
   (a) Whenever $M_1 \xrightarrow{\sigma} M_2$ with $\sigma \in (\mu T)^*$, and $N : M_2$ then $M_1' \xrightarrow{\eta\sigma} \beta M_2$ is an event (sequence) in N', and
   (b) Whenever $M_1' \xrightarrow{\sigma'} M_2'$ with $\sigma' \in (\mu T')^*$ and $N' : M_2'$ then there exists $N : M_2$ with $M_2' = \beta M_2$ such that $M_1 \xrightarrow{\eta^{-1}\sigma'} M_2$ is an event (sequence) of N.

In this case, N and N' are said to be *Observational equivalent* with respect to the (bisimulation) morphism (η,β). □

Note that provided η and β are invertible in the sense defined previously, the definition of morphism gives Observational Equivalence for all pairs of nets possessing such a morphism and respecting the additional bisimulation conditions.

Because η is an injective partial function, N' has at least as many transitions as N; the extra transitions of N' (i.e. those in $T' \setminus \eta(T)$) should be thought of as silent internal actions of N' (corresponding to τ actions in CCS). Best has show that the definition of bisimulation for Petri Nets as presented here preserves both safeness and liveness properties of nets, as will be seen in Chapter 4.

Quite apart from the (rather abstract) vehicle of morphism, explicit correspondence between Petri Nets and other models of concurrency has been made by several authors; in particular, classes of Petri Nets have been mapped to CCS [15,40], COSY [2] and CSP [16] with considerable success.

Chapter 3

Restricted and Extended Models

The basic Petri Net model, as defined in the previous chapter, has often been restricted in order to simplify analysis and provide deeper understanding. These simplified Petri Net models have been of considerable use in the development of Net Theoretical results. At the same time, when actually *applying* Nets to practical modeling problems, it has proved necessary to develop more powerful (essentially, more *concise*) representations of nets, and thus extensions have also appeared. This chapter presents a discussion of several of these diverging models and interests.

3.1 Safe Nets

One of the most theoretically-important Petri Net models is the *safe net*, which essentially simplifies what may happen dynamically in a Petri Net by restricting the static structure of the Net. Formally, a safe net is defined as follows.

Definition: A Petri Net $N = (P, T, pre, post, M_0)$ is *safe* if and only if $pre(p,t) \leq 1$ and $post(p,t) \leq 1$ for all $t \in T, p \in P$, and $M_p \leq 1$ for all reachable markings M and places $p \in P$. □

For safe nets, all multiset and multirelation components are binary (i.e. either 0 or 1), and thus *pre* and *post* may be considered as mere relations over $P \times T$, while any reachable marking $N : M$ is simply a set over P, and any event A with $N : M \xrightarrow{A}$ is a set over T.

Safe nets are often referred to as "Condition/Event" (or "C/E") nets in the literature, since a place may be regarded as representing a *condition* which either holds (with multiplicity 1) or does not hold at a marking. Similarly, *events* (transitions) either occur (with multiplicity 1) or do not occur when one marking is transformed into another. In safe nets, all markings are subsets of P, the set of places, and often a marking $M \subseteq P$ of a safe net is called a *case* of the net in the literature, but this report will retain the "marking" terminology.

Example: The Nets of figures 2.8, 2.9, 2.10 and 2.11 are all safe, but those of figures 2.6 and 2.7 are not. The net of figure 3.1 is a slightly more complex, though still safe, net. □

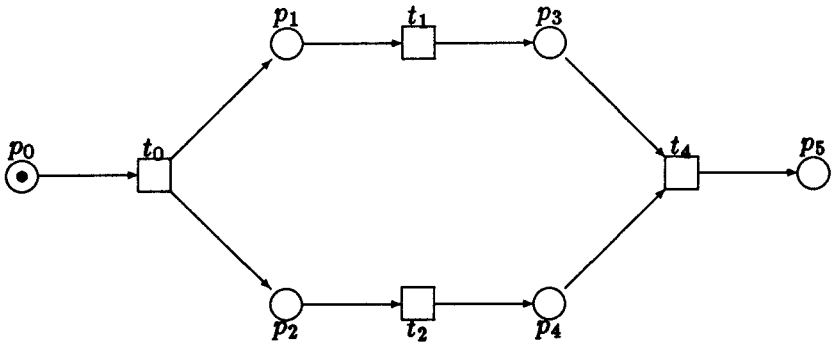

Figure 3.1: A safe net

Proposition 3 *Let $N = (P,T,pre,post,M_0)$ be a safe net with $N : M$ a reachable marking. If $N : M \xrightarrow{A} M'$ for a marking M' and finite multiset A of events then M, M' and A, $\bullet A$ and $A\bullet$ are all sets. Furthermore, $M \xrightarrow{A} M'$ iff*

$$(\forall t \in A.\ \bullet t \subseteq M) \text{ and } (\forall t,t' \in A.\ t \neq t' \Longrightarrow \bullet t \cap t\bullet = \emptyset) \text{ and } M' = (M \setminus \bullet A) \cup A\bullet. \quad \square$$

The Petri Net notions of sequence, conflict, concurrency, contact and confusion all restrict to safe nets, as do morphisms. There are, however, several useful properties of safe nets which permit significantly more powerful analysis than do the less-restricted Petri Nets, as will be seen in the remainder of this section, and in the following section on Occurrence nets.

Notation: Since, for a safe net $N = (P,T,pre,post,M_0)$, the multirelations $pre : T \rightarrow_\mu P$ and $post : T \rightarrow_\mu P$ are merely relations $pre \subseteq P \times T$ and $post \subseteq P \times T$, it is convenient to combine them into a single relation $F_N \subseteq (P \times T) \cup (T \times P)$ defined as $F_N = pre \cup post^{-1}$. Thus

$$\begin{array}{lll} pFt & \text{iff} & pre(p,t) \text{ for all } t \in T, p \in P, \text{ and} \\ tFp & \text{iff} & post(p,t) \text{ for all } t \in T, p \in P. \end{array}$$

and the safe net N may be manipulated as the 4-tuple $N = (P,T,F,M_0)$. Thus $\bullet t$ and $t\bullet$, for $t \in T$, may be expressed as sets in terms of F as follows:

$$\begin{array}{lcl} \bullet t & = & \{p \in P \mid pFt\}, \\ t\bullet & = & \{p \in P \mid tFp\}, \end{array}$$

and similarly for $\bullet A$ and $A\bullet$, where $A \subseteq T$ is a composite event. $\square$

The following definition and construction of P-completion is used in the calculation of synchronic distance for Occurrence nets (see section 3.2.3), but is defined on safe nets, and is thus presented at this point.

Definition: A safe net $N = (P,T,F,M_0)$ is *P-complete* iff for every pair (A_1,A_2) of nonempty events $A_1, A_2 \subseteq T$ there exists a place $p \in P$ with $\bullet p = A_1$ and $p\bullet = A_2$. $\square$

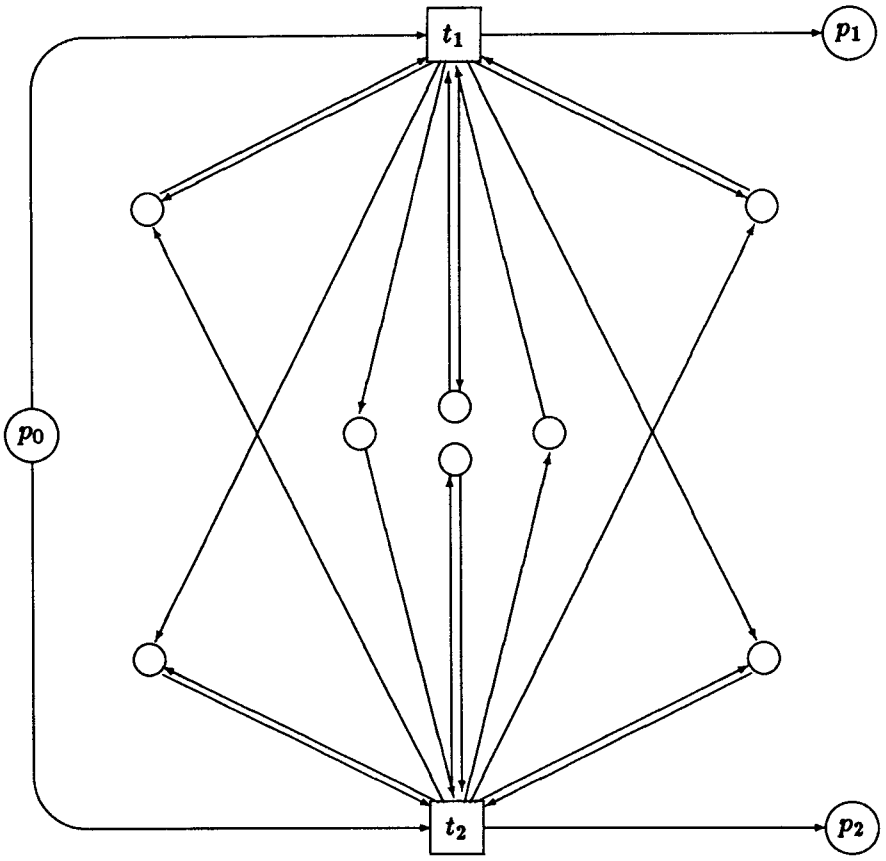

Figure 3.2: The P-completion of a net

Essentially, a net is P-complete if every pair of subsets of places in the net is connected by a single event. If a safe net $N = (P, T, F, M_0)$ is not P-complete, it is possible to construct its *P-completion*, $N' = (P', T, F', M_0)$, as follows:

$$\begin{aligned} P' &= P \cup Q, \text{ where} \\ Q &= \mathcal{P}(T) \times \mathcal{P}(T) \setminus (\{({}^\bullet p, p^\bullet) \mid p \in P\} \cup \{(x, y) \mid x, y \in \mathcal{P} \wedge x \neq \emptyset \wedge y \neq \emptyset\}) \\ F' &= F \cup \bigcup_{(A_1, A_2) \in Q} ((A_1 \times \{(A_1, A_2)\}) \cup (\{(A_1, A_2)\} \times A_2)) . \end{aligned}$$

This construction creates a great many new places, in general. For example, the P-completion of the net of figure 2.9 is illustrated in figure 3.2, which is unlabelled (save for the places and transitions of the original net) for clarity.

Safe nets with morphisms as described in section 2.5 form a subcategory of the category Net which possesses many agreeable properties, and although they will not be dwelt upon here, the references [54,51,55,50] make considerable use of this subcategory.

3.1.1 Facts in Safe Nets

The marked places (conditions) of a safe (C/E) net may be interpreted as determining a formula of propositional logic by treating each place name as a boolean variable, negating

the variables which correspond to unmarked places, and concatenating the variables into a formula with $\wedge$ operators. For example, the safe net of figure 3.1 has the reachable marking $M = \{p_1, p_4\}$, which corresponds to the formula

$$\neg p_0 \wedge p_1 \wedge \neg p_2 \wedge \neg p_3 \wedge p_4 \wedge \neg p_5.$$

Arbitrary well-formed formulae over the set P of variables may be tested for validity at a given marking of a safe net $N = (P, T, F, M_0)$ by substituting *TRUE* for each variable/place name which is currently marked and *FALSE* for each variable/place name which is not. Since conditions change as markings change, formulae will either be valid or not at each marking of the net. A formula which is *always* valid describes a *logical invariant* of the net. For example, the formula

$$\neg p_0 \Longleftrightarrow (((p_1 \vee p_3) \wedge (p_2 \vee p_4)) \vee p_5)$$

is a logical invariant of the safe net in figure 3.1.

There is a formula associated with each transition of a safe net, as defined in the following:

Definition: Let $N = (P, T, F, M_0)$ be a finite safe net with $t \in T$ a transition. Let ${}^\bullet t = \{p_1, \ldots, p_n\}$ and $t^\bullet = \{p'_1, \ldots, p'_m\}$. Then the formula associated with t, denoted $\xi(t)$, is the formula

$$(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \Longrightarrow (p'_1 \vee \cdots \vee p'_m).$$

If $t^\bullet = \emptyset$ then $\xi(t)$ is the formula $\neg(p_1 \wedge \cdots \wedge p_n)$, and if ${}^\bullet t = \emptyset$ then $\xi(t)$ is the formula $(p'_1 \vee \cdots \vee p'_m)$. $\square$

Lemma 4 *If $N = (P, T, F, M_0)$ is a finite safe net with $t \in T$ then for each reachable marking M of N, $\xi(t)$ is valid at M iff t does not have concession at M.*

Proof: The formula $\xi(t)$ is valid at a marking $M \subseteq P$ iff there exists a place $p_i \in {}^\bullet t$ with $p \notin M$ or there exists a place $p'_j \in t^\bullet$ with $p \in M$. This is true iff t does not have concession at M. $\square$

If a safe net is enlarged by the addition of ("dead") transitions which will never be enabled (and thus do not alter the net's behaviour), the formulae associated with these extra transitions will be logical invariants of the system. Such invariants (called *facts*) are represented graphically as $\boxplus$, and are denoted by a pair $t = (P_1, P_2)$ where $P_1, P_2 \subseteq P$ are the sets ${}^\bullet t$ and $t^\bullet$, respectively, where t is the introduced transition.

Example: The safe net of figure 3.3 represents two processes, each of which executes a critical region in exclusion by restricting the other from entering its critical region until the first process has exited theirs. The critical regions are represented by the places p_1 and p_2, and the fact that these places may not both have a token in the same marking may be expressed by the logical expression $\neg(p_1 \wedge p_2)$. This may be also be denoted by the addition of a fact t to the net, as shown in figure 3.4, where ${}^\bullet t = \{p_1, p_2\}$ and $t^\bullet = \emptyset$. $\square$

Theorem 5 *Let $N = (P, T, F, M_0)$ be a finite safe net and let ξ be an arbitrary well-formed formula of propositional logic over the set P of variables. ξ is valid over N if and only if there exist facts $t_n, \ldots t_n$ such that ξ is logically equivalent to $\xi(t_1) \wedge \cdots \wedge \xi(t_n)$.*

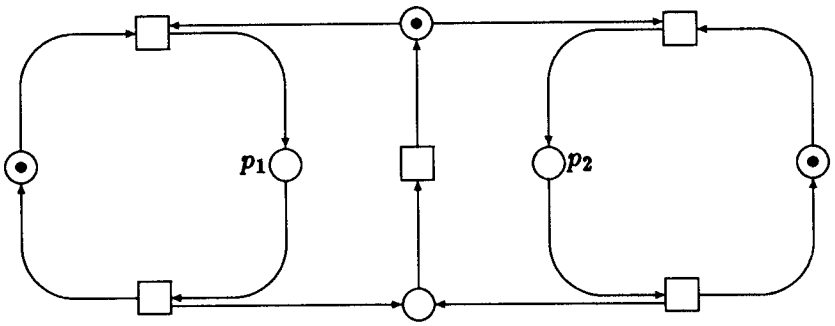

Figure 3.3: A safe net representing Critical-Regions

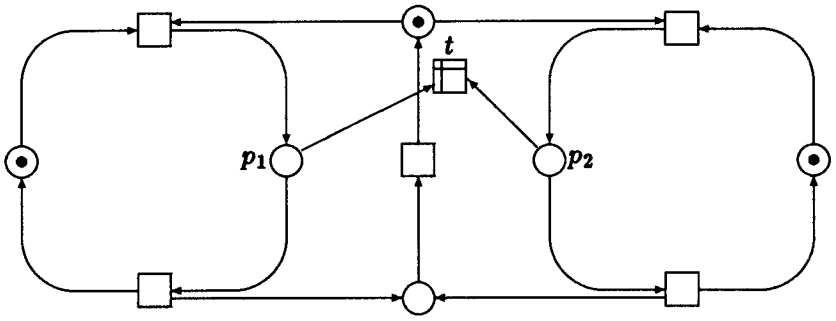

Figure 3.4: The Critical-Region net with a Fact

Proof: Each ξ may be transformed into a logically equivalent formula $\xi' = \xi_1 \wedge \cdots \wedge \xi_k$, where each ξ_i is a term of the form $\neg p_1 \vee \cdots \vee \neg p_r \vee p'_1 \vee \cdots \vee p'_s$ with $p_j, p'_j \in P$ (conjunctive normal form). Therefore, ξ_i is logically equivalent to a formula $\xi(t_i)$ with ${}^\bullet t_i = \{p_1, \ldots, p_r\}$ and $t_i{}^\bullet = \{p_1, \ldots p_s\}$.

Now, ξ is valid over N if and only if ξ' is valid over N, and this is true if and only if ξ_i is valid over N, for all $1 \leq i \leq k$, that is, if and only if $\xi(t_i)$ is valid over N, for all $1 \leq i \leq k$. This is true, finally, if and only if t_i is a fact of N, for all $1 \leq i \leq k$. □

Obviously, more advanced logical systems (such as temporal and modal logics) may be employed for reasoning about the behaviour of Petri Nets. The interested reader is directed to [5,45,44,49,51] for further details and references.

3.2 Occurrence Nets

Given a safe Petri Net, it is possible to "run" the net from the initial marking, resolving conflicts in an arbitrary fashion, and recording the non-sequential occurrences of events together with the resulting holdings of conditions during the run. Such a run is called a *process* of the safe net, and models a non-sequential stretch of history. The record of the process is itself a special form of safe net, called an *occurrence net*, and in addition to their relation to safe nets, occurrence nets possess several interesting properties of their own.

3.2.1 Basic Definitions

Definition: An unmarked, safe Petri Net $N = (P, T, F)$ is called an *occurrence net* if and only if

$$\begin{array}{lll} \text{(i)} & \forall x, y \in X_N.\ xF^+y \text{ iff } \neg(yF^+x) & \text{[N is cycle-free], and} \\ \text{(ii)} & \forall p \in P.\ (xFp \wedge yFp) \Longrightarrow x = y, \text{ and} & \\ & (pFx \wedge pFy) \Longrightarrow x = y. & \text{[Places are unbranched].} \end{array}$$

where $F = pre \cup post^{-1}$ is as defined for safe nets, and F^+ is the transitive closure of F. □

Example: The safe net of figure 3.1 is an occurrence net, with the exception of the token (marking) on place p_0. It is thus evident that occurrence nets are merely a special (i.e. more restricted) subclass of the unmarked safe nets. □

An occurrence net $N = (P, T, F)$ determines a partially ordered set $\langle X_N, \prec \rangle$ where $X_N = P \cup T$ are the elements of N and the relation $\prec$ is the transitive closure F^+ of F. The associated relations $\preceq$, $\succ$, $\succeq$ and $\prec\!\!\!\cdot$ are obtained as

$$\begin{aligned} \preceq &= \prec \cup \mathrm{id} = F^+ \cup \mathrm{id} = F^*, \\ \succ &= \prec^{-1}, \\ \succeq &= \succ \cup \mathrm{id}, \text{ and} \\ \prec\!\!\!\cdot &= F, \end{aligned}$$

and thus the relations $\underline{li}$ and $\underline{co}$, and the notions of discreteness and density of posets described in Appendix B also apply to occurrence nets.

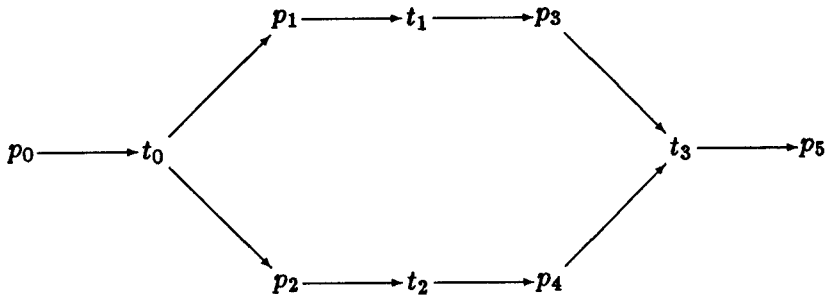

Figure 3.5: The poset determined by an occurrence net

Definition: A subset $S \subseteq P$ of places is called a *slice* of an occurrence net $N = (P, T, F)$ iff S is a *cut* of the poset $\langle X_N, \prec \rangle$ determined by N. □

Example: As the occurrence net of figure 3.1 determines the partially-ordered set illustrated in figure 3.5, the slices of the net are the cuts $\{p_1, p_2\}, \{p_1, p_4\}, \{p_3, p_2\}$ and $\{p_3, p_4\}$. For completeness, the remaining cuts of the poset are $\{p_1, t_2\}, \{p_3, t_2\}, \{p_2, t_1\}, \{p_4, t_1\}$ and $\{t_1, t_2\}$, while the lines of the poset are $\{p_0, t_0, p_1, t_1, p_3, t_3, p_5\}$ and $\{p_0, t_0, p_2, t_2, p_4, t_3, p_5\}$. □

The notion of a *process* relating an occurrence net and a safe net may now be formalised.

