RICE UNIVERSITY

ALGEBRAIC MODELS FOR
ASYNCHRONOUS CONTROL STRUCTURES

by

P. S. Thiagarajan

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING

Thesis Director's signature:

Houston, Texas

May 1972
ABSTRACT

ALGEBRAIC MODELS FOR
ASYNCHRONOUS CONTROL STRUCTURES

by

P. S. Thiagarajan

The study of asynchronous control structures used for enforcing the coordination required to carry out an activity that exhibits concurrency is of considerable interest and importance. A large class of these control structures can be adequately and concisely described by Petri-nets. The Petri-net, being a graphical model, promises to be a powerful tool in the design of such control structures. However, precisely because it is a graphical model, the Petri-net is not amenable to elegant mathematical analysis. Hence it is difficult to find answers to significant questions regarding the structure and behavior of asynchronous control structures if their sole available descriptions are in the form of Petri-nets.

For this reason, the problem of formulating valid algebraic models for the above-mentioned class of control structures is studied. In particular, attention is confined to those control structures that can be described by marked graphs, which are a restricted type of Petri-
nets. Two algebraic models called the C-D model and the Poset model are formulated and their validity is established. Both these models represent the control structure under study by specifying the cause-effect relationship imposed by the control structure on the flow of signals that are associated with it. Finally, as an example of the application of these two models, a simple scheme for realizing marked-graph-describable asynchronous control structures is demonstrated.
I wish to extend my sincere thanks to Professor J. R. Jump, my thesis advisor, for his active and able guidance. I am deeply indebted to him for helping me to refine my vague curiosity concerning the theoretical aspects of computer science into well-defined research goals. Thanks are also due to Professor R. C. Minnick and Dr. S. K. Rusk for serving on the thesis committee and for offering many useful suggestions. Professor E. A. Feustel is to be acknowledged for readily making available part of the related research material. Especial thanks are due to Miss Ruth Parks for her diligent typing of the manuscript. I am grateful to my brothers Bappu and Raju for the sacrifices that they have made towards my education. Finally, it is an immensely pleasurable task to express my deep appreciation for the varied forms of help that I have received from my friends Alberto Kinra, Kattan Kirtane, Susan Carter, Sher Singh, Carol and Shanker Bhatta, Arvind Caprihan and Jim Foltz.

This work was sponsored by National Science Foundation Grant GJ-750.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>THE C-D MODEL</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>THE POSET MODEL</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>THE PRECEDENCE GRAPH IMPLEMENTATION PROCEDURE</td>
<td>79</td>
</tr>
<tr>
<td>5</td>
<td>CONCLUSIONS</td>
<td>125</td>
</tr>
<tr>
<td>A</td>
<td>APPENDIX A</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>BIBLIOGRAPHY</td>
<td>132</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1-1

This thesis is concerned with the formulation of algebraic models to represent the behavior of concurrent systems. In particular, attention has been confined to those concurrent systems whose behavior can be described by safe and live marked graphs [1]. Two algebraic models have been proposed and their validity has been established. As an example of the application of these algebraic models, a simple and systematic scheme for obtaining a physical realization for marked-graph-describable concurrent systems has been developed.

1-2

The term "marked graph" is just one of the many new terms and concepts that have evolved recently in connection with the study of systems and computations that exhibit concurrency. A brief (and greatly simplified) summary of these new concepts is given below in order to aid the subsequent discussions.

A concurrent system is an entity that provides the coordination required to carry out an associated activity.
The main ingredients of an activity are:

1) A finite set of elements called conditions, and
2) A finite set of elements called events.

The activity consists of the occurrences (repeated or otherwise) of these events subject to certain constraints. Events and conditions are related to each other as follows: when certain conditions hold, an event is permitted to occur. When an event occurs, certain conditions cease to hold and certain other new conditions begin to hold, which in turn permit the occurrence of some other event and so on.

The constraints imposed on the occurrences are essentially of two kinds. They are,

1) A global constraint to the effect that the occurrences conform to a particular temporal ordering. In general this temporal ordering is a partial ordering.

2) A local constraint to the effect that under certain situations, the occurrence of an event $E_x$ preclude the possibility of the concurrent occurrence of an event $E_y$ and vice-versa.

In some concurrent systems the second constraint is absent.

The coordination achieved by a concurrent system consists of implementing the constraints imposed on the associated activity. Conversely, the specification of a
concurrent system consists of the specification of these constraints.