Definition: Let $N = (P, T, F)$ be a finite occurrence net and let $N' = (P', T', F', M_0)$ be a finite, contact-free safe net. A mapping $p : X_N \to X_{N'}$ is called a (finite) *process* (of N') iff

1. $\forall$ slices S of $N, p \mid S$ is injective and $M_0 \rightsquigarrow^* p(S)$,
2. $\forall t \in T.\ \ p(\bullet t) = \bullet p(t) \quad \wedge \quad p(t\bullet) = p(t)\bullet$.

□

Let $\pi(N)$ denote the set of finite processes of a finite, contact-free, safe net N.

Example: The confused net of figure 2.11 has (to within isomorphism) two finite processes, the occurrence nets of which are illustrated in figure 3.6. □

Note that the underlying poset $\langle X_N, \prec \rangle$ of an occurrence net $N = (P, T, F)$ determines the initial marking of N, precluding the need for an additional component M_0, since the initial marking may be viewed as comprising those places which are lower bounds of $\langle X_N, \prec \rangle$, i.e. M_0[1] may be constructed as $\{p \in P \mid \nexists\, p' \in X_N.\ \ p' \prec p\}$.

[1]Sometimes denoted $^\circ N$ in the literature.

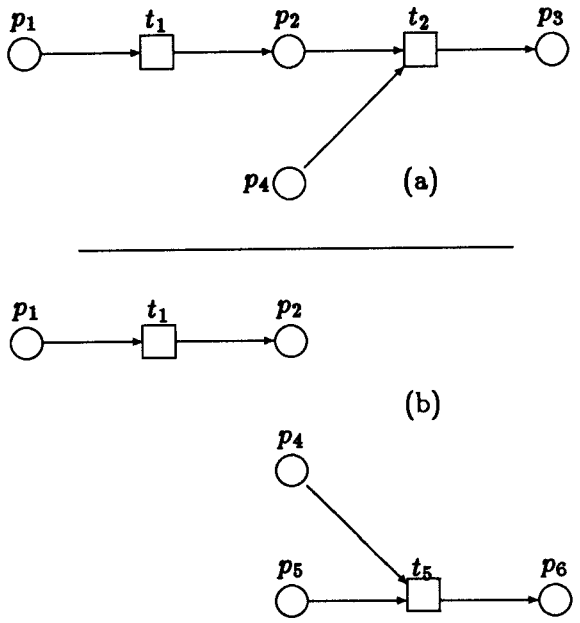

Figure 3.6: The processes of a safe net

Example: The initial marking of the confused net (figure 2.11) may be determined from the processes illustrated in figure 3.6. By observation, the places which are lower bounds are p_1, p_4 and p_5, and thus the initial marking of the net was $M_0 = \{p_1, p_4, p_5\}$. □

3.2.2 Properties of Occurrence Nets

Appendix B presents several of the more important properties which may be possessed by partially ordered sets, and many of these properties apply to Occurrence nets in particular. Clearly, the poset $\langle X_N, \prec \rangle$ associated with an occurrence net $N = (P, T, F)$ is *combinatorial*, but what may be said of its density properties?

Theorem 6 *If $N = (P, T, F)$ is an occurrence net then its associated poset $\langle X_N, \prec \rangle$ is N-dense.*

Proof: [9] Let $x, y, x', y' \in X_N$ such that

$$x \prec y \ \wedge \ x \prec x' \ \wedge \ y' \prec y \ \wedge \ (x \ \underline{co} \ y' \ \underline{co} \ x' \ \underline{co} \ y).$$

Since $\langle X_N, \prec \rangle$ is combinatorial, there exists a subset $\{x_1, x_2, \ldots x_n\} \subseteq X_N$ such that

$$x = x_1 \prec x_2 \prec \ \ldots \prec x_n = y.$$

Let $j \in \mathbb{N}$ with $1 \leq j \leq n$ be the smallest element of $\mathbb{N}$ such that $x_j \ \underline{co} \ x'$ (and j exists, since $y = x_n \ \underline{co} \ x'$). By the election of j it is clear that $x_{j-1} \ \underline{li} \ x'$, and as $x' \prec x_{j-1} \Longrightarrow x' \prec y = x_n$ causes a contradiction (since $x' \ \underline{co} \ x_n$) it follows that $x_{j-1} \prec x'$.

Similarly, both $x_{j-1} \prec y'$ and $x_{j-1} \succ y'$ lead to contradictions (since $x = x_1 \prec x_{j-1} \prec y'$ but $x = x_1 \ \underline{co} \ y'$, and $y' \prec x_{j-1} \prec x'$ but $y' \ \underline{co} \ x'$) so it must be the case that $y \ \underline{co} \ x_{j-1}$.

Clearly $x_{j-1} \in T$ and $x_j \in P$. Now both $x_j \prec y'$ and $x_j \succ y'$ lead to contradictions (since $x = x_1 \prec x_j \prec y'$ but $x = x_1 \ \underline{co} \ y'$, and $y' \ \underline{co} \ x_{j-1}$ but $x_j \in P$) and thus $x_j \ \underline{co} \ y'$.

Thus, letting $z = x_j$ gives $x \prec z \prec y \ \wedge \ (y' \ \underline{co} \ z \ \underline{co} \ x')$, and hence $\langle X_N, \prec \rangle$ is N-dense. □

Theorem 7 *Finite occurrence nets are K-dense.*

Proof: See [1,24]. □

3.2.3 Synchronic Distance

In this section it is assumed that all (safe and occurrence) nets are finite and contact-free. When considering events (or multisets of events) in a Net, it is often the case that important insights into the net's behaviour may be obtained by examining the way in which these events are related or synchronised. The notion of *synchronic distance* was developed [14,13,47] in an attempt to characterise the degree of event synchronisation in a safe net.

The procedure of P-completion which was introduced in section 3.1 creates many new places, and it is usually necessary to provide these places with initial tokens in addition to those of the original initial marking, in order to get the P-complete net to behave exactly the same way as it did before P-completion. Because some of the places constructed by P-completion must initially hold more that 1 token in order not to influence the system's behaviour, the P-complete net is no longer safe, but it is still a Petri net.

The maximal variance of the number of tokens on a place p which was constructed by P-completion is called the *synchronic distance* (denoted $\sigma(p)$) of this element, and similarly, $\sigma(A_1, A_2)$ denotes the synchronic distance of the place constructed by P-completion over events $A_1, A_2 \subseteq T$. When the events $A_1 = \{t_1\}$ and $A_2 = \{t_2\}$ are singleton events (merely transitions), write $\sigma(t_1, t_2)$ rather than $\sigma(\{t_1\}, \{t_2\})$.

Synchronic distance is useful in the design of nets, in detection of deadlocking or over-tightly-coupled subnets, and analysis of concurrent systems in general. It may be used to distinguish true concurrency from arbitrary interleaving, since it gives an upper bound on concurrency. If two events occur concurrently, their synchronic distance will be at least 2 (i.e. $(\alpha \parallel \beta) \Longrightarrow \sigma(\alpha, \beta) \geq 2$), but if they are simply alternating/interleaving then it is only certain that their synchronic distance will exceed 0 (i.e. $\alpha\beta$ or $\beta\alpha \Longrightarrow \sigma(\alpha, \beta) \geq 1$). Since synchronic distance is merely an upper bound on concurrency, a purely sequential system may have $\sigma \geq 2$, but if $\sigma = 1$ then events may *never* occur concurrently. Note that $\sigma = 0$ means that the events are coincident in both time and space (i.e. $\sigma(\alpha, \alpha) = 0$ for all events α).

Definition of Synchronic Distance

Before the actual synchronic distance function may be defined, another function is required for counting the occurrences of events in a process with respect to pairs of slices of an occurrence net.

Definition: Let $N = (P, T, F)$ be an occurrence net with S_1, S_2 slices of $\langle X_N, \prec \rangle$ and $T' \subseteq T$. The measure m is defined as

$$m(T', S_1, S_2) = \left|\{t \in T' \mid S_1 \prec t \prec S_2\}\right| - \left|\{t \in T' \mid S_2 \prec t \prec S_1\}\right|$$

where

$$t \prec S_i \quad \text{means that} \quad \forall s \in S_i.\ t \preceq s, \text{ and}$$
$$t \succ S_i \quad \text{means that} \quad \forall s \in S_i.\ t \succeq s.$$

□

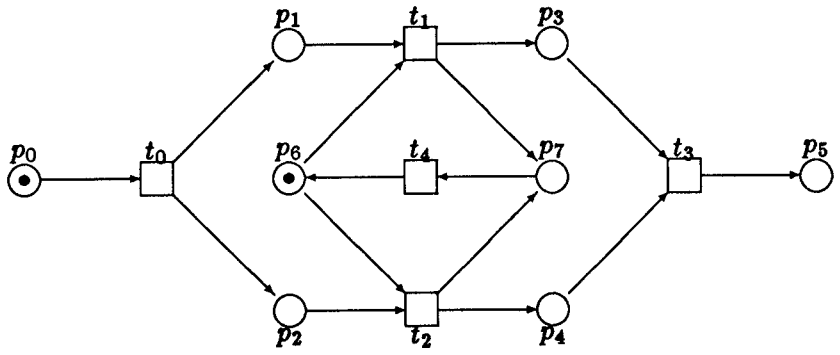

Figure 3.7: A safe net with concurrency restricted

Given the measure m, it is possible to define the variance of the occurrence of a certain set of transitions against that of a distinct set of transitions, as follows;

Definition: Let $N = (P, T, F)$ be an occurrence net and let $\underline{sl}(N)$ be the set of slices of $\langle X_N, \prec \rangle$. Let $N' = (P', T', F', M_0)$ be a safe net and let $T_1, T_2 \subseteq T'$. Let $p : X_N \rightarrow X_{N'}$ be a finite process of N'. Define

$$\tau(p, T_1, T_2) = \max_{S_1, S_2 \in \underline{sl}(N)} \{m(p^{-1}(T_1), S_1, S_2) \; - \; m(p^{-1}(T_2), S_1, S_2)\}$$

to be the *variance in process p* of the occurrence of T_1-type events against the occurrence of T_2-type events. □

Finally, the synchronic distance between two events (sets of transitions) of a safe net may be defined, where the $\underline{sup}$ function returns the greatest of its arguments (which may possibly be ∞).

Definition: [14] Let $N = (P, T, F, M_0)$ be a safe net and $T_1, T_2 \subseteq T$. Call

$$\sigma(T_1, T_2) = \underline{sup}\{\tau(p, T_1, T_2) \mid p \in \pi(N)\}$$

the *synchronic distance* between T_1 and T_2. □

Example: In the net of figure 3.1, the transitions t_1 and t_2 may occur completely concurrently, and by constructing a place p between them with ${}^\bullet p = \{t_1\}$ and $p^\bullet = \{t_2\}$, their synchronic distance may be calculated as $\sigma(t_1, t_2) = 2$. If the net is altered as illustrated in figure 3.7, where a *regulator* section has been added to the net so as to restrict t_1 and t_2 to occurring alternately, the synchronic distance is reduced to $\sigma(t_1, t_2) = 1$. □

Properties of Synchronic Distance

The synchronic distance function as defined above possesses several agreeable technical properties, a few of which are presented in this section. These properties justify, to a degree, the precise choice of definition for synchronic distance.

Theorem 8 *Let $N = (P,T,F,M_0)$ be a safe net with $T_1, T_2, T_3 \subseteq T$. Then*

1. $\sigma(T_1,T_2) = 0 \iff T_1 = T_2$.
2. $\sigma(T_1,T_2) = \sigma(T_2,T_1)$.
3. $\sigma(T_1,T_2) \leq \sigma(T_1,T_3) + \sigma(T_3,T_2)$.

i.e. σ is a metric on the elements of $P(T)$.

Proof: [47] Parts 1 and 2 follow immediately from the definition of synchronic distance. To prove part 3, let $p : X_N \to X_{N'}$ be a process of the safe net $N' = (P',T',F',M_0)$, with $N = (P,T,F)$ an occurrence net. Let S_1 and S_2 be slices of $\langle X_N, \prec \rangle$ such that

$$\tau(p,T_1,T_2) = m(p^{-1}(T_1),S_1,S_2) - m(p^{-1}(T_2),S_1,S_2).$$

Then, defining $[T_i] = m(p^{-1}(T_i),S_1,S_2)$ for $i = 1,2,3$, it is the case that

$$\begin{aligned} \tau(p,T_1,T_2) &= [T_1] - [T_2] \\ &= [T_1] - [T_3] + [T_3] - [T_2] \\ &\leq \tau(p,T_1,T_3) + \tau(p,T_3,T_2). \end{aligned}$$

Thus, by the definition of $\underline{sup}$,

$$\begin{aligned} \sigma(T_1,T_2) &= \underline{sup}\{\tau(p,T_1,T_2) \mid p \in \pi(N')\} \\ &\leq \underline{sup}\{\tau(p,T_1,T_3) + \tau(p,T_3,T_2) \mid p \in \pi(N')\} \\ &\leq \underline{sup}\{\tau(p,T_1,T_3) \mid p \in \pi(N')\} \ + \ \underline{sup}\{\tau(p,T_3,T_2) \mid p \in \pi(N')\}. \end{aligned}$$

□

Theorem 9 *The synchronic distance function $\sigma : P(T) \times P(T) \to \mathbb{N}$ satisfies*

1. $\sigma(T_1 \cup T_2, T_3 \cup T_4) \leq \sigma(T_1,T_3) + \sigma(T_2,T_4) + \sigma(T_1 \cap T_2, T_3 \cap T_4)$.
2. $\sigma(T_1,T_2) = \sigma(T_1 \setminus T_2, T_2 \setminus T_1)$.

where $T_1,T_2,T_3,T_4 \subseteq T$ are events of an occurrence net $N = (P,T,F)$.

Proof: [47] Let $p : N \to N' \in \pi(N')$ be a process of the safe net $N' = (P',T',F',M_0)$, with $N = (P,T,F)$ an occurrence net. For $X_T \subseteq T'$ let $[X_T] = m(p^{-1}(X_T),S_1,S_2)$.

1. Let S_1 and S_2 be slices of $\langle X_N, \prec \rangle$ such that

$$\tau(p,T_1 \cup T_2, T_3 \cup T_4) = m(p^{-1}(T_1 \cup T_2),S_1,S_2) - m(p^{-1}(T_3 \cup T_4),S_1,S_2).$$

Obviously for all $X_T, Y_T \subseteq T'$:

$$\begin{aligned} [X_T \cup Y_T] &= [X_T] + [Y_T \setminus X_T], \\ [X_T \setminus Y_T] &= [X_T] - [X_T \cup Y_T] \text{ and} \\ [X_T] - [Y_T] &\leq \tau(p,X_T,Y_T) \leq \sigma(X_T,Y_T). \end{aligned}$$

Therefore

$$\begin{aligned} \tau(p,T_1 \cup T_2, T_3 \cup T_4) &= [T_1 \cup T_2] - [T_3 \cup T_4] \\ &= [T_1] + [T_2 \setminus T_1] - [T_3] - [T_4 \setminus T_3] \\ &= [T_1] + [T_2] - [T_2 \cap T_1] - [T_3] - [T_4] + [T_4 \cap T_3] \\ &\leq \tau(p,T_1,T_3) + \tau(p,T_2,T_4) + \tau(p,T_1 \cap T_2, T_3 \cap T_4), \end{aligned}$$

and the result follows by the definition of $\underline{sup}$.

2. Let $S_1, S_2 \in \underline{sl}(N)$. For $X_T \subseteq T'$ let $[X_T] = m(p^{-1}(X_T), S_1, S_2)$. Then

$$\begin{aligned}[T_1] - [T_2] &= [(T_1 \setminus T_2) \cup (T_1 \cap T_2)] - [(T_2 \setminus T_1) \cup (T_1 \cap T_2)] \\ &= [T_1 \setminus T_2] + [T_1 \cap T_2] - [T_2 \setminus T_1] - [T_1 \cap T_2] \\ &= [T_1 \setminus T_2] - [T_2 \setminus T_1]\,.\end{aligned}$$

Hence $\tau(p, T_1, T_2) = \tau(p, T_1 \setminus T_2, T_2 \setminus T_1)$ and the result follows.

□

Using σ, the places of the P-completion of a safe net $N = (P, T, F, M_0)$ may be collected into equivalence classes where, for $p_1, p_2 \in P$, $p_1 \sim p_2$ iff $\sigma({}^\bullet p_1, p_1{}^\bullet) = \sigma({}^\bullet p_2, p_2{}^\bullet)$. P together with this relation is called the *synchronic structure* of N.

Weighted Synchronic Distance

The synchronic distance measure of mutual (in)-dependence of events is sometimes not as precise as could be desired. Synchronic distance ought to be interpreted as an upper bound for independence of events, and it is often possible to obtain a tighter bound by using *weighted* synchronic distances[14,13,47]. In particular, weighting a synchronic distance may give a finite value where the original definition of synchronic distance gave ∞ for a particular pair of events.

For a safe net $N' = (P', T', F', M_0)$, a *weight-function* is a function $g : T' \to \mathbb{N} \setminus \{0\}$.

Definition: Let N' and g be as above, with $T_1, T_2 \subseteq T'$.

1. Let $p : N \to N'$ be a process of N', for some occurrence net $N = (P, T, F)$. The *g-weighted variance* of the events T_1 and T_2, denoted $\tau_g(p, T_1, T_2)$ is defined to be

$$\max_{S_1, S_2 \in \underline{sl}(N)} \left\{ \sum_{t \in T_1} g(t) \cdot m(p^{-1}(t), S_1, S_2) - \sum_{t \in T_2} g(t) \cdot m(p^{-1}(t), S_1, S_2) \right\}.$$

2. The *g-weighted synchronic distance* of T_1 and T_2 is defined as

$$\sigma_g(T_1, T_2) = \underline{sup}\{\tau_g(p, T_1, T_2) \mid p \in \pi(N')\}.$$

□

Example: Using the unweighted definition of synchronic distance, the synchronic distance of the transitions α and β in figure 3.8 would be infinite, since the difference between the number of occurrences of α and β may grow unlimitedly. However, the two events are actually tightly coupled, as two occurrences of β alternate with single occurrences of α. This is captured by weighted synchronic distance, where event α is weighted by 2 and all other transitions of the net by 1, and the weighted synchronic distance is $\sigma(\alpha, \beta) = 2$. □

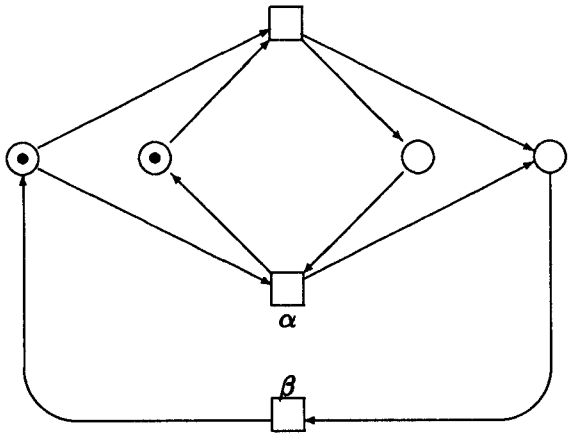

Figure 3.8: Weighted σ < Normal σ

3.3 Free-Choice Nets

As will be seen in chapter 4, it is the behavioural analysis of Petri Nets which is of most practical interest and importance. However, it will also become evident that many analysis problems are very difficult, if not undecidable, for general Petri Nets, and as the static structure of Petri Nets is easier to analyse that their dynamic behaviour, interest has been shown in classes of nets whose very structure permits behavioural inference.

One of the most structurally attractive net classes is that called "Free-Choice Nets", which prohibit confusion by their very structure and permit relatively simple liveness and safeness analysis. Free Choice Nets constitute an aesthetically pleasing combination of the notion of a state machine (conflict, but no concurrency) and the dual notion of a marked graph (concurrency, but no conflict). As a consequence of this structure, Free-Choice Nets exhibit no confusion (adjectivally, they are confusion-free), and it is this factor which essentially simplifies their analysis.

As Free-Choice Nets are a generalisation of two more-restrictive classes (called P- and T-nets), these two subclasses will be presented first.

3.3.1 P- and T-nets

In this section, two measures of the connectedness of an unmarked Petri Net are used, *viz.* *weak* and *strong* connectedness, as defined below.

Definition: An unmarked Petri Net $N = (P, T, pre, post)$ is called *weakly connected* iff for all $x, y \in X_N$ either there is a directed path from x to y, or there is a directed path from y to x, or both. N is called *strongly connected* iff for all $x, y \in X_N$, there is a directed path from x to y. □

Example: The net of figure 3.9(a) is weakly connected, but not strongly connected, while that of figure 3.9(b) is both weakly and strongly connected. The net of figure 2.10 which illustrated concurrent behaviour is neither weakly nor strongly connected. □

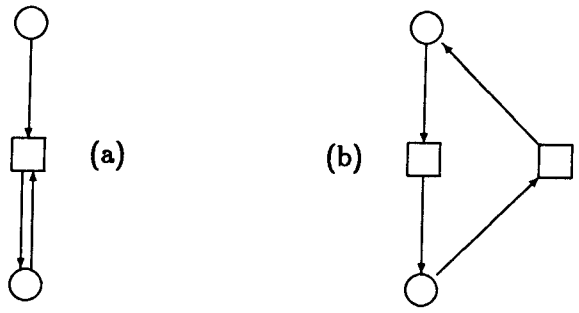

Figure 3.9: Weakly and Strongly Connected Nets

A P-net is a restricted form of (unmarked) Petri Net which admits conflict but no synchronisation (and concurrency only when it is not safe or not weakly connected).

Definition: A *P-Net* is an unmarked Petri Net $N = (P, T, pre, post)$ such that for all $t \in T$, $|{}^\bullet t| \leq 1$ and $|t^\bullet| \leq 1$. □

P-nets possess no synchronisation, simply because there are no (backward) branched transitions; a safe, weakly connected P-net is thus purely sequential.

The dual notion to P-nets are called *T-nets*, and these permit concurrency and synchronisation but no conflict.

Definition: A *T-net* is an unmarked Petri Net $N = (P, T, pre, post)$ such that for all $p \in P$, $|{}^\bullet p| \leq 1$ and $|p^\bullet| \leq 1$. □

There is no conflict present in T-nets simply because there are no (forward) branched places.

Clearly, P- and T-nets are duals of each other, as illustrated in figures 3.10(a) and (b), which show a P-net and its dual T-net, respectively. The analysis of T-nets in chapter 4 makes some use of the *cycles* of a net, as defined below.

Definition: A *cycle* of an unmarked net $N = (P, T, pre, post)$ is a sequence $x_0, x_1, \ldots x_m$ with $x_i \in X_N$ for all $0 \leq i \leq m$ and with $pre(x_{i+1}, x_i) \geq 1$ if $x_i \in T$ or $post(x_i, x_{i+1}) \geq 1$ if $x_i \in P$ for all $0 \leq i < m$ and $x_0 = x_m$. A cycle is called *simple* if no element except $x_0 = x_m$ appears twice in it, i.e. $\forall k, j,\ \ 0 \leq k < j \leq m.\ \ (x_k \neq x_j)$. □

A net is said to be *covered by simple cycles* iff every $x \in X_N$ lies on a simple cycle.

3.3.2 Definition of Free-Choice Nets

Free-Choice nets may now be defined as a generalisation of both P- and T-nets, and they allow both synchronisation (but "only in the T-net way") and conflict (but "only in the P-net way"). Essentially, if two places share a common output transition then they may not have any more output transitions, and similarly for input transitions.

Definition: An unmarked Petri Net $N = (P, T, pre, post)$ is called a *Free-Choice Net* (abbreviated *FC net*) iff for each pair $(p, t) \in P \times T$ with $post(p, t) \geq 1$, $p^\bullet = \{t\}$ or ${}^\bullet t = \{p\}$. □

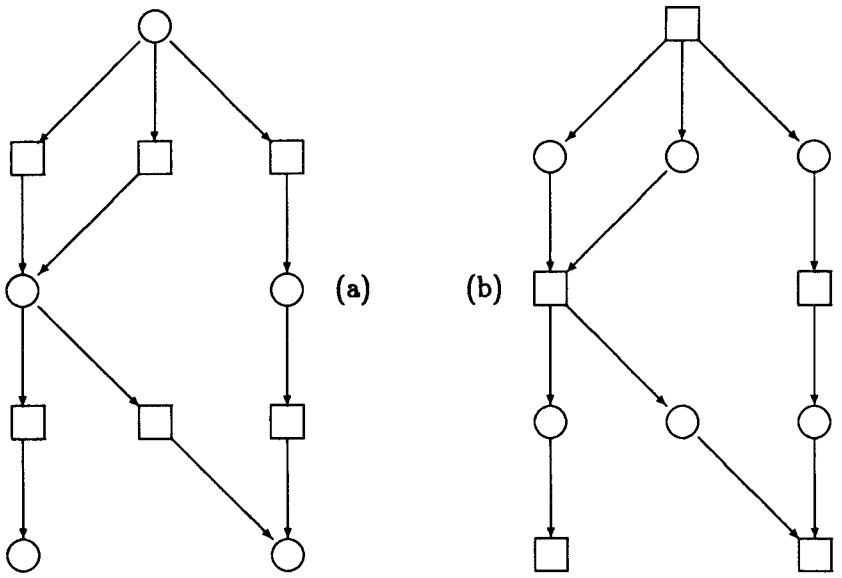

Figure 3.10: Dual P- and T-nets

Theorem 10 *The following properties of an unmarked Petri Net* $N = (P, T, pre, post)$ *are equivalent;*

(1). *N is a Free-Choice Net.*

(2). $p \in P \ \wedge \ |p^\bullet| > 1 \Longrightarrow \forall t \in p^\bullet. \ {}^\bullet t = \{p\}$.

(3). $p \in P \ \wedge \ |p^\bullet| > 1 \Longrightarrow {}^\bullet(p^\bullet) = \{p\}$.

(4). $p_1, p_2 \in P \ \wedge \ p_1{}^\bullet \cap p_2{}^\bullet \neq \emptyset \Longrightarrow \exists t \in T$ *with* $p_1{}^\bullet = p_2{}^\bullet = \{t\}$.

(5). $t_1, t_2 \in T \ \wedge \ {}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset \Longrightarrow \exists p \in P$ *with* ${}^\bullet t_1 = {}^\bullet t_2 = \{p\}$.

Proof: [47,48]

[$1 \Longrightarrow 2$]. If $|p^\bullet| > 1$ then for each $t \in p^\bullet, p^\bullet \neq \{t\}$. Using the definition of FC nets, ${}^\bullet t = \{p\}$.

[$2 \Longrightarrow 1$]. Let $post(p,t) \geq 1$. If $|p^\bullet| = 1$ then immediately $p^\bullet = \{t\}$. If $|p^\bullet| > 1$ then by (2), ${}^\bullet t = \{p\}$ and thus N is Free-Choice.

[3]. Is obviously equivalent to (2).

[$1 \Longrightarrow 4$]. Let $t \in p_1{}^\bullet \cap p_2{}^\bullet$. Since $\{p_1, p_2\} \subseteq {}^\bullet t, {}^\bullet t \neq \{p_1\}$ and ${}^\bullet t \neq \{p_2\}$. By the definition of an FC net, $p_1{}^\bullet = \{t\}$ and $p_2{}^\bullet = \{t\}$.

[$4 \Longrightarrow 1$]. Let $post(p_1,t) \geq 1$. If ${}^\bullet t \neq \{p_1\}$, there exists $p_2 \in P, p_2 \neq p_1$, with $t \in p_2{}^\bullet$. Then $t \in p_1{}^\bullet \cap p_2{}^\bullet \neq \emptyset$ and by (4), $p_1{}^\bullet = \{t\}$, and N is Free Choice.

[5]. Is obviously equivalent to (4).

□

It is clear that every P-net is a Free-Choice net, as is every T-net, so FC nets are truly a generalisation. Clearly, also, the dual of an FC net is an FC net. The notions of P-net, T-net and FC-net each extend immediately to *marked* Petri Nets $N = (P, T, pre, post, M_0)$.

Use will be made of those properties of FC nets which make them particularly amenable to analysis in chapter 4. Two final notions which will be of use there are presented in the following section.

3.3.3 Deadlocks and Traps

Definition: A set $Q \subseteq P$ of places of an unmarked Petri Net $N = (P, T, pre, post)$ is called a *deadlock*[2] iff $^\bullet Q \subseteq Q^\bullet$. The subset Q is called a *trap* iff $Q^\bullet \subseteq {}^\bullet Q$. A deadlock (resp. trap) is called *minimal* if no proper subset of it is also a deadlock (resp. trap).

Let $N = (P, T, pre, post)$ be an unmarked net and let $P' \subseteq P$. A trap Q is said to be the *maximal* trap contained in P' iff $Q \subseteq P'$ and for every other trap Q' contained in P', $Q' \subseteq Q$. □

Deadlocks and traps are both special sets of places. If a deadlock is empty under some marking then it will remain empty under each successor marking; if a trap is marked under some marking then it will remain marked under each successor marking, as shown in the following.

Theorem 11 *Let $N = (P, T, pre, post)$ be a safe, unmarked Petri Net with M a marking of N and $Q \subseteq P$.*

(1). *If Q is a deadlock which is unmarked under M then Q is unmarked under each reachable marking M' with $M \rightsquigarrow^* M'$.*

(2). *If Q is a trap which is marked under M then Q is marked under each reachable marking M' with $M \rightsquigarrow^* M'$.*

(3). *The union of deadlocks is a deadlock.*

(4). *The union of traps is a trap.*

(5). *Q contains a maximal deadlock and a maximal trap.*

Proof: [47]

1. Let Q be unmarked under M and let $M \xrightarrow{t} M'$. Assume that Q is marked under M'. Then $t \in {}^\bullet Q$. If Q is a deadlock then $t \in Q^\bullet$, but this is not possible, since t has concession at M and N is safe.
2. Let Q be marked under M and let $M \xrightarrow{t} M'$. Assume that Q is unmarked under M'. Then $t \in Q^\bullet$, but if Q is a trap then $t \in {}^\bullet Q$ and so Q is marked under M'.
3. $^\bullet P_1 \subseteq P_1^\bullet \ \wedge \ {}^\bullet P_2 \subseteq P_2^\bullet \Longrightarrow {}^\bullet(P_1 \cup P_2) = {}^\bullet P_1 \cup {}^\bullet P_2 \subseteq P_1^\bullet \cup P_2^\bullet = (P_1 \cup P_2)^\bullet$.
4. $P_1^\bullet \subseteq {}^\bullet P_1 \ \wedge \ P_2^\bullet \subseteq {}^\bullet P_2 \Longrightarrow (P_1 \cup P_2)^\bullet = P_1^\bullet \cup P_2^\bullet \subseteq {}^\bullet P_1 \cup {}^\bullet P_2 = {}^\bullet(P_1 \cup P_2)$.
5. Follows from (3) and (4) since ϕ is both a deadlock and a trap.

□

Deadlocks and Traps will be put to considerable use in chapter 4.

[2]This is established but not very agreeable terminology.

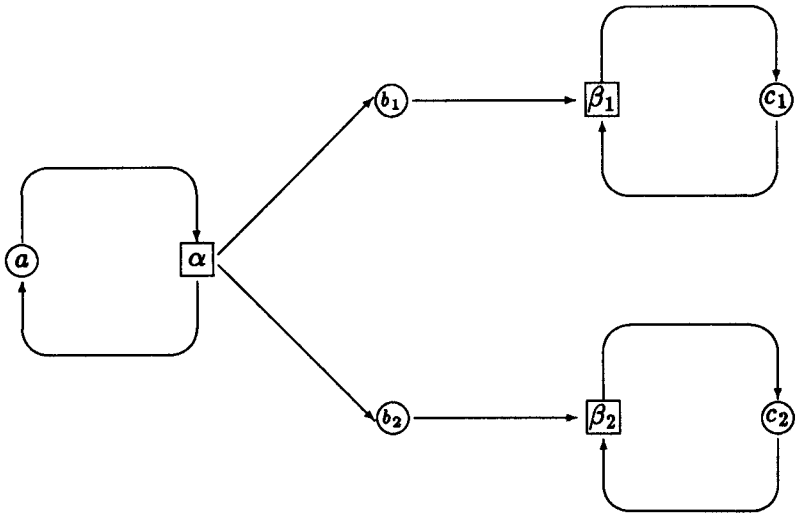

Figure 3.11: A Net with two different types of Consumer

3.4 Coloured Petri Nets

Sometimes, when modeling concurrent systems using Petri Nets, it is inconvenient to use the plain Petri Nets of chapter 2, and several Petri Net extensions have been developed in order to facilitate more concise representation of certain classes of concurrent system.

One such extension, called *Coloured Petri Nets* [28], essentially attaches a *colour* (or "type") to the tokens of a net, thus avoiding the duplication of equivalent subnets of the net structure. This allows simpler, more concise representation of a net, and also reduces the amount of work which must be performed when analysing many net models.

By way of demonstrating the value of such an extension, consider the producer-consumer net introduced in chapter 2. If a system consists of one producer (which produces two tokens per cycle) and *two* consumers (each of which consume one token per cycle), the net of figure 2.6 may be extended to model this situation simply by placing *two* tokens in the place c. Each of these two tokens represents the "state" of one of the consumer processes. This model may represent, for example, a computer system with two printers, each of which can cope with approximately half of the printed output of the system. Although this extension is quite adequate to model the system as described above, if it is further specified that some documents must be printed on one (designated) printer and the remainder of the documents on the second printer (distinguishing, for example, between a letter-quality printer and a normal line-printer), then the model is forced to split the buffer b into two parts, b_1 and b_2, as shown in figure 3.11. Now although this models the behaviour of the system correctly it does not reflect accurately the fact that items of differing types are actually residing in the same buffer (e.g. printer queue) in the system being modeled, nor does it reflect the fact that the two printer processes are essentially the same. Another problem with this system arises when the number of consumers (e.g. printers) is increased — the complexity of the net increases significantly, and analysis of the net must be re-performed *each time* the number of consumers is altered.

Both of these inadequacies may be rectified by use of a Coloured Petri Net, as illustrated in figure 3.12. Note that this net has exactly the same structure as *one* producer-consumer model; the only difference is the presence of annotations on the places, transi-

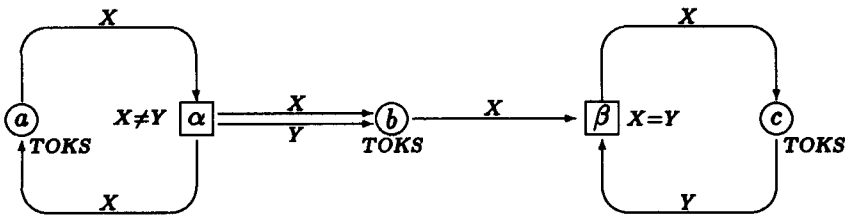

Figure 3.12: A Coloured Petri Net

tions and arcs of the net. In this case, each of the places is inscribed with a set of possible token *colours* (eg. The inscription of place b with the name of the set $TOKS = \{LQ, LP\}$ means that tokens of colours LQ and LP may reside in place b, where LQ denotes a document which is to be printed on the letter-quality printer and LP denotes a document which is to be line-printed). Each arc has an attached *expression* consisting of a single variable name, where variables may take token colours as values. When a transition occurs, all its X-variables must take identical values. Each transition may have an attached *predicate* over the set of variables specifying a condition which must be satisfied (in addition to the normal transition-concession requirements) before the transition may occur (e.g. the transition β requires that the two tokens which are in places b and c before it fires be of the same colour, since $X = Y$ is its inscription).

Whenever a place, transition or arc is uninscribed it is understood that token colour is irrelevant to that component of the net.

The net is initially marked by a token of any colour in place a and two tokens, one LQ and one LP in place c; the buffer place b is initially empty.

More formally, Coloured Petri Nets are defined as follows;

Definition: A Coloured Petri Net is a 6-tuple $N = (P, T, C, pre, post, M_0)$, where

P is a nonempty set of *places*,
T is a nonempty set of *transitions*, with $T \cap P = \phi$,
C is the *colour function* from $P \cup T$ into nonempty sets which attaches to each place a set of possible token colours and to each transition a set of possible occurrence colour tuples.
pre and *post* are functions defined on $P \times T$ such that
$pre(p,t), post(p,t) : \mu C(t) \rightarrow \mu C(p)$.
M_0, the *initial marking*, is a multiset defined over P such that $M_0 \in \mu C(p)$.

which satisfies the restrictions

$$\forall t \in T.\ \exists p \in P. \qquad pre(p,t) \neq \mathbf{0} \ \text{ or } \ post(p,t) \neq \mathbf{0}$$
$$\forall p \in P.\ \exists t \in T. \qquad pre(p,t) \neq \mathbf{0} \ \text{ or } \ post(p,t) \neq \mathbf{0}$$

where $\mathbf{0}$ is the null function which maps everything to ϕ. The colour function is defined so that $C(p)$ is the set of token colours which may be present in place p, which $C(t)$ is

given by

$$C(t) = \{(d_1, d_2, \ldots d_n) \in D_1 \times\ D_2 \times \cdots D_n \mid (\lambda(v_1, v_2, \ldots v_n).PRED)(d_1, d_2, \ldots d_n)\}$$

where $D_1, D_2, \ldots D_n$ are the types (colour-sets) of the expressions inscribed on the incident arcs to transition t, $v_1, v_2, \ldots v_n$ are free variables of types $D_1, D_2, \ldots D_n$, respectively, and $PRED$ is the predicate attached to the transition t.

The functions *pre* and *post* assign to each pair $(p,t) \in P \times T$ the function

$$\lambda(v_1, v_2, \ldots v_n).EXP\ ,$$

linearly extended to multisets, where $v_1 : D_1, v_2 : D_2, \ldots v_n : D_n$ are free variables of the indicated types D_i, which are the types of the arcs incident on the transition t, and where EXP is the expression inscribed on the arc (p,t). If there is more than one arc between place p and transition t, then the entry of *pre* or *post* corresponding to (p,t) is the multiset sum of the above functions. The null function signifies that there are no arcs between p and t. □

Example: The Coloured Petri Net of figure 3.12 may be represented as the 6-tuple $N = (P, T, C, pre, post, M_0)$ where the set of colours is the set $TOKS = \{LQ, LP\}$, and

$$\begin{aligned}
P &= \{a, b, c\} \\
T &= \{\alpha, \beta\} \\
(M_0)_a &= \{LQ\},\ (M_0)_b = \emptyset,\ \text{ and }\ (M_0)_c = \{LQ, LP\} \\
C(a) &= TOKS = C(b) = C(c) \\
C(\alpha) &= \{(d_1, d_2) \in TOKS^2 \mid (\lambda(X,Y).X \neq Y)(d_1, d_2)\} \\
&= \{(LQ, LP), (LP, LQ)\} \\
C(\beta) &= \{(d) \in TOKS \mid (\lambda(X).X)(d)\} \\
&= \{(LP), (LQ)\}
\end{aligned}$$

and the only non-null entries of *pre* and *post* are

$$\begin{aligned}
pre(a,\alpha) &= \lambda(X,Y).X & \qquad post(a,\alpha) &= \lambda(X,Y).X \\
pre(b,\beta) &= \lambda(X).X & post(b,\alpha) &= \lambda(X,Y).X + Y \\
pre(c,\beta) &= \lambda(X).X & post(c,\beta) &= \lambda(X).X
\end{aligned}$$

□

The transition relation of Petri Nets is extended to Coloured Petri Nets by treating an event as a function from T to $\mu C(T)$. Letting $\bullet A$ and $A\bullet$ denote the expressions

$$\forall p \in P.\ \sum_{t \in T} pre(p,t)(A(t)) \ \text{ and } \ \forall p \in P.\ \sum_{t \in T} post(p,t)(A(t)),$$

(vectors over P), respectively, the transition $A : T \to \mu C(T)$ has concession at a marking M iff $\bullet A \leq M$, treating both as vectors over P.

Definition: Let $N = (P, T, C, pre, post, M_0)$ be a Coloured Petri Net with two markings M and M', and let $A : T \to \mu C(t)$ be an event of N. The *transition relation* for N is defined as

$$M \xrightarrow{A} M' \quad \text{iff} \quad \bullet A \leq M \ \text{ and } \ M' = M - \bullet A + A\bullet$$

□

Example: The Coloured Net of figure 3.12 admits the event X, which maps T to $\mu C(t)$ according to $X : \alpha \longmapsto \{(LQ, LP)\}$ and $X : \beta \longmapsto \{(b)\}$, which may occur at the indicated marking to give a new marking M', where $M'_a = \{LQ\}, M'_b = \{LP, LQ\}$, and $M'_c = \{LP, LQ\}$. □

Coloured Petri Nets have been used for solving many practical modeling problems, and when combined with techniques for determining Net invariants, they constitute an important concurrent system design paradigm.