Previous work by other researchers which is directly related to this thesis can now be summarized.

1-3

Parallelism as a property of a system first occurred in the work of Muller [2]. He formulated the concept of speed-independent circuits and carried out an extensive analysis of their properties. He also established the rules of communication that the subsystems must follow so that a composite system may function in a speed-independent manner [3]. The realization of control structures which can be modelled as concurrent systems is of considerable interest. In order that a realization of such a system be faithful it is crucial that it be speed-independent. Hence Muller's work has assumed great importance recently.

However, the concept of a concurrent system in all its generality was first formulated in terms of the occurrence system model by Anatol Holt [4]. In fact, the summary given in Section 1-2 is directly based on Holt's work.

One of the more appealing aspects of the occurrence system model (apart from its generality) is that a certain class of occurrence systems can be precisely and concisely
described by a graphical model called a Petri-net. Petri-nets were first proposed and studied by Carl Adam Petri in connection with the modelling of information flow in complicated organizational processes [5]. Marked graphs, the concurrent systems model studied in this thesis, are a restricted class of Petri-nets. Holt has also carried out an exhaustive analysis on the properties of these graphs [1].

Given the description of a system in the form of a Petri-net with suitable interpretations, the problem of obtaining a corresponding physical implementation has been studied by Suhas Patil [6]. The primitives that he uses in his implementation scheme are the so-called Macro modules that have been proposed by Jack B. Dennis [7]. Patil's realization scheme is to be properly thought of as rigorous constructive proof of the assertion that every safe Petri-net with suitable interpretations has a corresponding physical implementation.* Dennis has obtained efficient "implementations" for a number of Petri-nets of varying complexity using his Macro modules [7]. Unfortunately, his approach is unsystematic.

* To be precise, this assertion is correct only if the situation in which the same input signal enables two different input transitions at different times is taken care of. But this can be done as Dennis has demonstrated through the construction of the X-module [7].
The problem of realizing the Macro modules themselves (with the exception of the arbiter module) with smaller primitives called micro modules (Muller’s C-elements, exclusive-or-gates and inverters) has been studied by Suhas Patil [8]. Seitz has also investigated the problem of implementing a certain class of Petri-nets making more direct use of Holt’s work on occurrence systems [9]. Unfortunately his approach at the actual implementation level is the flow-table-approach used for realizing asynchronous switching circuits. Hence, the complexity of this procedure grows exponentially with the "size" of the system.

Bruno, Altman and Denning have studied the properties of loop-free networks of certain macro modules **[10,11]. It is interesting to note that, in these studies, the behavior of networks consisting of WYE, SEQUENCE and JUNCTION modules has been characterized by precedence graphs and I - O relations. One of the major results of this thesis is that a marked-graph-describable system can be completely characterized by a single partially ordered set, and it can be easily shown that networks of SEQUENCE, WYE and JUNCTION modules can be described by marked graphs.

** They are the JUNCTION, SEQUENCE, WYE, SELECT, and INTERATOR modules.
A large class of control structures that exhibit concurrency and function in a speed-independent manner can be looked upon as concurrent systems. Typical examples are:

1) A control structure which is the hardware implementation of a particular resource allocation algorithm in an m-user, n-server environment.

2) A control structure that provides the necessary coordination for carrying out the various steps of a computation that exhibits concurrency in a multi-functional-unit environment.

Such control structures may be termed as asynchronous control structures (and abbreviated as ACS). Using this terminology, it can be seen that in almost all of the related work mentioned in Section 1-2, the concurrent system under study is an ACS, so that considering the physical implementation problem is meaningful. In this thesis also, the starting point of the investigation is an ACS rather than an abstract concurrent system. The two main reasons for this choice are:

1) Such a choice does not involve any loss of generality and most of the results (except those concerned with
the realization problem itself) that have been obtained are applicable to any general concurrent system with suitable modifications.

2) The realization problem for ACS's is of considerable and immediate interest and a satisfactory solution to this problem is yet to appear.

Apart from the problem of implementing an ACS, two other questions of importance, the answers to which should be applicable to any concurrent system are:

1) Given the individual descriptions of the subsystems of a network of ACS's, how do we arrive at a behavioral description of the overall network?

2) How do we determine whether two given ACS's are "equivalent" or not?