Chapter 4

Net Analysis

A major motivation for the development of Petri Net theory is that the Petri Net model provides a vehicle for the *analysis* of concurrent systems. Such systems include concurrent computer programs, parallel machine architectures, and models of information flow in business, to name but a few examples. Systems analysis has traditionally asked several questions of a concurrent system, *viz.* will it ever *deadlock?* Will *part* of it deadlock? Are there enough *resources* present? Is the system organisation "optimal" in some sense?, etc.

The classical safeness, liveness and invariance properties of concurrent systems apply immediately to Petri Nets, and there are also several other properties specific to Petri Nets which prove to be of interest. As a brief intuition, safeness[1] properties require that nothing "bad" ever happens during the net's execution (e.g. no deadlock occurs). Liveness properties require that something "good" eventually happens (e.g. fair execution, no livelock, etc). Invariance properties require that certain aspects of the system remain in balance (e.g. the number of resource tokens in a net is constant).

For Petri Nets, safeness and liveness properties may generally be expressed in terms of the question

Can a particular marking M *ever* be reached by a given net N?

This question, formalised as the Reachability Problem (RP) for Petri Nets, subsumes many other Petri Net problems, as will be shown later. Some of the important Petri Net analysis problems which are variously related to the RP are;

1. The Boundedness Problem (BP). This problem asks whether a particular place of a net may ever contain more than k tokens (for some constant k). This k-boundedness problem is sometimes called k-safeness, and clearly 1-safeness (setting k to 1) corresponds to the notion of a *safe* net as introduced in section 3.1. Related to this problem is that of determining whether two distinct subsets of places are *simultaneously unbounded* in any reachable marking of the net.

2. The Equivalence Problem (EP) asks whether two particular nets behave equivalently in the sense of having equivalent sets of reachable markings. A related problem is the Containment Problem (CP), which asks whether a given net's reachability set is contained in another net's reachability set. The Equivalence Problem represents a variation on Net bisimulations as a means of comparing two nets.

[1] These should not be confused with "safe" Petri Nets

3. The Coverability Problem (*KP*) asks whether there exists a marking $M \in \mathcal{R}_N$ such that a given marking M' *covers* M (i.e. $M' \geq M$ as vectors).

4. The Liveness Problem (*LP*) represents a combination of deadlock and livelock questions. There are several definitions of liveness for a Petri Net, essentially corresponding to the definitions of *fairness* of Francez[10], and ranging from "Can this net still execute any transition?" (Is it deadlocked?) to "Does every transition of this net have concession infinitely often in any infinite run of the net?".

Many other analysis questions have been asked about Petri Nets in the literature, but the aforementioned are the most important. The next section examines the *RP* in detail and shows its relationship to the *BP*, *EP*, *CP* and *LP*. Invariants will be discussed in section 4.2, where it will be seen that they provide a more sophisticated, and often more computationally feasible method of Petri Net analysis than do solutions to the *RP*. Finally, in section 4.3, analysis methods will be applied to several of the net models introduced in Chapter 3 and it will be seen that restricting the structure of nets can simplify their analysis significantly.

4.1 The Reachability Problem

Firstly, the *RP* will be formally defined, so that observations may be made regarding the problem and its solution.

Definition: The *Reachability Problem for Petri Nets* is formulated as follows;

> "Given a Petri Net $N = (P, T, pre, post, M_0)$ and a marking M, is M a reachable marking of N?" (or alternatively, "Is $M \in \mathcal{R}_N$?")

□

The Reachability Problem for Petri Nets was originally expressed [30] as a problem over Vector Addition Systems – a linear-algebraic formalism which is equivalent to the dynamic behaviour of Petri Nets (see Appendix C). All solutions to the RP consider only singleton events, as opposed to the multisets of events used in Chapters 2 and 3 of this report. This merely simplifies the (rather complex) solutions and proofs for the Reachability Problem, and the extension to composite events is immediate.

It was shown very early that the RP for VAS was decidable for limited dimension cases [25], but for even this restricted result, the new formalism of Vector Addition Systems with States (VASS) was required. The exact structure of a VASS is detailed in the appendix, and for present purposes it will simply be shown that VAS and VASS are equivalent to a sufficient degree that it is possible to treat their reachability problems as also being equivalent.

Lemma 12 *An n-dimensional VASS can be simulated by an $n+3$-dimensional VAS.*

Proof: [25] The proof follows the construction of a simulating VAS from a given VASS. The last three coordinates of the VAS are used to encode the VASS' state while the first n coordinates are exactly as in the VASS. Assume that the VASS has k states $q_1, \ldots, q_k$ and let $a_i = i$ and $b_i = (k+1)(k+1-i)$ for $i \in \{1, \ldots, k\}$. If the VASS is at v in state q_i then the VAS will be at $(v, a_i, b_i, 0)$. For each i the VAS has two dummy transitions

t_i and t_i' defined so that t_i goes from $(v, a_i, b_i, 0)$ to $(v, 0, a_{k-i+1}, b_{k-i+1})$ and t_i' goes from $(v, 0, a_{k-i+1}, b_{k-i+1})$ to $(v, b_i, 0, a_i)$. It should be noted that t_i and t_i' modify only the last three components of any vector in the VAS. In addition, there is a transition t_i'' for each step $i \longrightarrow (j, w)$ of the VASS, defined by

$$t_i'' = (w, a_j - b_i, b_j, -a_i).$$

Clearly, any path of the VASS can be mimicked by the VAS. It remains to show that the VAS cannot do anything unintended; it will merely be shown that t_i'' can only be applied if the last three components of the vector in question are $b_i, 0$, and a_i, respectively, as the other cases are similar. Observe that for each i and j, $a_i < a_{i+1}$, $b_i > b_{i+1}$, $a_i < b_j$ and $b_i - b_{i+1} = k + 1 > a_j$. Let v_i'' be the vector $(w, a_j - b_i, b_j, -a_i)$ which accomplishes the transition t_i''. Note that the $n+1^{st}$ and last components are negative, and hence t_i'' cannot be applied when the last three coordinates are $(a_i, b_i, 0)$ or $(0, a_{k-i+1}, b_{k-i+1})$ since either the first or third components are 0. Let the last three components be $(b_m, 0, a_m)$. Then if $m < i$, t_i'' cannot be applied since $a_m - a_i < 0$. If $m > i$, then t_i'' cannot be applied since

$$b_m + a_j - b_i \;\leq\; a_j - (k+1) \;<\; 0.$$

□

Since an n-dimensional VASS can trivially simulate an n + 3-dimensional VAS, the reachability problem for a VAS is decidable iff the reachability problem for VASS (see Appendix C) is decidable.

From VASS, Kosaraju designed the Generalised VASS (GVASS) formalism, and in 1982 he published the following decision procedure which satisfactorily[2] solved the GVASS (and hence, VASS, VAS and Petri Net RPs) reachability problem [31].

Theorem 13 (Kosajaru 1982) *The Reachability Problem for GVASS is decidable.*

Proof: The essence of the proof is contained in the following decision procedure for the Reachability problem, where G is a GVASS and θ is a property as described below.

```
procedure Decide (G, RESULT);
begin
   if G satisfies θ then
      RESULT := yes;
      exit;
   elsif size(G) is not trivial then
      compute a finite set GS of reduced GVASS for G;
      for all g ∈ GS do
         Decide(g, RESULT);
      end for all;
   else
      RESULT := no;
      return;
   end if;
end procedure Decide;
```

[2]In 1981, Mayr published a paper [36,37] in which he claimed to have solved the General RP, but his extremely complex proofs have not satisfied several of the major contributors to the field, and it remains an open question whether his solution to the RP is adequate.

The property θ is crucial to the proof, and as it is rather complex, the reader is directed to [31,39] for details. The correctness of the procedure follows from the following four theorems of Kosaraju;

1. "G satisfies θ" is decidable.
2. If G satisfies θ then G has a CR-path.
3. If G does not satisfy θ and size(G) is not trivial then a finite set of reduced GVASS can be effectively computed such that
   (a) $\forall g \in GS.\ \ size(g) < size(G)$, and
   (b) G has a CR-path iff $\exists g \in GS:\ \ g$ has a CR-path.
4. If G does not satisfy θ and size(G) is trivial then G has no CR-path.

where the measure size(G) represents a balance of the number of nodes and arcs in the underlying VASSs of each GVASS G. □

The exact complexity of this decision procedure for the Reachability Problem for Petri Nets remains an open problem, but it has been known for some time that the RP itself requires a lower bound of exponential space [46], and that its time complexity is unbounded by any primitive recursive function (i.e. NP-hard) [29]. It has recently been determined that the Reachability Problem is in fact NP-complete in both space and time[27]. The solution of the RP was a major breakthrough, as it had been demonstrated that many of the other Petri Net problems were related or equivalent to the RP (as is shown in the next section) and results for these problems were thus obtained automatically from Kosaraju's proof.

4.1.1 Relationship of RP to Other Problems

The Boundedness Problem (BP)

Formally, Petri Net boundedness is defined as follows. Let $N = (P, T, pre, post, M_0)$ be a marked Petri Net; a place $p \in P$ is n-bounded (for any $n \in \mathbb{N}$) iff $\forall M \in \mathcal{R}_N : M_p \leq n$. The net N itself is n-bounded (for $n \in \mathbb{N}$) iff $\forall p \in P : p$ is n-bounded. In accordance with the previous definition of safe nets, N is *safe* iff N is 1-bounded.

Obviously, if a place is unbounded in a net N, the reachability set $\mathcal{R}_N$ must also be unbounded in size, and thus the solution to the RP is of no use in solving the BP.

The BP was shown to be decidable by its inventors [30], and the complexity of the problem has since been refined to lie between a lower bound of $O(2^{c\sqrt{n}})$ [27,35] and an upper bound of $O(2^{cn \log n})$ [46] space, where n is the number of bits in the problem instance and c is any constant. Solutions to the Coverability Problem (see below) provide a conceptually manageable (though computationally unattractive) method for solving the BP.

The Equivalence (EP) and Containment (CP) Problems

Using the undecidability of Hilbert's 10th problem, Rabin gave a very early (though unpublished) proof for the undecidability of the CP, and Hack applied this result to the EP by proving that the CP is recursively reducable to the EP [19].

Thus both the CP and the EP are, in general, undecidable; it was discovered, however, that for certain classes of nets (namely those whose reachability sets are effectively computable semilinear sets) the CP and the EP are decidable.

Classes of nets with semilinear reachability sets include those representable as VAS of up to 5 dimensions, those which are conflict-free, and those which are persistent. This result led to considerable work being performed with such restricted net models, although the eventual solution to the RP makes their use less necessary.

The Coverability Problem (KP)

One of the early *approximate* solutions to the RP was provided in the form of the KP, in the sense that it can be determined whether a marking M may be in $\mathcal{R}_N$ simply by building a structure called a *Coverability Tree* for N and examining this to see if there is an M' in the tree with $M' \geq M$. If no such M' exists, then $M \notin \mathcal{R}_N$ definitely; if $\exists M' : M' = M$ then $M \in \mathcal{R}_N$ definitely, and if $\exists M' > M$ then M *may be* in $\mathcal{R}_N$ (and herein lies the "approximation").

A Coverability Tree is an approximation to the intuitive notion of a Reachability Tree (a tree of all reachable markings, connected by arcs labelled with the transitions going from marking to marking) which is "pruned" to abbreviate any infinite branches of the tree. This pruning causes the loss of some information, but this loss is countered by the fact that a conceptually simple approximate decision procedure is obtained.

The following procedure for constructing a coverability tree is due to Hack [18].

A Coverability Tree is a rooted, labelled tree, where the node labels are $|P|$-dimensional ω-multisets for a net $N = (P, T, pre, post, M_0)$. The arcs of the tree are labelled by transition names (members of T). In addition to the arcs of the tree, there are two types of *backpointers*, which can point from a node α to an antecedent of that node. These pointers are not considered to be arcs of the tree (in the interests of it remaining a well-formed tree), but are used for book-keeping purposes. If β is an antecedent of α, this is denoted $\beta \prec \alpha$ (consistent with the poset notation used in section 3.2 and appendix B). The label of a node α is denoted L_α.

The *root node* is an antecedent to every other node in the tree, and its label will be the initial marking M_0. A *leaf node* is not antecedent to any node. The node labels reflect the corresponding marking changes, but as soon as a node α is reached whose label L_α covers the label of some antecedent β (i.e. $L_\alpha \geq L_\beta$), there is a possibility of unboundedness, and ω is introduced to those coordinates where arbitrarily many tokens can be generated if the sequence of transitions expressed by the arc labels along the path from β to α is repeated sufficiently often. To express this more conveniently, a ω-*backpointer*, labelled ω_i if ω is introduced in the i^{th} coordinate, is included from the node α to the corresponding antecedent β.

If a node α is reached whose label is *equal* to that of an antecedent β then α is made a leaf-node and a *loop-backpointer* is introduced, labelled λ (for the empty string), from α to β. This procedure is expressed more formally in the following definition.

Definition: Given a Petri Net $N = (P, T, pre, post, M_0)$, its *Coverability Tree* T_N is defined recursively by the following procedure. The label of the root node ρ is the original marking ($L_\rho = M_0$). The Coverability Tree is generated from ρ by a call to the procedure `CoverTree(`ρ`);`.

```
begin
   if no Transition has concession at Marking L_α then
      α is a leaf-node called a dead-end;
   elseif ∃ an Antecedent β of α with L_β = L_α then
      α is a leaf-node called a loop-end;
      Add a loop-backpointer from α to β;
      { This is denoted α ~λ~> β }
   else
      for each Transition t which has concession at L_α do
         Create a new node β;
         Compute the marking L' = L_α − •t + t•;
         {Which results from t's occurrence at the marking L_α }
         Compute the set A_β = {γ | γ ≺ β and L_γ ≤ L'};
         {Of Antecedents of β with smaller markings than L_β };
         if A_β = ∅ then
            L_β = L' ; { is the label for node β }
         else
            for each coordinate i for which (L'(i) ≠ ∞) and
               (L'(i) > L_γ(i)) for some γ ∈ A_β do
               Introduce an ∞-backpointer from β to γ;
               {This is denoted β ~∞i~> γ }
            end for;
            {L_β is then determined componentwise as follows}
            for all i with 1 ≤ i ≤ |P| do
               L_β(i) := if (∃γ ∈ A_β : β ~∞i~> γ) then ∞ else L'(i) end if;
            end for all;
         end if;
         Call CoverTree(β);
      end for;
   end if;
end procedure CoverTree;
```

□

Clearly, the algorithm will terminate iff the tree is finite, and Hack showed (by the following theorem) that this is guaranteed by the structure of the algorithm.

Theorem 14 *Every Coverability Tree is finite, and can be effectively constructed.*

Proof: See [18]. □

Example: A Coverability Tree for the "Producers-and-Consumers" Petri Net of figure 2.6 which was constructed using the above algorithm is illustrated in figure 4.1. All backpointers and loop-end leaf nodes have been omitted for clarity. From the Coverability tree, it is easy to see that the marking $(0,1,1)$ can *never* be reached by the net; that the marking $(1,0,1)$ is *definitely* reachable by the net, and that the marking $(1,324,1)$ *may possibly* be reachable.It should be noted that the Coverability Tree for a given net may not necessarily be unique. □

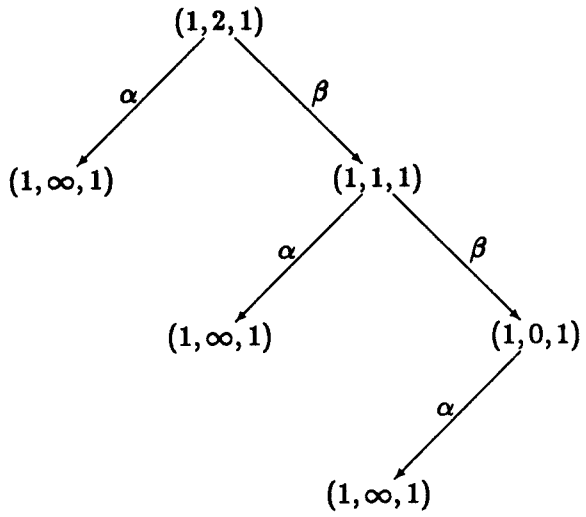

Figure 4.1: Coverability Tree for the Producer/Consumer Net

The following two results relating to the BP are obtained from the Coverability Tree construction.

Theorem 15 *A place $p \in P$ of a Petri Net $N = (P, T, pre, post, M_0)$ is unbounded iff a coverability tree for N contains a node α with label L_α in which the corresponding coordinate is $L_\alpha(p) = \infty$.*

Proof: See [18] □

Theorem 16 *It is decidable whether a set of places is simultaneously unbounded.*

Proof: [18] The Coverability Tree (which is finite and effectively constructible) can be examined to see whether it contains a label in which the coordinates corresponding to these places are all ∞. □

It should be noted, however, that the problem of generating of a coverability tree is of the same order of complexity as the BP (i.e. unbounded by any primitive recursive function), and it is thus not computationally feasible, in general, to make much use of coverability trees for practical net analysis.

The Liveness Problem (LP)

Finally, the Liveness and Reachability problems were proved to be recursively equivalent by Hack in 1974 [20]. This important result placed even more emphasis on the need to know whether the Reachability Problem was decidable, and helped to spur on those who were searching for solutions to the RP. The definition of liveness used in most Petri Net work (including section 4.3 of this report) is given below;

Definition: Let $N = (P, T, pre, post, M_0)$ be a Petri Net;

1. $t \in T$ is called *live* iff $\forall M \in \mathcal{R}_N(M_0).\ \exists M' \in \mathcal{R}_N(M) s.t.\ t$ has concession at M'.
2. N is *live* iff $\forall t \in T.\ t$ is live.

□

Liveness as expressed here is thus equivalent to the *weak liveness* of Francez[10] where essentially all that is required is absence of global deadlock. More complex (and more demanding) liveness requirements may be phrased for Petri Nets, but the above (weak) liveness definition is the most usual.

4.2 Net Invariants

This section presents the theory and intuition related to sets of Petri Net places which maintain a constant token count during any run of the net; the so-called *place-invariants* of a net.

Knowledge of such sets of places helps in analysing safeness, liveness and other properties of Petri Nets, and corresponds essentially to the use of invariant assertions in the program proof techniques of [7] and [17].

Definition: Let $N = (P, T, pre, post, M_0)$ be a Petri Net and let W be a matrix called the *incidence matrix* of N, such that

$$W(p,t) = post(p,t) - pre(p,t)$$

for all $p \in P, t \in T$. Let I be a vector over P (i.e. $I \in \nu P$). Then

1. I is called a *place-invariant* of N iff $W \cdot I = \mathbf{0}$,
2. $P_I \subseteq P$ is called the *support* of I iff $P_I = \{p \in P \mid I_p \neq 0\}$, and
3. A place invariant $I > \mathbf{0}$ of N is called *minimal* iff $\nexists\ I' > \mathbf{0}$ of N with $I' < I$.

□

The set of invariants of a Petri Net N is denoted $InvN$. Since invariants are characterised by solutions of a system of linear equations, they can be computed by well-known methods of linear algebra. Also from linear algebra, it is clear that the following Lemma is valid.

Lemma 17 *Let I_1 and I_2 be place-invariants of a net N, and let $z \in \mathbb{Z}$ be any integer. Then $I_1 + I_2$ and $z \cdot I_1$ are also place-invariants of N.*

Net invariants thus form a $\mathbb{Z}$-Module (An Abelian group with composition $+$ and identity $\mathbf{0}$, together with a scalar product operation $\cdot : \mathbb{Z} \times \nu P \rightarrow \nu P$).

It should be noted that as a morphism between $\mathbb{Z}$-modules M and N is a function $\alpha : M \rightarrow N$ which is *linear* in the sense that $\alpha(zv) = z(\alpha v)$ and $\alpha(u+v) = \alpha u + \alpha v$ for all $u, v \in M$ and $z \in \mathbb{Z}$, it follows that $\mathbb{Z}$-modules correspond to Abelian groups and their morphisms correspond to homomorphisms on Abelian groups [55].

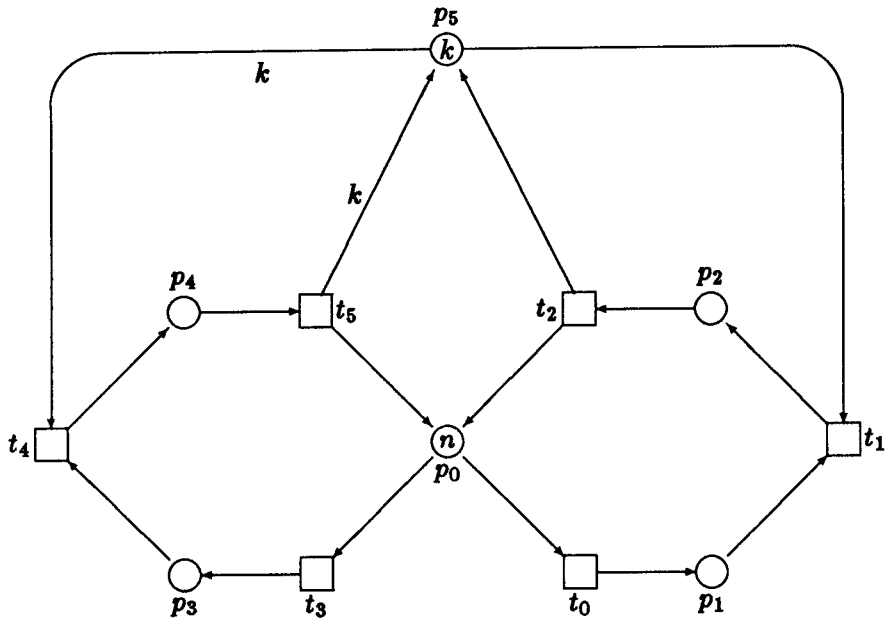

Figure 4.2: A System of Readers and Writers

Such *invariant* vectors I are so-called because a solution I to the equation $W \cdot I = \mathbf{0}$ will preserve markings of the net N in the sense of the following theorem. By way of notation, the vector $\underline{t} : P \to \mathbb{Z}$ is defined componentwise as follows, for all transitions $t \in T$;

$$\underline{t}_p = \begin{cases} post(p,t) & \text{iff} \quad p \in t^\bullet \setminus {}^\bullet t \\ -pre(p,t) & \text{iff} \quad p \in {}^\bullet t \setminus t^\bullet \\ post(p,t) - pre(p,t) & \text{iff} \quad p \in {}^\bullet t \cap t^\bullet \\ 0 & \text{otherwise} \end{cases}$$

and clearly it is the case that $M \xrightarrow{t} M'$ is an event of the net, with M and M' reachable markings, if and only if $M' = M + \underline{t}$.

Theorem 18 *Let $N = (P, T, pre, post, M_0)$ be a petri net. For each place-invariant I of N and each reachable marking $M \in \mathcal{R}_N(M_0)$ it is the case that $M \cdot I = M_0 \cdot I$.*

Proof: [47] Let $M_1, M_2 \in \mathcal{R}_N(M_0)$ and let $t \in T$ such that $M_1 \xrightarrow{t} M_2$ is an event of N. Then, in particular, $M_2 = M_1 + \underline{t}$ and $\underline{t} \cdot I = \mathbf{0}$ (since I is a place-invariant). Therefore,

$$M_2 \cdot I = (M_1 + \underline{t}) \cdot I = M_1 \cdot I + \underline{t} \cdot I = M_1 \cdot I,$$

and the result follows by induction from the initial marking M_0. □

Example: The net of figure 4.2 illustrates a system of n processes (indicated by n tokens on place p_0, the "inactive processes" place) which can either perform a read or a write operation (i.e. can execute transitions t_1 or t_4, respectively) on a shared data base.

The system is restricted so that a maximum of $k \leq n$ processes may read the data base at any one time, and so that writing may only take place when there are *no* processes reading. There are initially k tokens on the synchronisation place p_5.

The other places correspond to the states "ready to read" (p_1), "reading" (p_2), "ready to write" (p_3) and "writing" (p_4).

Given the above specification for the net, the incidence matrix W for N is as follows,

	p_0	p_1	p_2	p_3	p_4	p_5
t_0	-1	1	0	0	0	0
t_1	0	-1	1	0	0	-1
t_2	1	0	-1	0	0	1
t_3	-1	0	0	1	0	0
t_4	0	0	0	-1	1	$-k$
t_5	1	0	0	0	-1	k

and it is a simple matter to solve the equation $W \cdot I = \mathbf{0}$ (using normal linear-algebraic techniques, such as Gaussian elimination) to determine that all place-invariants of the net are of the form

$$I = \alpha \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} -k \\ -k \\ -k \\ 1-k \\ 0 \\ 1 \end{bmatrix},$$

where α and β are any integers. □

It has been shown [55] that net morphisms preserve invariants. Invariants are closely related to deadlocks and traps, as shown in the following result; there c_Q denotes the *characteristic vector* of $Q \subseteq P$, meaning that $c_p \in \{0,1\}$ for all $p \in P$ and $c_p = 1$ iff $p \in Q$; otherwise $c_p = 0$.

Lemma 19 *Let $N = (P, T, pre, post, M_0)$ be a Petri Net with a positive place-invariant $I > \mathbf{0}$ and let $Q = \{p \in P \mid I_p > 0\}$. Then $Q^\bullet = {}^\bullet Q$.*

Proof: [47] Assume that there exists $t \in Q^\bullet \setminus {}^\bullet Q$. Then

$$\exists p \in P : t_p < 0 \text{ and } \forall p \in P : \neg(t_p > 0).$$

Then clearly $t \cdot c_Q < 0$ and since I is positive, $c_Q \leq I$ and therefore $t \cdot I < 0$. So I is, under this assumption, not a place-invariant. Similarly for $t \in {}^\bullet S \setminus S^\bullet$, it is the case that $t \cdot I > 0$. □

Corollary 20 *Every positive place-invariant is both a deadlock and a trap.*

Recent work on the generation of all invariants for a Petri Net has concentrated on altering classical computation techniques for linear Diaphantine equations $W \cdot I = \mathbf{0}$ to produce techniques which are tailored specifically towards invariants, and which are more computationally attractive [32].

Place-invariants for Coloured Petri Nets are rather more complicated structures, but the same concepts are preserved, and the interested reader is directed to [28].

4.2.1 Nets covered by place-invariants

Unbounded places of a net may not feature in any place-invariant, as indicated in the remainder of this section.

Definition: A net $N = (P, T, pre, post, M_0)$ is said to be *covered by* place invariants iff for each place $p \in P$ of N there exists a positive place-invariant I of N with $I_p > 0$. □

Thus, to feature in a place-invariant I, a place $p \in P$ must correspond to a non-zero component of I. The following result concludes that any place which does not feature in any such I must be unbounded in the net N.

Theorem 21 *Let $N = (P, T, pre, post, M_0)$ be a net with a finite initial marking M_0. If N is covered by place invariants then N is k-bounded for some $k \in \mathbb{N}$.*

Proof: [47] Let $p \in P$ and let I be a positive place invariant of N with $I_p > 0$; let $M \in \mathcal{R}_N(M_0)$ be a reachable marking. Since

$$M_p \cdot I_p \leq \sum_{p' \in P} M_{p'} \cdot I_{p'} = M \cdot I = M_0 \cdot I,$$

it is the case that

$$M_p \leq \frac{M_0 \cdot I}{I_p}.$$

□

4.2.2 Using Invariants to Prove Assertions about Nets

Not surprisingly, invariants are useful for describing invariant behaviour of nets. In addition, they comprise a powerful tool for proving safeness and liveness properties about Petri Nets. For example, two particular invariants of the net in figure 4.2 are (setting $\alpha = 1$ and $\beta = 0$)

$$I_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

and (setting $\alpha = k$ and $\beta = 1$)

$$I_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ k \\ 1 \end{bmatrix},$$

and these two invariants define safeness properties as described below.

Firstly, I_1 states that in all markings of the net, the number of processes remains constant (n), since

$$\sum_{i=0}^{4} M_{p_i} = \sum_{i=0}^{4} (M_0)_{p_i} = n.$$

The second invariant, I_2, gives the formula

$$M_{p_2} + k \cdot M_{p_4} + M_{p_5} = (M_0)_{p_2} + k \cdot (M_0)_{p_4} + (M_0)_{p_5} = k$$

for all reachable markings M. This facilitates the deduction of the following facts;

1. Place p_4 contains at most *one* token under any M (i.e. there is at most *one* process writing);
2. When p_4 carries a token, p_2 and p_5 are empty (i.e. when a process is writing, no other process may read the buffer);
3. p_2 carries at most k tokens (i.e. there are at most k processes reading concurrently).

In addition, liveness properties may also be deduced from invariants, as shown in the following result.

Proposition 22 *With the initial marking as indicated, the net of figure 4.2 is live.*

Proof: [47] Liveness is demonstrated by showing that at every marking will enable at least one transition. In a marking M where $M_{p_0} + M_{p_2} + M_{p_4} > 0$, the net structure reveals that at least one of the transitions t_0, t_2, t_3 or t_5 is enabled. If $M_{p_0} + M_{p_2} + M_{p_4} = 0$ then I_1 gives the result that $M_{p_1} + M_{p_3} = n$ and I_2 gives that $M_{p_5} = k$; then t_1 or t_4 is enabled. Now, if p_0 is empty for some reachable marking $M \in \mathcal{R}(M_0)$, it may be marked by some succession of firings. This implies the liveness of t_0 and t_3. The liveness of the other transitions follows immediately. □

Thus, as demonstrated, invariants provide a powerful basis for reasoning about Petri Nets, especially when combined with a form of logic capable of expressing time-invariant assertions such as temporal logic.

4.3 Analysis of Restricted Classes of Nets

Due to the complexity of the Petri Net Reachability Problem, and also because the solution to this problem was only discovered recently, a large (and growing) body of theory has been developed in an effort to determine salient features of a net's dynamic behaviour from its static structure. Called *Structure Theory* [3], this school soon determined that deep results were scarce for general Petri Nets, but for some subclasses of nets, significant results were easier to obtain.

This section first presents some of the (rare) general results relating Petri Net structure and such dynamic properties as liveness and boundedness/safeness. Then properties relating specifically to P- and T-nets will be examined, followed, finally, by several major results for Free Choice Nets.

4.3.1 General Results

There exist very few non-trivial necessary conditions for liveness and safeness in General Petri nets. Sufficient conditions are even fewer and farther between. The first two propositions below represent necessary conditions for the liveness of a net and for the safeness of a live net, respectively.

Proposition 23 *If a Petri Net* $N = (P, T, pre, post, M_0)$ *is live then* $\forall p \in P : {}^\bullet p \neq \mathbf{0}$.

Proof: [4] Suppose that ${}^\bullet p = \mathbf{0}$ for some $p \in P$. By the definition of a Petri Net, $p^\bullet \neq \mathbf{0}$. Because of the liveness of N, all transitions in $p^\bullet$ may eventually fire, and thus there is a reachable marking $M \in \mathcal{R}_N(M_0)$ at which $M_p = 0$. Because ${}^\bullet p = \mathbf{0}$, it is the case that $M'_p = 0$ for all $M' \in \mathcal{R}_N(M)$, so none of the transitions in $p^\bullet$ can ever be enabled again, contradicting liveness. □

Proposition 24 *If a live Petri Net* $N = (P, T, pre, post, M_0)$ *is safe then* $\forall p \in P : p^\bullet \neq \mathbf{0}$.

Proof: [4] Suppose N is live and safe, and $p^\bullet = \mathbf{0}$ for some $p \in P$. Again, by the definition of a Petri Net, it must be the case that ${}^\bullet p \neq \mathbf{0}$. By the liveness of N, all of the transitions in ${}^\bullet p$ may eventually fire, and when one does, p will hold a token; thus, there is a reachable marking $M \in \mathcal{R}_N(M_0)$ such that $M_p = 1$ (possibly $M = M_0$). Because $p^\bullet = \mathbf{0}$ and because of the safeness of N, it is the case that $M'_p = 1$ for all $M' \in \mathcal{R}_n(M)$, so none of the transitions in ${}^\bullet s$ can ever occur again, contradicting liveness. □

Conversely to the above, the following two propositions provide necessary conditions for safeness, and for the liveness of a safe net, respectively.

Proposition 25 *If a Petri Net* $N = (P, T, pre, post, M_0)$ *is safe then* $\forall t \in T : {}^\bullet t \neq \mathbf{0}$.

Proof: [4] Suppose that N is safe and that $t \in T$ such that ${}^\bullet t = \mathbf{0}$. Then $t^\bullet \neq \mathbf{0}$ by the definition of a net. Let $p \in t^\bullet$. Because there are no preconditions to t, t has concession arbitrarily often, and in particular, the event $M_0 \xrightarrow{tt} {}^*M$ may occur, with the result that $M_p > 1$. This contradicts the safeness of N. □

To prove the next proposition which provides a necessary condition for the liveness of safe nets, the following observation must be made.

Let $N = (P, T, pre, post)$ be an unmarked Petri Net with two markings M, M' of N such that $M' \geq M$ (i.e. $\forall p \in P : M'_p \geq M_p$). Suppose that $M \xrightarrow{\tau} {}^*M_1$ where $\tau \in T^*$ is a transition sequence. Then $M' \xrightarrow{\tau} {}^*M'_1$ where M'_1 satisfies

$$\forall p \in P : M'_{1p} = M_{1p} + (M'_p - M_p).$$

In other words, if τ is a transition sequence at M then it is also a transition sequence at M'. Moreover, the resultant markings M_1 and M'_1 are related to each other exactly as M and M' are related.

Proposition 26 *If a safe Petri net* $N = (P, T, pre, post, M_0)$ *is live then* $\forall t \in T : t^\bullet \neq \mathbf{0}$.

Proof: [4] Assume that the safe net $N = (P, T, pre, post, M_0)$ is live. Suppose that $t \in T$ such that $t^\bullet = \mathbf{0}$. Let $M_0 \xrightarrow{\tau} {}^*M'$ such that t is enabled at M'. The existence of τ and M' are assured because of the assumption that N is live. Let $M' \xrightarrow{t} M$. Then clearly $M' \geq M$ since $t^\bullet = \mathbf{0}$. In fact, for each $p \in {}^\bullet t$, $M'_p > M_p$. Since N is live and $M \in \mathcal{R}_N(M_0)$ it is possible to find a transition sequence τ_1 and a marking M_1 such that $M \xrightarrow{\tau_1} {}^*M_1$ and t is enabled at M_1. By the preceding observation, a marking M'_1 may be found such that $M' \xrightarrow{\tau_1} {}^*M'_1$ where M'_1 satisfies

$$\forall p \in P : M'_{1p} = M_{1p} + (M'_p - M_p).$$

Let $p \in {}^\bullet t$. Such a p exists because from $t^\bullet = \mathbf{0}$ it follows that ${}^\bullet t \neq \mathbf{0}$. Then $M_{1p} > 0$ because t is enabled at M_1. As noted earlier, from $M' \xrightarrow{t} M$ it follows that $M'_p > M_p$. Hence $M'_{1p} > 1$, which contradicts the safeness of N. □

The following Corollary is a simple consequence of the preceding four propositions, and represents at once a necessary condition for Liveness and Safeness.

Corollary 27 *If $N = (P, T, pre, post, M_0)$ is live and safe then $\forall x \in X_N : {}^\bullet x \neq \mathbf{0} \neq x^\bullet$.*

The remainder of this chapter assumes that all Petri Nets are both finite and Weakly Connected. These restrictions permit one further result in this section on general Petri Nets.

Theorem 28 *If a (finite, weakly connected) Petri Net N has a live and safe marking then it is strongly connected.*

Proof: See [3]. Actually, this result holds for a live and n-bounded net, for any $n \in \mathbb{N}$. □

4.3.2 Results for P- and T-nets

The behavioural Structure Theory of P-nets is rather simpler than that of T-nets, and as such will be presented first; intuitively, a strongly connected P-net is live iff it carries *at least* 1 token, and safe iff it carries *at most* 1 token.

Proposition 29 *A P-net $N = (P, T, pre, post, M_0)$ is live iff it is strongly connected and $M_0 > \mathbf{0}$, i.e. there is at least one token.*

Proof: [4]

[$\Longrightarrow$] Note that in the empty initial marking (i.e. $M_0 = \mathbf{0}$) none of the transitions of N is live; hence N is also not live, since $T \neq \phi$ by the definition of a Petri Net. Next, it will be seen that liveness implies strong connectedness (assuming weak connectedness). To this end, consider the strongly connected components of N, i.e. all sets $R \subseteq X_N$ such that $\forall x, y \in R : x \prec y$ and $y \prec x$, treating the components of X_N as a partially ordered set. This defines a partitioning of X_N. Define $R_1 \sqsubseteq R_2$ where R_1 and R_2 are two strongly connected components. If $\exists x \in R_1, y \in R_2 : x \prec y$, then $\sqsubseteq$ is a partial order on the set of strongly connected components of N. Since N is finite, $\sqsubseteq$ has maximal elements. Because a path leads from every place of N into some maximal element of $\sqsubseteq$, every token of M_0 can be moved into one of the maximal elements of $\sqsubseteq$ where it will stay. Hence by the liveness of N, $\sqsubseteq$ cannot have any non-maximal elements. But then by the weak connectedness of N, $\sqsubseteq$ cannot have more than *one* maximal element. Hence the unique maximal element of $\sqsubseteq$ is X_N itself, which implies that N is strongly connected.

[$\Longleftarrow$] Consider any marking $M \in \mathcal{R}_N$ and any transition $t \in T$. It is the case that $|M| = |M_0|$; this follows directly from the definition of a P-net and the transition rule. Because $|M| = |M_0| > 0$, there is a place $p \in P$ which is marked under M, i.e. $M_p > 0$. By strong connectedness, a directed X_N-path can be found which leads to t, and this path can be used to enable t; hence N is live.

□

Proposition 30 *A live P-net $N = (P, T, pre, post, M_0)$ is safe iff $|M_0| = 1$, i.e. there is exactly one token.*

Proof: [4]

[$\Longrightarrow$] Use Proposition 29 to show that $|M_0| > 0$ and N is strongly connected. If $|M_0| > 1$ then either there is a place $p \in P$ such that $M_{0p} > 1$ (in which case the Net N is not safe in the initial marking), or there are places $p_1, p_2 \in P$ such that $p_1 \neq p_2$ and $M_{0p_1} + M_{0p_2} > 1$. By strong connectedness, a path leads from p_1 to p_2 and the net is not safe. Hence $|M_0| = 1$.

[$\Longleftarrow$] It is sufficient to notice that $|M| = |M_0|$ for all $M \in \mathcal{R}_N$, since N is a P-net.

□

The following results for T-nets were determined very early in the development of Net Theory, and were originally phrased for Marked Directed Graphs [6,12].

Theorem 31 *A T-net $N = (P, T, pre, post, M_0)$ is live iff all of its simple cycles carry at least one token, and for all places $p \in P$: $|{}^\bullet p| = 1$.*

Proof: [6,12] If some simple cycle of N is unmarked then no transition of this cycle may occur (as N is a T-net); since the number of tokens in a cycle cannot be changed by other transitions firing (as T-net places are unbranching), no transition can be made firable through firings.

Now assume that all simple cycles of N are marked, and let $t \in T$ be any transition of N. Consider the unmarked places $p \in P$ in ${}^\bullet t$. If there are none then t is firable; otherwise consider the transitions t' with $p \in P$ in $t'^\bullet$ (and there are some places in ${}^\bullet t$, since $|{}^\bullet p| = 1$ for all $p \in P$). If each of these is immediately firable, then, clearly, t will become firable after every one of them has fired. If some are not, consider the unmarked places $p' \in P$ in ${}^\bullet t'$, etc. As this backtracking continues, a subnet is being constructed of t, the places preconnected to t, the transitions postconnected to those places, etc. This process must terminate, and the generated subnet must be cycle-free , since N is finite and since there are no unmarked simple cycles in N, respectively. Thus, the subnet must contain at least one transition which has no preceding places belonging to the subnet. This transition is firable in the present marking of N. After firing it, the subnet of the token-free backtracking from t is reduced in size (by one transition). By repeating this process, t is firable and hence t is live. As t was chosen arbitrarily, N is also live. □

A necessary condition for the safeness of a live T-net is given in the following.

Theorem 32 *A live T-net $N = (P, T, pre, post, M_0)$ is safe iff it is covered by simple cycles which carry at most one token.*

Proof: [6,12] If, for each place $p \in P$, there exists a simple cycle upon which p resides, with exactly one token on it, then the token count of this cycle will remain constant (since T-nets prohibit branching places) and thus the cycle is safe.

Assume that there exist two transitions α and β with a place p between them (i.e. $\alpha^\bullet{}_p = 1$ and ${}^\bullet\beta_p = 1$), such that all simple cycles passing through α, p, and β carry 2 or more tokens. It must be demonstrated that by a sequence of transition occurrences it is possible to place 2 (or more) tokens on p.

If there are no tokens on p, backtrack the token-free subnet, as in Theorem 31, starting with transition α. Thus, α may be made firable, and firing it places 1 token on p. If this

construction is repeated, again, the token-free subnet backtracked from α does not include β, since that would imply the existence of a cycle of token-count 1 through p. Thus α may be fired again, without firing β, and therefore placing a second token on p, and thus contradicting the safeness of N. □

The following results tie liveness of T-nets to Strong Connectivity, via the notion of a reproducible marking.

Definition: A marking $M \in \mathcal{R}_N(M_0)$ of a Petri Net $N = (P,T,pre,post,M_0)$ is called *reproducible* iff there is a nonempty sequence $\sigma \in T^*$ of transition occurrences which lead from the marking M back to itself (i.e. $M \xrightarrow{\sigma} {}^*M$, meaning that it is the case that $M \in \mathcal{R}_N(M)$). □

Theorem 33 *A strongly connected T-net $N = (P,T,pre,post,M_0)$ is live iff its initial marking M_0 can be reproduced by a transition sequence σ in such a way that every transition occurs exactly once in σ.*

Proof:

[$\Longrightarrow$] Rather complex – see [6,12].

[$\Longleftarrow$] The reproducing sequence σ requires at least one token on each cycle, and by Theorem 31, N is live. □

The following corollary is a simple consequence of the preceding results.

Corollary 34 *In a strongly connected T-net $N = (P,T,pre,post,M_0)$ the following are equivalent.*

1. *N is live.*
2. *All simple cycles of N carry at least one token.*
3. *The initial marking of N is reproducible such that every transition occurs exactly once.*

Theorem 35 *A T-net N can be endowed with a live and safe marking iff it is strongly connected.*

Proof: [6,12] Clearly, a live marking is obtained by putting one token on each place of N (by Theorem 31). The technique used in the proofs of Theorems 31 and 32 may now be employed to change this marking until it becomes safe, without changing its liveness.

Assume that for a given place the least token count for the simple cycles through it is $k > 1$. It is possible to describe a sequence of transition occurrences that will bring k tokens to this place. By lifting $k - 1$ of them, no cycle becomes token-free and there is now a circuit through this place with a token count of 1. If this is repeated until there are no places lying on cycles with a token count greater (or less) than 1, N is now both live and safe. □

The preceding results highlight the deductive power which is available when dealing with P- and T-nets. This power is due, in the main, to their simple structure, and it is because this structure is often found to be *too* restrictive that the generalisation to Free Choice nets was made. Despite this increased power and complexity, it is still possible to determine some important results for Free Choice nets, as illustrated in the next section.

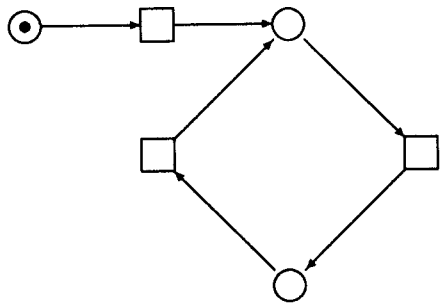

Figure 4.3: A Net which is Deadlock-free but not Live

4.3.3 Results for Free Choice (FC) Nets

Several significant results have been determined for Free-Choice Nets, highlighting their amenable nature and their importance as a class of Petri Net. Some of the results refer to a weaker, easier to check, property than liveness, called *deadlock-freedom*. It must be emphasised that care is needed to distinguish between the classical use of the word "deadlock" (denoting non-live systems) and the specific term (introduced in section 3.3.3) to describe a subset of transitions of a FC net, as both are used in this section.

Definition: A Petri Net $N = (P, T, pre, post, M_0)$ is called *deadlock-free* iff for all $M \in \mathcal{R}_N(M_0)$. $\exists t \in T : M$ enables t. □

Liveness implies deadlock-freedom, by definition, but the converse is not true, as illustrated by the net of figure 4.3 which is deadlock-free, but not live. The following results relate Deadlock-freedom and liveness for safe FC nets.

Theorem 36 *Let $N = (P, T, pre, post, M_0)$ be a safe FC net which is deadlock-free and let $t \in T$ be such that $\forall t' \in T : t' \prec t$ (i.e. t can be reached from all other transitions). Then t is live.*

Proof: [3] The proof is by contradiction. Assume that t is not live; there then exists a marking $M_1 \in \mathcal{R}(M_0)$ at which t is dead, i.e. no successor marking of M_1 enables t.

Consider any $p \in P$ in $\bullet t$; then, by the definition of a FC net, all $t' \in T$ in $p^\bullet$ (not just t itself) are dead at M_1. But this implies that any token put on p after M_1 will stay there.

By safeness, if follows that the transitions in $\bullet p$ can occur at most once, i.e. there is a marking $M_2 \in \mathcal{R}(M_1)$ at which all transitions in $\bullet p$ are dead.

Since this holds for all $p \in \bullet t$ (and since the net is finite), there is same $M_3 \in \mathcal{R}(M_1)$ at which all transitions in $\bullet(\bullet t)$ are dead.

Repeating this argument shows that every transition in the set $\{t' \in T \mid t' \prec t\}$ can be made dead; but since, by assumption, the latter set equals T, this means that a deadlock can be reached and the contradiction follows. □

Corollary 37 *Let $N = (P, T, pre, post, M_0)$ be a safe, strongly connected FC net. Then N is live iff N is deadlock-free.*

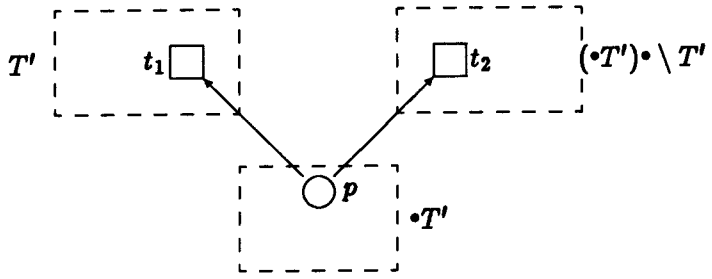

Figure 4.4: Illustrating the proof of Lemma 38

Characterisation of Liveness of FC Nets

The liveness of a Free Choice net may be characterised (i.e. both necessary and sufficient conditions given) as shown in theorem 42 below; the following lemmata are required to support the theorem. This is one of the most significant Petri Net analysis results, and is due to Commoner and Hack (although this proof is from [47]).

Lemma 38 *Let $N = (P, T, pre, post, M_0)$ be a Free Choice net, and let $T' \subseteq T$. If $(\bullet T')\bullet$ may be enabled in $\mathcal{R}(M_0)$ then T' may be enabled in $\mathcal{R}(M_0)$ too.*

Proof: [47] Let $t_1 \in T'$, $p \in P$ in $\bullet t_1$ and $t_2 \in P\bullet \setminus T'$. Since $t_1 \neq t_2$ it is the case that $p\bullet \neq \{t_1\}$ and $p\bullet \neq \{t_2\}$. By the definition of a Free Choice net, $\bullet t_1 = \bullet t_2 = \{p\}$. t_2 is enabled iff p is marked, but in this case, t_1 is enabled, too. □

The following three lemmata refer to the set $\overline{M}$ which is defined to be the set of places which carry no tokens in a given marking M (i.e. $\overline{M} = \{p \in P \mid M_p = 0\}$).

Lemma 39 *Let $N = (P, T, pre, post, M_0)$ be a Free Choice net, and let $T' \subseteq T$ be a set of transitions, none of which is enabled at any marking in $\mathcal{R}(M_0)$. Then there exists a marking $M \in \mathcal{R}(M_0)$ such that none of the transitions in $\bullet(\bullet T' \cap \overline{M})$ is enabled by any marking in $\mathcal{R}(M)$.*

Proof: [47] Let $M \in \mathcal{R}(M_0)$ be a marking such that $\exists$ a transition $t \in \bullet(\bullet T' \cap \overline{M})$ which fires to a marking M_1 and thereby marks a place $p \in \bullet T' \cap \overline{M}$. Using Lemma 38, the transitions firing from M_0 to M_1 do not belong to $(\bullet T')\bullet$. Hence all places of $\bullet T' \setminus \overline{M_0}$ are marked under M_1 too, and therefore, in $\bullet T$, only the places of $\bullet T \cap \overline{M_0}$ are unmarked. Since p is marked under M_1, it is the case that $\bullet T' \cap \overline{M_1} \subset \bullet T' \cap \overline{M} \subseteq \bullet T' \cap \overline{M_0}$.

By iterating this procedure (starting from M_1), it is possible to find in finitely many steps a marking M' such that $\bullet(\bullet T' \cap \overline{M'})$ may not be enabled in $\mathcal{R}(M')$. Otherwise all elements of $\bullet T'$ could be marked. □

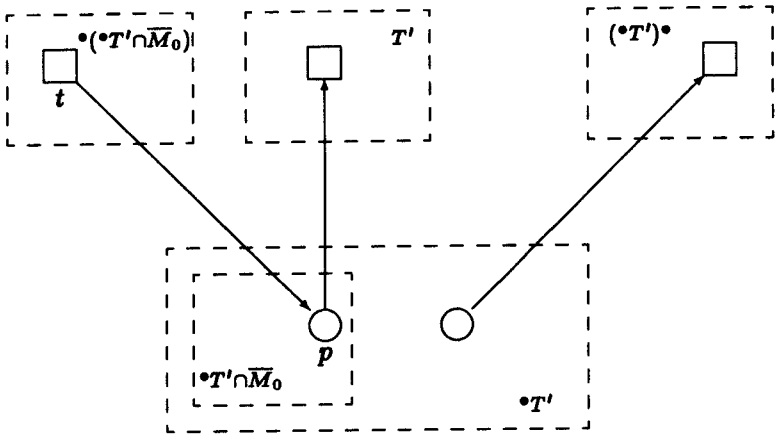

Figure 4.5: Illustrating the proof of Lemma 39

Lemma 40 *Let $N = (P, T, pre, post, M_0)$ be a Petri Net and let $T' \subseteq T$. If $\bullet(\bullet T' \cap \overline{M}_0) \subseteq T'$ then either there exists a transition in T which has concession at M, or $\bullet T' \cap M_0$ is an unmarked deadlock.*

Proof: [47] Assume that no transition in T' has concession at M_0. Let $Q = \bullet T' \cap M_0$ and let $t \in \bullet Q$. By the hypothesis, $t \in T'$. Since T' does not have concession at M_0, $\bullet t \cap \overline{M}_0 \neq \phi$ and hence $\bullet t \cap Q \neq \phi$, that is, $t \in Q\bullet$. Since this is true for each $t \in \bullet Q$ it is the case that $\bullet Q \subseteq Q\bullet$ and $\bullet T' \cap \overline{M}_0$ is a deadlock (and unmarked). □

Lemma 41 *Let $N = (P, T, pre, post, M_0)$ be a Free Choice net and let $T' \subseteq T$ be a set of transitions, none of which is enabled by any marking in $\mathcal{R}(M_0)$. Then there exists a marking $M \in \mathcal{R}(M_0)$ and a deadlock of N which is unmarked under M.*

Proof: [47] By induction on $|T \setminus T'|$.

$|T \setminus T'| = 1$: Since $T = T'$, trivially $\bullet(\bullet T' \cap \overline{M}_0) \subseteq T$, where $\overline{M}_0$ denotes the set of places unmarked under M_0. Using Lemma 40, $\bullet T' \cap M_0$ is an unmarked deadlock.

Induction Hypothesis: The proposition is true if $|T \setminus T'| = n$. Now let $|T \setminus T'| = n+1$. Using Lemma 39, there exists a marking $M \in \mathcal{R}(M_0)$ such that no transition $\bullet(\bullet T' \cap \overline{M})$ may be enabled in $\mathcal{R}(M)$. If $\bullet(\bullet T' \cap \overline{M}) \subseteq T$ the result follows using Lemma 40. Otherwise, let $t \in \bullet(\bullet T' \cap M) \setminus T'$. Since $T' \cup \{t\}$ may not be enabled in $\mathcal{R}(M)$ (Lemma 39) and $|T \setminus (T' \cup \{t\})| = n$, the induction hypothesis yields that there exists a marking $M' \in \mathcal{R}(M)$ such that some deadlock of N is unmarked under M'. In particular, $M' \in \mathcal{R}(M_0)$. □

It is now possible to present the main result of this section;

Theorem 42 *Let $N = (P, T, pre, post, M_0)$ be a free choice net. If every nonempty deadlock of N contains a trap which is marked under the initial marking M_0 of N then N is live.*

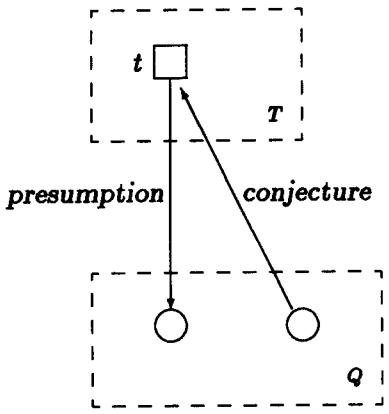

Figure 4.6: Illustrating the proof of Lemma 40

Proof: [21,47] If N is not live then there exists a marking $M \in \mathcal{R}(M_0)$ and a nonempty set of transitions which may not be enabled in $\mathcal{R}(M)$. Then, using Lemma 41 above, there exists a marking $M' \in \mathcal{R}(M_0)$ and a deadlock Q which is unmarked under M'. Theorem 11 states that Q may not become empty in $\mathcal{R}(M_0)$ if Q contains a trap which is marked under M_0. □

This result characterises the liveness of Free Choice nets in a manner which has proved impossible for general Petri Nets, and represents one of the more significant Petri Net Theory results yet achieved.

4.3.4 Liveness and Safeness under Net morphisms

Best has shown that the definition of morphism and bisimulation for Petri Nets preserves liveness and safeness as shown in the following two theorems.

Theorem 43 *Suppose that $N = (P, T, pre, post, M_0)$ and $N' = (P', T', pre', post', M_0')$ are two Petri Nets and that there exists a bisimulation morphism (η, β) relating N and N'. Then N is n-safe iff N' is n'-safe (for some $n, n' \in \mathbb{N}$).*

Proof: [3]

[$\Longrightarrow$] Assume that N is n-safe for some $n \in \mathbb{N}$; hence the reachability class $\mathcal{R}_N$ of N will be finite. It follows from this that

$$\bigcup_{M \in \mathcal{R}_N} \beta(M)$$

is finite, since β is injective, and hence $\mathcal{R}_{N'}$ is finite, since

$$\left|\mathcal{R}_{N'}\right| \leq \sum_{M \in \mathcal{R}_N} \left|\beta(M)\right|.$$

Thus N' is also safe.

[$\Longleftarrow$] Conversely, assume that N' is n'-safe for some $n' \in \mathbb{N}$. Then $\mathcal{R}_{N'}$ is finite, and hence $\mathcal{R}_N$ is also finite, because β^{-1} is surjective and thus $\left|\mathcal{R}_N\right| \leq \left|\mathcal{R}_{N'}\right|$. Thus N is also safe.

□

Theorem 44 *Let $N = (P, T, pre, post, M_0)$ and $N' = (P', T', pre', post', M_0')$ be two Petri Nets and suppose that there exists a bisimulation morphism $(\eta, \beta) : N \longrightarrow N'$. Then N is live iff $\forall t' \in \eta(T) : t'$ is live in N'.*

Proof: [3]

[$\Longrightarrow$] Assume that N is live. Let $M_0' \xrightarrow{w} {}^*M_1'$ for $w \in (\mu T)^*$ and let $t' \in \eta(\mu T)$. By the definition of Petri Net bisimulation, parts (1) and (2b), $M_0 \xrightarrow{\eta^{-1}(w)} {}^*M_1$ where $M_1 = \beta^{-1}(M_1')$. Because N is live, there exists $v \in (\mu T)^*$ such that

$$\exists M_2 \in \mathcal{R}_N(M_0) : M_1 \xrightarrow{v} {}^*M_2$$

and t occurs in v. By (2a) and because β is injective, there are a sequence $v' \in (\mu T')^*$ and a marking $M_2' \in \mathcal{R}_N(M_0')$ such that $M_1' \xrightarrow{v'} {}^*M_2'$ and t' occurs in v'. Hence all $t' \in \eta(\mu T)$ are live in N'.

[$\Longleftarrow$] Assume that all $t' \in \eta(\mu T)$ are live in N'. Let $M_0 \xrightarrow{v} {}^*M_1$ for $v = v_1 v_2 \ldots v_m \in (\mu T)^*$ and let $t \in \mu T$. By the definition of Petri Net bisimulation, parts (1) and (2a), there exist $w_1, \ldots, w_m : M_0' \xrightarrow{w_1 \ldots w_m} {}^*M_1'$ and $M_1 = \beta^{-1}(M_1')$. Since all $t' \in \eta(\mu T)$ are live in N', there are a sequence $w' \in (\mu T)^*$ and a marking $M_2' \in \mathcal{R}_{N'}(M_0')$ such that $M_1' \xrightarrow{w} {}^*M_2'$ and $\eta(t)$ occurs in w'. By (2b), $M_1 \xrightarrow{\eta^{-1}(w')} {}^*M_2$ for $M_2 = \beta^{-1}(M_2')$ and t occurs in $\eta^{-1}(w')$ by the definition of η^{-1}. Hence N is live.

□

These results complete the Petri Net analysis picture, by showing that bisimulation morphisms of Petri Nets preserve dynamic behaviour at least as far as the important safeness and liveness considerations, and give a tidy conclusion to this discussion of Petri Net Theory.

Chapter 5

Conclusion

This report has attempted to summarise and correlate as much as possible of the varying research into the theory of Petri Nets, and to present the whole in a reasonably standardised notation. An important objective was to preserve the power of multiset/composite event notation, and to emphasise the ablility of nets to express truly concurrent events.

Despite its size, this report has neglected to consider many important areas of Net Theory, and has abbreviated many others — notwithstanding these deficiencies, it is hoped that the result is still a useful, detailed introduction to the field, and that the presentation of selected proofs has adequately illustrated the depth and elegance of Petri Net Theory.

Appendix A

Multisets and Multirelations

Multisets and multirelations are used and manipulated throughout this report, and as these structures are defined in many different ways in the literature, the definitions used in the body of the report are stated here; they are taken from [55].

A.1 Vectors and Operations on Vectors

A vector f over a set X is a function $f : X \to \mathbb{Z}$ assigning an integer to each element of X. Write f_x for the x-component of f (the member of $\mathbb{Z}$ corresponding to $x \in X$. Call νX the *space* of vectors over X, and X its *basis*. Say that a vector is *finite* if all but finitely many of its components are 0.

Useful operations and relations on vectors are induced pointwise by operations and relations on integers. Letting f and g be vectors over the set X, define

$$\begin{aligned}(f+g)_x &= f_x + g_x \\ (f-g)_x &= f_x - g_x\end{aligned}$$

to be the sum and difference of vectors, for $x \in X$, and

$$f \leq g \quad \text{iff} \quad \forall x \in X.\ f_x \leq g_x,$$

to be an ordering relation on vectors.

Define the *projection* $\Pi_S^X : f \longmapsto g$ of a vector $f \in \nu X$ to a vector $g \in \nu S$, where S is a subset of X, componentwise to be

$$\left(\Pi_S^X(f)\right)_s = f_s \quad \text{for all } s \in S.$$

Letting $n \in \mathbb{Z}$ and $f, g \in \nu X$, the scalar product nf is defined as $(nf)_x = nf_x$ for $x \in X$, and the inner (dot) product $f \cdot g$ is defined to be

$$f \cdot g = \sum_{x \in X} f_x \cdot g_x$$

when the set $\{x \in X \mid f_x \cdot g_x \neq 0\}$ is finite, and undefined otherwise; finally, the vector product ("intersection") $f \cap g$ is defined componentwise to be

$$(f \cap g)_x = f_x \cdot g_x \quad \forall x \in X.$$

A.2 Multisets

A multiset over a set X is a vector in which all of the components are nonnegative, i.e. a function $f : X \to \mathbb{N}$. Call μX the *space* of multisets over X, and X its *basis*. By convention, subsets of X are identified with those multisets $f \in \mu X$ such that $f_x \leq 1$ for all $x \in X$. The *cardinality* of a multiset $f : X \to \mathbb{N}$ is defined as

$$|f| = \sum_{x \in X} f_x,$$

and represents the number of "members" of f, where members are weightings over X.

The vector operations of addition, subtraction, projection and scalar, inner and vector product restrict to multisets, but for two multisets f and g, the difference $f - g$ is defined (i.e. is a multiset) iff $g \leq f$.

Some particular multisets are $\mathbf{0}$, the *null* multiset over any set X, which is defined as the function $\mathbf{0} : x \longmapsto 0$ for all $x \in X$, and $\hat{x}$, the *singleton* multiset for each element x of a set X, which is defined to be the function

$$\hat{x} : y \longmapsto \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

For convenience, a particular multiset f may sometimes be represented as a set containing f_x copies of each element x, so that for example $f = \{a, a, b, b, b\}$ represents the multiset over {a, b, c} with $f_a = 2$, $f_b = 3$, and $f_c = 0$.

A.3 Natural Numbers extended by ∞

In order to manipulate the object ∞, it is necessary to expand the ordering $<$ and the operations $+$, $-$ and $\cdot$ defined on $\mathbb{N}$ to $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$ as follows:

$$\begin{aligned} \forall n \in \mathbb{N} \quad &: \quad n < \infty, \\ \forall n \in \mathbb{N}^\infty \quad &: \quad \infty + n = n + \infty = \infty \\ \forall n \in \mathbb{N} \quad &: \quad \infty - n = \infty \\ \forall n \in \mathbb{N}^\infty - \{0\} \quad &: \quad \infty \cdot n = n \cdot \infty = \infty \\ \text{but} \quad &\quad \infty \cdot 0 = 0 \cdot \infty = 0. \end{aligned}$$

Define sums over $\mathbb{N}^\infty$ as follows: let $\{f_i \mid i \in I\}$ be an *indexed sum* of $\mathbb{N}^\infty$. Say that such an indexed sum is *finite* precisely in the case where each $f_i \neq \infty$, for $i \in I$, and the set $\{i \in I \mid f_i \neq 0\}$ is finite.

When $\{f_i \mid i \in I\}$ is finite, define $\sum_{i \in I} f_i$ to be the usual sum and otherwise to be ∞.

A.4 Matrices and Operations on Matrices

A $\mathbb{Z}$-matrix from a set X to a set Y is a vector $\alpha : Y \times X \to \mathbb{Z}$ which associates an integer $\alpha_{y,x}$ with each pair $(y, x) \in Y \times X$. Write $\alpha : X \to_\nu Y$ to mean that α is a $\mathbb{Z}$-matrix from X to Y. As matrices are vectors, they inherit the aforementioned vector operations such as sum, difference and products.

Let $f \in \nu X$ be a vector over a set X, and let $\alpha : X \to_\nu Y$ be a $\mathbb{Z}$-matrix from X to another set Y. Define the *application* of α to f to be the vector $\alpha f \in \nu Y$ satisfying

$$(\alpha f)_y = \sum_{x \in X} \alpha_{y,x} \cdot f_x \ ,$$

for all $y \in Y$, provided each indexed sum of integers $\{\alpha_{y,x} \cdot f_x \mid x \in X\}$ is finite; otherwise take αf to be undefined.

Let $\alpha : X \to_\nu Y$ and $\beta : Y \to_\nu Z$ be two matrices, over sets X, Y, and Z. Define their *composition* $\beta \circ \alpha : X \to_\nu Z$ to be the matrix given by

$$(\beta \circ \alpha)_{z,x} = \sum_{y \in Y} \beta_{z,y} \cdot \alpha_{y,x} \ ,$$

for $x \in X, z \in Z$, provided each sum is defined; otherwise take the matrix composition to be undefined.

The *projection* $\Pi_{S,T}^{X,Y} : \alpha \to \beta$ which maps a matrix $\alpha : Y \times X \to \mathbb{Z}$ to another matrix $\beta : T \times S \to \mathbb{Z}$, where $S \subseteq X$ and $T \subseteq Y$, is defined componentwise to be the vector associating an integer $\beta_{t,s} = \alpha_{t,s}$ with each pair $(t, s) \in T \times S$.

A.5 Multirelations

A multirelation from X to Y is a matrix $\alpha : X \to_\nu Y$ in which all entries $\alpha_{y,x}$ are nonnegative, i.e. $\alpha : Y \times X \to \mathbb{N}$. Write $\alpha : X \to_\mu Y$ to mean that α is a multirelation from X to Y.

By convention, the relations between a set X and a set Y are identified with those multirelations $\theta : X \to_\mu Y$ for which $\theta_{x,y} \leq 1, \forall x \in X, \forall y \in Y$. Write xRy to denote that x and y are in the relation R.

Before the restriction of matrix operations to multirelations may be examined, it is necessary to extend multisets and multirelations to include ∞.

A.6 ∞-multisets and ∞-multirelations

A ∞-multiset over a set X is a function $f : X \to \mathbb{N}^\infty$, which associates f_x, a nonnegative integer or ∞, with each $x \in X$. Let $\mu^\infty X$ denote the space of ∞-multisets over X.

A ∞-multirelation from a set X to a set Y is a matrix $\alpha : Y \times X \to \mathbb{N}^\infty$. Write $\alpha : X \to_\mu^\infty Y$ to mean α is a ∞-multirelation from X to Y.

A.7 Operations on Multirelations

Matrix projection restricts directly to multirelations, but the application and composition operations must be altered slightly to avoid infinite sums.

Let $f \in \mu^\infty X$, and let $\alpha : X \to_\mu^\infty Y$. Define the *application* of α to f to be the vector $\alpha f \in \mu^\infty Y$ satisfying

$$(\alpha f)_y = \sum_{x \in X} \alpha_{y,x} \cdot f_x \ ,$$

where the sums may be infinite.

Let $\alpha : X \to_{\mu}^{\infty} Y$ and $\beta : Y \to_{\mu}^{\infty} Z$ be two ∞-multirelations over sets X, Y, and Z. Define their *composition* $\beta \circ \alpha : X \to_{\mu}^{\infty} Z$ to be the matrix given by

$$(\beta \circ \alpha)_{z,x} = \sum_{y \in Y} \beta_{z,y} \cdot \alpha_{y,x} \ ,$$

where again the sums may be infinite.

Appendix B

Partially Ordered Sets

The section (3.2) on Occurrence nets makes considerable use of several significant results relating to partially ordered sets, the details and selected proofs of which appear here. They are taken in the main from [8,9,47].

B.1 Definitions

A partially-ordered set (abbreviated *poset*) is a structure $\langle X, \prec \rangle$ consisting of a set X together with an irreflexive, transitive (and hence antisymmetric) ordering relation $\prec$ over X.

The ordering relation $\prec$ may be associated with the relations $\preceq$, $\succ$, $\succeq$ and $\prec\!\!\!\!\cdot$ which are defined as follows:

$$\begin{aligned} x \preceq y &\Leftrightarrow (x \prec y) \text{ and } (x = y), \\ x \succ y &\Leftrightarrow x \not\preceq y, \\ x \succeq y &\Leftrightarrow (x \succ y) \text{ and } (x = y), \text{ and} \\ x \prec\!\!\!\!\cdot\, y &\Leftrightarrow (x \prec y) \text{ and } \nexists z : x \prec z \prec y. \end{aligned}$$

for elements $x, y \in X$.

For a poset $\langle X, \prec \rangle$ with elements $x, y \in X$, the relationship $x \prec\!\!\!\!\cdot\, y$ may be represented graphically by a directed arc $x \longrightarrow y$, while $x \prec y$ (but not necessarily $x \prec\!\!\!\!\cdot\, y$) may be represented as $x \rightsquigarrow y$.

Example: The illustration of figure B.1 shows a poset $\langle X, \prec \rangle$ where $X = \{a, b, c, d, e, f\}$ and the $\prec$ relation is defined so that

$$\begin{aligned} a &\prec \{b, c, d, e, f\} & b &\prec \{c, d, f\} \\ c &\prec \{d, f\} & d &\prec \phi \\ e &\prec \{f\} & f &\prec \phi \end{aligned}$$

□

An element u of a poset S is said to be an *upper bound* for a subset $A \subseteq S$ if $x \preceq u$ for all $x \in A$. The element u is a *least upper bound* for A if it is an upper bound and $u \preceq v$ for each upper bound v of A, and hence least upper bounds are unique.

A poset $\langle X, \prec \rangle$ is called *dense* iff the relation $\prec\!\!\!\!\cdot$ is empty (i.e. $\prec\!\!\!\!\cdot \equiv \phi$), meaning that for all $x, y \in X$ such that $x \prec y$ there *always* exists a $z \in X$ such that $x \prec z \prec y$. An

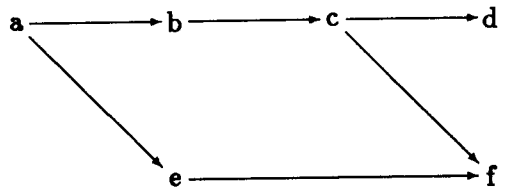

Figure B.1: A partially ordered set

example of a dense poset is $\langle \mathbb{R}, < \rangle$, the real numbers together with the usual "less than" ordering relation on $\mathbb{R}$.

A poset $\langle X, \prec \rangle$ is called *combinatorial* iff the relation $\prec$ is the transitive closure of $\prec$ (i.e. $\prec \equiv (\prec)^+$), meaning that for any $x, y \in X$ with $x \prec y$ there is a finite subset $\{x_1, x_2, \ldots x_n\} \subseteq X$ such that

$$x = x_1 \prec x_2 \prec \cdots \prec x_n = y.$$

The integers determine a combinatorial poset $\langle \mathbb{Z}, < \rangle$, since between any finite integers x and y with $x \prec y$ there is a finite ascending sequence of $y - x$ integers.

B.2 Lines and Cuts of Posets

Let $\langle X, \prec \rangle$ be a partially ordered set. The relations $\underline{li}$ and $\underline{co}$ over X are defined as follows;

$$\begin{array}{rll} \underline{li} \subseteq X \times X & \text{is given by} & a \mathrel{\underline{li}} b \text{ iff } (a \prec b) \text{ or } (b \prec a) \text{ or } (a = b) \\ \underline{co} \subseteq X \times X & \text{is given by} & a \mathrel{\underline{co}} b \text{ iff } \neg(a \mathrel{\underline{li}} b) \text{ or } (a = b) \\ & & \text{(i.e. } a \mathrel{\underline{co}} b \text{ iff } \neg(a \prec b \text{ or } b \prec a)). \end{array}$$

Thus, if $\langle X, \prec \rangle$ is a poset with $x, y \in X$ then either $x \mathrel{\underline{li}} y$ or $x \mathrel{\underline{co}} y$, and if both $x \mathrel{\underline{li}} y$ and $x \mathrel{\underline{co}} y$ then $x = y$.

Example: The $\underline{li}$ and $\underline{co}$ relations corresponding to the poset of figure B.1 are illustrated in figure B.2, where a line x —— y between elements x and y of the poset indicates that x and y are in the appropriate ($\underline{li}$ or $\underline{co}$) relation. □

A subset $Y \subseteq X$, where $\langle X, \prec \rangle$ is a poset, is called a *region* of a relation $\rho \subseteq X \times X$ iff

$$\begin{array}{lll} \forall x, y \in Y. & x \rho y & [\rho \text{is full on} Y], \text{ and} \\ \forall x \in X. & x \notin Y \Longrightarrow \exists y \in Y.\ \neg(x \rho y) & [Y \text{is a maximal subset on which} \rho \text{is full}]. \end{array}$$

Let $\langle X, \prec \rangle$ be a poset, and let $Y \subseteq X$. Then Y is called a *line* iff Y is a region of $\underline{li}$, and Y is called a *cut* iff Y is a region of $\underline{co}$. The set of lines of a poset $\langle X, \prec \rangle$ is denoted $L\langle X, \prec \rangle$, and the set of cuts is denoted $C\langle X, \prec \rangle$. Where $\langle X, \prec \rangle$ is understood, L and C suffice.

Example: In the poset of figure B.1, the lines are $\{a, b, c, d\}, \{a, b, c, f\}$ and $\{a, e, f\}$, while the cuts are $\{e, b\}, \{e, c\}, \{e, d\}$, and $\{d, f\}$. □

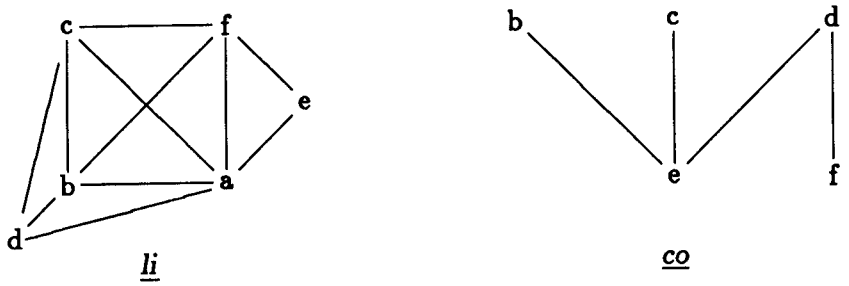

Figure B.2: The $\underline{li}$ and $\underline{co}$ relations

Proposition 45 *Let $\langle X, \prec \rangle$ be a poset, and let $Y \subseteq X$. Then Y is a line iff*

$$(a)\ \ \forall x, y \in Y.\ \ (x \prec y) \vee (y \prec x) \vee (x = y) \text{ and}$$
$$(b)\ \ \forall x \in X \setminus Y.\ \exists y \in Y \text{ with } \neg(x \prec y \vee y \prec x),$$

and Y is a cut iff

$$(a)\ \ \forall x, y \in Y.\ \ \neg(x \prec y \vee y \prec x) \text{ and}$$
$$(b)\ \ \forall x \in X \setminus Y.\ \exists y \in Y \text{ with } x \prec y \text{ or } y \prec x. \ \square$$

B.3 Discreteness Properties of a Poset

Although the definition of a combinatorial poset $\langle X, \prec \rangle$ essentially implies that the elements of X are discrete, there is more than one intensity of "discrete", as described below.

Definition: A poset $\langle X, \prec \rangle$ is called *weakly-discrete* iff

$$\forall x, y \in X.\ \ \forall l \in L\langle X, \prec \rangle.\ \ \big|[x, y] \cap l\big| \in \mathbb{N}\ ,$$

where $[x, y] = \{z \in X \mid x \preceq z \preceq y\}$. $\square$

Thus, if there are no infinite chains between elements in any line of a poset then it is weakly discrete. It is also clear that weakly-discrete implies combinatorial. The poset of figure B.1 is clearly both combinatorial and weakly discrete, while that of figure B.3 is combinatorial but not weakly-discrete, thus showing that the reverse implication does not hold in general.

If, in addition to being finite, chains are all required to have bounded length, the notion of bounded-discreteness is arrived at.

Definition: A poset $\langle X, \prec \rangle$ is called *boundedly-discrete* (abbreviated *b-discrete*) iff

$$\forall x, y \in X.\ \ \exists n \in \mathbb{N}\ :\ \forall l \in L.\ \ \big|[x, y] \cap l\big| \leq n.$$

$\square$

Lemma 46 *If a poset $\langle X, \prec \rangle$ is b-discrete then it is also weakly-discrete.*

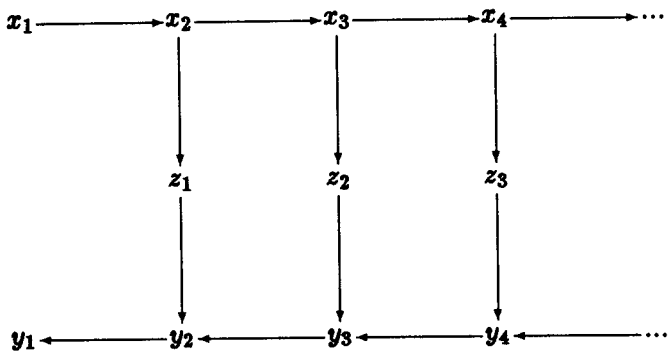

Figure B.3: A poset which is not weakly-discrete

Proof: Obvious, from the definitions of b-discrete and weakly-discrete. □

B-discreteness is closely related to the concept of observability as defined in [53], *viz.* An observer f of $\langle X, \prec \rangle$ is a function $f : X \to \mathbb{Z}$ such that

$$\forall x, y \in X.\ x \prec y \Longrightarrow f(x) \prec f(y).$$

If such an observer exists, $\langle X, \prec \rangle$ is said to be *observable.*

Theorem 47 *If a poset $\langle X, \prec \rangle$ is observable then it is boundedly discrete. If X is countable and $\langle X, \prec \rangle$ is boundedly discrete then $\langle X, \prec \rangle$ is observable.*

Proof: See [53]. □

Although the poset of figure B.4 is both weakly and boundedly discrete, the lines going through x are unbounded (in length). The concept of discreteness with respect to a cut restricts such situations.

Definition: Let $\langle X, \prec \rangle$ be a poset and let $c \in C\langle X, \prec \rangle$. Say that $\langle X, \prec \rangle$ is *discrete with respect to c* iff

$$\forall x \in X.\ \exists n \in \mathbb{N}.\ \forall l \in L.\quad \left|[c, x] \cap l\right| \leq n \text{ and } \left|[x, c] \cap l\right| \leq n,$$

where

$$\begin{aligned} [c, x] &= \{z \in X \mid \exists y \in c.\ y \preceq z \preceq x\} \\ [x, c] &= \{z \in X \mid \exists y \in c.\ x \preceq z \preceq y\} \end{aligned}$$

□

Discreteness with respect to a cut is a stronger property than b-discreteness, as indicated by:

Theorem 48 *If there exists a cut c of a poset $\langle X, \prec \rangle$ such that $\langle X, \prec \rangle$ is discrete w.r.t. c then $\langle X, \prec \rangle$ is also boundedly discrete.*

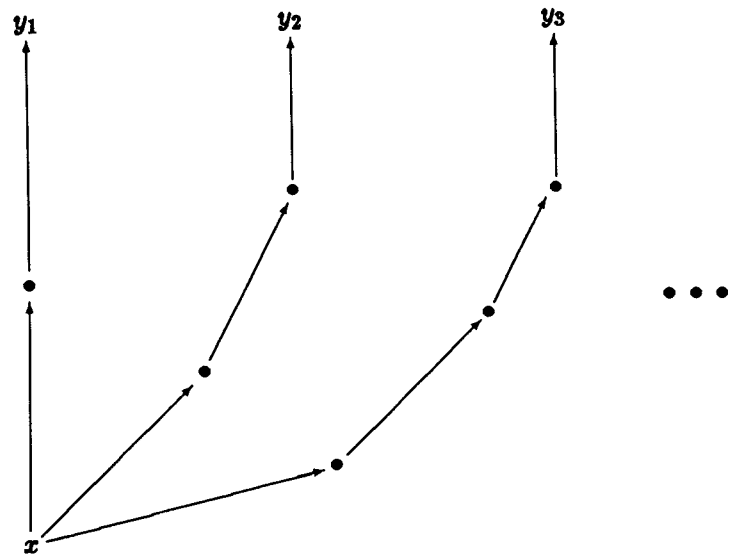

Figure B.4: A Weakly and Boundedly-Discrete Poset

Proof: [9] Let $x, y \in X$. It must be shown that there exists $n \in \mathbb{N}$ such that for all $l \in L$. $\left|[x,y] \cap l\right| \leq n$. If $x \underline{co}\ y$ then the result follows from choosing $n = 0$. Without loss of generality, assume $x \prec y$ and consider the following 3 cases.

1. $[\exists z \in c.\ z \preceq x]$. In this case $[x,y] \subseteq [c,y]$ and thus $\forall l \in L.\ ([x,y] \cap l) \subseteq ([c,y] \cap l)$, but since $\langle X, \prec \rangle$ is discrete w.r.t. c, $\exists n \in \mathbb{N}$ such that for all $l \in L$,

$$\left|[x,y] \cap l\right| \ \leq\ \left|[c,y] \cap l\right| \ \leq\ n$$

   giving the required result.

2. $[\exists z \in c.\ y \preceq z]$. In this case $[x,y] \subseteq [x,c]$ and the result follows similarly to case 1.

3. $[\exists z_1, z_2 \in c : x \prec z_1 \wedge z_2 \prec y]$. In this case, $[x,y] \subseteq [x,c] \cup [c,y]$ and then $\forall l \in L$.

$$[x,y] \cap l \ \subseteq\ ([x,c] \cup [c,y]) \cap l \ =\ ([x,c] \cap l) \cup ([c,y] \cap l) \qquad \text{(B.1)}$$

   and since $\langle X, \prec \rangle$ is discrete w.r.t. c, there exist $n_1, n_2 \in \mathbb{N}$ such that $\left|[x,c] \cap l\right| \leq n_1$ and $\left|[c,y] \cap l\right| \leq n_2$. From (B.1) it follows that $\exists n \in \mathbb{N}$ with $n = n_1 + n_2$ such that

$$\left|[x,y] \cap l\right| \leq n.\ \ \forall l \in L\ ,$$

   and the result again follows.

□

There thus exists the following hierarchy of discreteness properties;

$$\begin{matrix}\text{Discrete}\\ \text{w.r.t. a cut}\end{matrix} \Longrightarrow \begin{matrix}\text{Boundedly}\\ \text{Discrete}\end{matrix} \Longrightarrow \begin{matrix}\text{Weakly}\\ \text{Discrete}\end{matrix} \Longrightarrow \text{Combinatorial}$$

where the implications are, in general, not reversible (see [9,53]).

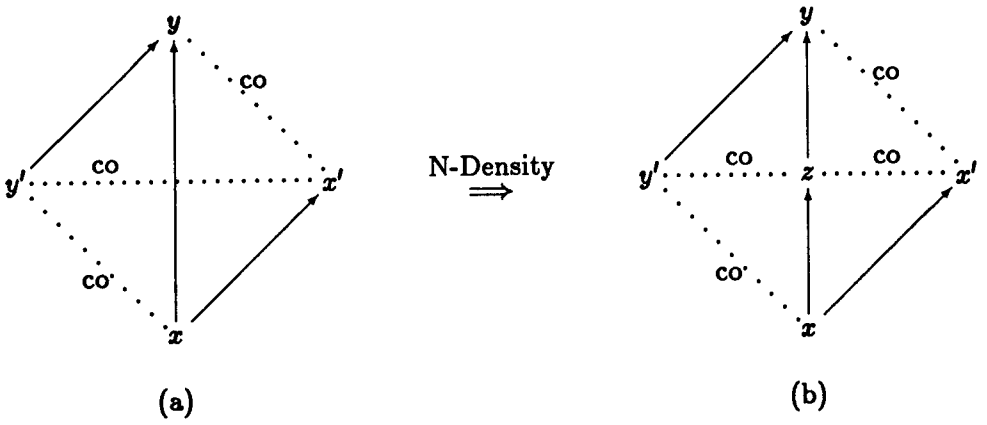

Figure B.5:

B.4 Density Properties of a Poset

Clearly from the definitions of *dense* and *combinatorial* posets in section B.1 of this appendix, a dense poset cannot be combinatorial, and vice versa. The following introduces two notions of density due to by Petri [43] which *do* apply to combinatorial posets.

The first of these density notions, called *N-density*, intuitively means that in an N-shaped diagram like that of figure B.5(a) there must exist a point z between x and y as shown in figure B.5(b).

More formally:

Definition: A poset $\langle X, \prec \rangle$ is called *N-dense* iff for all $x, y, x', y' \in X$ with

$$x \prec y \ \wedge \ x \prec x' \ \wedge \ y' \prec y \ \wedge \ (x \underline{co}\ y' \underline{co}\ x' \underline{co}\ y)$$

there exists $z \in X$ such that

$$x \prec z \prec y \ \wedge \ (x' \underline{co}\ z \underline{co}\ y')$$

□

The second density property is called *K-density* (from German "Kombinatorisch"), which postulates that every element of a line must be in a definitive cut.

Definition: A poset $\langle X, \prec \rangle$ is *K-dense* iff

$$\forall c \in C\langle X, \prec \rangle. \ \forall l \in L\langle X, \prec \rangle. \quad c \cap l \neq \emptyset.$$

□

Note that since $|c \cap l| \leq 1$, $c \cap l \neq \emptyset$ implies that $|c \cap l| = 1$ for all cuts c and lines l. The two forms of density are related by the following result.

Theorem 49 *If a poset $\langle X, \prec \rangle$ is K-dense then it is also N-dense.*

Proof: [9] Letting $x, y, x', y' \in X$ such that

$$x \prec y \ \wedge \ x \prec x' \ \wedge \ y' \prec y \ \wedge \ (x \ \underline{co} \ y' \ \underline{co} \ x' \ \underline{co} \ y)$$

it is required to show that $\exists z \in X$ such that

$$x \prec z \prec y \ \wedge \ (x' \ \underline{co} \ z \ \underline{co} \ y') \, .$$

Let $c \in C$ be such that $\{x', y'\} \subseteq c$ and let $l \in L$ be such that $\{x, y\} \subseteq l$. Since $\langle X, \prec \rangle$ is K-dense, there exists $z \in X$ such that $l \cap c = \{z\}$ and since $z, y', x' \in c$ and $c \in C$ it follows that $x' \ \underline{co} \ z \ \underline{co} \ y'$.

It remains to show that $x \prec z \prec y$, where $x, y, z \in l$. Now if $z \preceq x$ then $z \prec x'$ which leads to a contradiction, since $z \ \underline{co} \ x'$, and hence $x \prec z$. Finally, if $z \succeq y$ then $y' \prec y \preceq z$ which also gives a contradiction, since $y' \ \underline{co} \ z$, and hence $z \prec y$, giving the result that

$$\exists z \in X. \ \ x \prec z \prec y \ \wedge \ (x' \ \underline{co} \ z \ \underline{co} \ y'),$$

and $\langle X, \prec \rangle$ is N-dense as required. □

The reverse implication is not, in general, true.

Appendix C

Vector Addition Systems

The original definition of Vector Addition Systems (VAS) was due to Karp and Miller [30], and that presented here is essentially the same as theirs, but was taken from [20]. VAS are equivalent to the dynamic behaviour of pure (self-loop free) Petri Nets. The pureness restriction is removed in the related model of Vector Replacement Systems (VRS) which has been used by several authors, but which will not be considered here.

Definition: An r-dimensional *Vector Addition System* is a pair $\mathcal{W} = (q, W)$ where q is an r-dimensional vector over the nonnegative integers ($q \in \mathbb{N}$) and W is a finite set of r-dimensional integer vectors ($W \subseteq \mathbb{Z}^r$). The reachability set $\mathcal{R}_\mathcal{W}$ of $\mathcal{W}$ is the set of all vectors of the form

$$q + w_1 + w_2 + \ldots + w_n$$

such that $\forall i \leq n$,

$$w_i \in W \quad \text{and} \quad q + \sum_{j=1}^{i} w_j \geq 0.$$

□

The relationship between a VAS $\mathcal{W}$ and its corresponding Petri Net N is obvious if q is viewed as being the initial marking and w_i represent the effect of transition i on a given (current) marking of the net. Thus $\mathcal{R}_\mathcal{W}$ is precisely the set of reachable markings of the Petri Net N (i.e. $\mathcal{R}_\mathcal{W} = \mathcal{R}_N = \{M \mid M_0 \rightsquigarrow^* M\}$), and the reachability problem may be examined for VAS without requiring the extra machinery present in the definition of a Petri Net.

Example: The confused Petri Net of figure 2.11 has an identical reachability class to that of the VAS given by $\mathcal{W} = (q, W)$ where $q, w_i \in \nu\{1, \ldots, 5\}$ and q is the vector mapping only 1, 4, and 5 to 1, denoted

$$q = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

and $W = \{w_1, w_2, w_3\}$ where

$$w_1 = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, w_2 = \begin{bmatrix} 0 \\ -1 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \text{ and } w_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 1 \end{bmatrix}.$$

The isomorphism from the net to the VAS $\mathcal{W}$ maps each place p_i to the coordinate i in $\{1, \ldots, 5\}$. □

Vector Addition Systems formed the basis of most early attempts at solutions to the Petri Net reachability problem, but it was found that extensions to the basic VAS model were useful, and these involved the notion of a semilinear set.

Definition: A *semilinear set* $SLS = L_1 \cup L_2 \cdots L_n$ is a finite union of linear sets L_i, for $1 \leq i \leq n$, where a set $L \subseteq \mathbb{N}^n$ is called *linear* iff $\exists c, p_1, \ldots, p_m \in \mathbb{N}^n$ such that

$$L = L(c; p_1, \ldots, p_m) = \{c + \sum_{j=1}^{m} x_j p_j \mid x_1, \ldots, x_m \in \mathbb{N}\}.$$

A set is said to be *effectively semilinear* if a semilinear representation of the set may be computed effectively. □

Solutions to the Reachability Problem for VAS (and Petri Nets) make use of extensions of the basic VAS model called Vector Addition Systems with States (VASS) and Generalised Vector Addition Systems with States (GVASS), which are described in the sequel.

Intuitively, a VASS may be viewed as the combination of a VAS and a finite automaton, or alternatively, as a finite directed graph whose edges are labelled by vectors of integers and with one initial and one final node. Each configuration of a VASS comprises a pair (q, x) where q is the name of the state and x is a vector of the underlying VAS.

A *step* $q_i \longrightarrow (q_{i+1}, t)$ of a VASS leads from a configuration (q_i, x_i) to a configuration (q_{i+1}, x_{i+1}) if an arc labelled by a vector t goes from states q_i to q_{i+1} of the automaton, and $x_{i+1} = x_i + t$.

Define $\Pi_A(x, \ldots, x_n) = (x_{i_1}, \ldots, x_{i_k})$ to be the projection on a subset A of the coordinates, where $A = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ with $i_1 < i_2 < \ldots < i_k$, and with $\Pi_A(X) = \{\Pi_A(x) \mid x \in X\}$ and $\Pi_{-A} = \Pi_{\{1,\ldots,n\}-A}$.

Sequences of steps of a VASS define *r-paths* (with vectors $x_i \in \mathbb{Z}^n$), *R-paths* (with vectors $x_i \in \mathbb{N}^n$) and *SR-paths* (if $\Pi_A(x_i) \geq \mathbf{0}$, for vectors x_i) with respect to some subset A of the coordinates

Definition: The Reachability Problem for a VASS is to decide whether there exists an R-path r of the VASS such that $(init, x) \overset{r}{\rightsquigarrow} (fin, y)$. The Reachability Set for a VASS is

$$\mathcal{R}_{(q,x)} = \{(q', y) \mid \exists r : (q, x) \overset{r}{\rightsquigarrow} (q', y)\}.$$

□

As shown by Hopcroft and Pansiot [25], the VAS and VASS constructions are equivalent in the sense that an n-dimensional VASS can be simulated by an $n+3$-dimensional VAS, and that the reachability problem for VASS is thus equivalent to that for VAS.

Kosaraju, in constructing the first (accepted) complete decision procedure for the reachability problem, extended the VASS formalism to what he called *Generalised Vector Addition Systems with States*, or GVASS.

Definition: A GVASS is a finite chain of VASS G_i linked by edges from G_i to G_{i+1}. The G_i are also subjoined by ("constraint") vectors V_i and V_i' which express the fact that no information is known about the behaviour of some components between the G_i (i.e. "don't care" values), and by sets R_i of *rigid* coordinates (which remain invariant between G_i) which satisfy the conditions

1. . $t_i \in \mathbb{Z}^n$,
2. . $V_i, V_i' \in (\mathbb{N}^\infty)^n$,
3. . $S_i = \{j \mid \Pi_j(V_j) \neq \infty\}$,
4. . $S_i' = \{j \mid \Pi_j(V_j') \neq \infty\}$,
5. . $R_i \subseteq S_i \cap S_i'$, and
6. . $\forall t \in G_i.\ \Pi_{R_i}(t) = 0$.

□

Although this construction appears to be more complex than both the VAS and VASS, it is simpler in one important sense, related to the ease with which paths in the GVASS may be analysed.

Definition: A *cr-path* p of a GVASS is a composition of r-paths p_i in the underlying VASSs G_i and connecting edges which satisfies the constraints

$$p : (q_1, x_1) \overset{p_1}{\rightsquigarrow_r} (q_1', y_1) \overset{t_1}{\rightsquigarrow_r} (q_2, x_2) \ldots (q_i, x_i) \overset{p_i}{\rightsquigarrow_r} (q_i', y_i) \overset{t_i}{\rightsquigarrow_r} (q_{i+1}, x_{i+1}) \ldots \overset{p_s}{\rightsquigarrow_r} (q_s', y_s)$$

such that $x_i, y_i \geq \mathbf{0}$ and

$$\Pi_{S_i}(x_i) = \Pi_{S_i}(V_i) \quad \text{and} \quad \Pi_{S_i'}(y_i) = \Pi_{S_i'}(V_i').$$

□

A *CR-path* p is a cr-path which is an R-path from (q_1, x_1) to (q_s', y_s).

Definition: The Reachability Problem for GVASS is to decide whether there exists any CR-path from the initial node to a particular node (q_s', y_s). □

As indicated in the body of this report (Chapter 4), Kosaraju also presented a decision procedure for determining the existence of such a CP-path, thus effectively solving the Petri Net Reachability Problem.

Bibliography

[1] E. Best. The relative strength of k-density. *Lecture Notes in Computer Science,* Springer-Verlag, 84:261–276, 1979.

[2] Eike Best. COSY: its relation to nets and to CSP. *Lecture Notes in Computer Science,* Springer-Verlag, 255:416–440, 1987.

[3] Eike Best. Structure theory of petri nets: the free choice hiatus. *Lecture Notes in Computer Science,* Springer-Verlag, 254:168–205, 1987.

[4] Eike Best and Pazhamaneri S. Thiagarajan. Some classes of live and safe petri nets. *Concurrency and Nets,* Springer-Verlag, 71–94, 1987.

[5] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, April 1986.

[6] F. Commoner, A.W. Holt, S.Even, and A. Pnueli. Marked directed graphs. *Journal of Computer and System Sciences*, 5:511–523, 1971.

[7] E.W. Dijkstra. *A Discipline of Programming.* Prentice-Hall, Englewood Cliffs, 1976.

[8] John R. Durbin. *Modern Algebra: An Introduction.* John Wiley & Sons, Inc., New York, 1979.

[9] César Fernández. Non-sequential processes. *Lecture Notes in Computer Science,* Springer-Verlag, 254:95–115, 1987.

[10] Nissim Francez. *Fairness.* Springer-Verlag, New York, 1986.

[11] H.J. Genrich and E. Stankiewicz-Wiechno. A dictionary of some basic notions of net theory. *Lecture Notes in Computer Science,* Springer-Verlag, 84:519–535, 1979.

[12] H.J. Genrigh and K.Lautenbach. Synchronisationsgraphen. *Acta Informatica*, 2:143–161, 1973.

[13] Ursula Goltz. Synchronic distance. *Lecture Notes in Computer Science,* Springer-Verlag, 254:338–358, 1987.

[14] Ursula Goltz and Yuan Chong-Yi. Synchronic structure. *Lecture Notes in Computer Science,* Springer-Verlag, 222:233–252, 1986.

[15] Ursula Goltz and Alan Mycroft. On the relationship of CCS and Petri nets. *Lecture Notes in Computer Science,* Springer-Verlag, 172, 1984.

[16] Ursula Goltz and Wolfgang Reisig. CSP-Programs as nets with individual tokens. *Lecture Notes in Computer Science,* Springer-Verlag, 188:169–196, 1985.

[17] David Gries. *The Science of Programming.* Springer-Verlag, New York, 1981.

[18] Michel Hack. *Decision Procedures for Petri Nets and Vector Addition Systems.* Project MAC, Computational Structures Group Memo 95, Massachusetts Institute of Technology, Cambridge, MA, March 1974.

[19] Michel Hack. *The Equality Problem for Vector Addition Systems is Undecidable.* Project MAC, Computational Structures Group Memo 121, Massachusetts Institute of Technology, Cambridge, MA, April 1975.

[20] Michel Hack. *The Recursive Equivalence of the Reachability Problem and the Liveness Problem for Petri Nets and Vector Addition Systems.* Project MAC, Computational Structures Group Memo 107, Massachusetts Institute of Technology, Cambridge, MA, August 1974.

[21] Michel Henri Théodore Hack. *Analysis of Production Schemata by Petri Nets.* Project MAC Technical Report TR-94, Massachusetts Institute of Technology, Cambridge, MA, February 1972.

[22] Robert H. Halstead. Multilisp: a language for concurrent symbolic computation. *ACM Transactions on Programming Languages and Systems,* 7(4):501–538, October 1985.

[23] C.A.R. Hoare. Communicating sequential processes. *Communications of the ACM,* 21(8):666–677, August 1978.

[24] A.W. Holt. *The Final Report of the Project on Information Systems Theory.* Technical Report Applied Data Research ADR6606, USAF Rome Air Development Center RADC-TR-68-305, 1968.

[25] John Hopcroft and Jean-Jaques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science,* 8:135–159, 1979.

[26] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[27] Rodney R. Howell and Louis E. Rosier. Recent results on the complexity of problems related to petri nets. *Lecture Notes in Computer Science,* Springer-Verlag, 266:45–72, 1987.

[28] Kurt Jensen. Coloured petri nets. *Lecture Notes in Computer Science,* Springer-Verlag, 254:248–299, 1987.

[29] N. Jones, L. Landweber, and Y. Lien. Complexity of some problems in petri nets. *Theoretical Computer Science,* 4:277–299, 1977.

[30] R.M. Karp and R.E. Miller. Parallel program schemata: a mathematical model for parallel computation. In *IEEE Conference Record, 8th Annual Switching and Automata Theory Symposium,* pages 55–61, October 1967.

of petri nets. *Lecture Notes in Computer Science*, Springer-Verlag, 266:164–181, 1987.

[33] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, July 1978.

[34] L.H. Landweber and E.L. Robertson. Properties of conflict-free and persistent petri nets. *Journal of the ACM*, 25(3):352–364, July 1978.

[35] R. Lipton. *The Reachability Problem Requires Exponential Space.* Department of Computer Science Report 62, Yale University, January 1976.

[36] Ernst Mayr. An algorithm for the general petri net reachability problem. In *Proceedings of the 13th Annual Symposium on Theory of Computing*, pages 238–246, 1981.

[37] Ernst W. Mayr. An algorithm for the general petri net reachability problem. *SIAM Journal on Computing*, 13(3):441–460, August 1984.

[38] Robin Milner. *A Calculus of Communicating Systems.* Volume 92 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1980.

[39] Horst Müller. The reachability problem for VAS. *Lecture Notes in Computer Science,* Springer-Verlag, 188:376–391, 1985.

[40] Mogens Nielsen. CCS - and its relationship to net theory. *Lecture Notes in Computer Science,* Springer-Verlag, 255:393–415, 1987.

[41] James L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice Hall, 1981.

[42] C. A. Petri. *Kommunikation mit Automaten.* PhD thesis, Institut für Instrumentelle Mathematik, Bonn, F.R.G., 1962.

[43] C. A. Petri. *Non-Sequential Processes.* Interner-Bericht ISF-77-5, GMD, St. Augustin, FRG, 1977.

[44] A. Pnueli. Applications of temporal logic to the specification and verification of reactive systems: a survey of current trends. *Lecture Notes in Computer Science,* Springer-Verlag, 224:510–584, 1986.

[45] A. Pnueli. The temporal semantics of concurrent programs. *Theoretical Computer Science*, 13:45–60, 1981.

[46] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6:223–231, 1978.

[47] Wolfgang Reisig. *Petri Nets: An Introduction. EATCS Monographs on Theoretical Computer Science*, Springer-Verlag, Berlin, 1985.

[48] G. Rozenberg and P.S. Thiagarajan. Petri nets: basic notions, structure, behaviour. *Lecture Notes in Computer Science,* Springer-Verlag, 224:585–668, 1986.

[49] Colin Sterling. A complete modal proof system for a subset of SCCS. *Lecture Notes in Computer Science,* Springer-Verlag, 185:253–266, 1985.

[50] Glynn Winskel. A category of labelled petri nets and compositional proof system. 1988. To appear in Proceedings of Theoretical Computer Science Conference, Edinburgh, June 1988.

[51] Glynn Winskel. A complete modal proof system for SCCS with modal assertions. *Lecture Notes in Computer Science,* Springer-Verlag, 206:392–410, 1985.

[52] Glynn Winskel. Event structures. *Lecture Notes in Computer Science,* Springer-Verlag, 255:325–392, 1987.

[53] Glynn Winskel. *Events in Computation.* PhD thesis, University of Edinburgh, 1980.

[54] Glynn Winskel. A new definition of morphism on petri nets. *Lecture Notes in Computer Science,* Springer-Verlag, 166, 1984.

[55] Glynn Winskel. Petri nets, algebras, morphisms, and compositionality. *Information and Computation,* 72:197–238, March 1987.

[56] S.J. Young. *An Introduction to Ada.* Volume 23 of *Computers and their Applications,* Ellis Horwood, Chichester, 1983.