It is obvious that an algebraic model for an ACS rather than a graphical model would be the appropriate tool for attacking any one of the above-mentioned problems in a systematic manner. Hence the motivation for the two algebraic models proposed in this thesis. The basic idea behind the formulation of these two models is as follows:

An ACS whose behavior is marked-graph-describable is also completely characterized by:
1) The cause-effect relationship that it induces between its input and output signals.

2) The temporal partial ordering that it induces on the signals that flow into and out of the system.

A more detailed discussion of this idea is postponed to Chapter 2.

The plan of the thesis is given below:

In Chapter 2, the ACS system model and the conventions to be followed in its Petri-net description are first stated. (Though a distinct representation scheme exists for marked graphs, the Petri-net representation has been retained for the sake of convenience). Then the first of the two algebraic models called the C-D model is formulated, based on the structure of the Petri-net. The rest of the chapter is devoted to establishing the validity of this model from the point of view of all possible feasible sequences that the ACS together with the external world generates.

The C-D model appears to be a suitable tool for solving the analysis problem for marked-graph-describable ACS's. However, for solving the equivalence problem a more conventional algebraic model relative to which morphisms can be easily defined is required. Hence in Chapter 3, the second model called the poset model is formulated based on the C-D
model and its validity is also established.

Given a precedence graph with suitable interpretations, a speed-independent physical realization for a control structure that implements the given graph can be quite easily obtained. In Chapter 4, the poset model is used to develop a straightforward method for obtaining a realization for marked-graph-describable ACS's, making direct use of the above-mentioned scheme for implementing precedence graphs.

Chapter 5 is devoted to stating the conclusions and the various unsolved problems associated with concurrent systems in general and ACS's in particular.
CHAPTER 2

THE C-D MODEL

2-1

In this chapter, the first of the two algebraic models called the C-D model is formulated and its validity is established. In Section 2-2, the ACS system model and the conventions followed in its Petri-net description are stated. Then the restrictions imposed on the class of ACS's to be studied in this thesis are stated in a precise manner. Finally, the concept of a standard form Petri-net (SFP) is introduced and the rules for converting the given Petri-net into an equivalent SFP are given. In Section 2-3, the C-D model is formulated and the rule for "simulating" it is given. In Section 2-4, its validity is established.

2-2

The ACS under study is assumed to have the form shown in FIG-2.1. It has "n" input links numbered from 1 to n and m output links numbered from n+1 to n+m . A link consists of two wires called the "ready wire" and the "acknowledge wire". The system is permitted to send out "ready" signals on the ready wires of its output links and "acknowledge" signals on the acknowledge wires of its input links to the external world. Conversely, the external world is
permitted to send in ready signals on the input links and acknowledge signals on the output links (on the appropriate wires) into the system.

**Definition**

a) \( I = (\text{Input Signals}) = \{lr, 2r, \ldots, nr, (n+1)a, (n+2)a, \ldots, (n+m)a\} \)

b) \( O = (\text{Output Signals}) = \{(n+1)r, (n+2)r, \ldots, (n+m)r, 1a, 2a, \ldots, na\} \)

c) \( PI = (\text{Primary Input Signals}) = \{lr, 2r, \ldots, nr\} \)

d) \( SI = (\text{Secondary Input Signals}) = \{(n+1)a, (n+2)a, \ldots, (n+m)a\} \)

e) \( PO = (\text{Primary Output Signals}) = \{(n+1)r, (n+2)r, \ldots, (n+m)r\} \)

f) \( SO = (\text{Secondary Output Signals}) = \{1a, 2a, \ldots, na\} \)

g) \( S = (\text{Signals}) = I \cup O ("lr" \rightarrow "Ready" on the link "1", etc.) \)
Definition

Let $x \in S$. If $x$ is of the form "jr" then $x'$ denotes the signal "ja". If $x$ is of the form "ka", then $x'$ denotes the signal "kr".

It is assumed that the specification of the ACS having the structure shown in FIG-2.1 is given in the form of a Petri-net. It is further assumed that the reader is already familiar with Petri-nets*. In this thesis, the following definition of Petri-nets will be used.

Definition

A Petri-net is an ordered quadruple,

$$G = \langle P, T, \cdot, B^0 \rangle,$$

where,

1) $P$ is a finite, non-empty set of places of $G$.
2) $T$ is a finite, non-empty transitions of $G$.
3) $\cdot$ is a binary relation such that, $\cdot \subseteq P \times T \cup T \times P$
4) $B^0$ is the set of places that contain stones initially.

* The uninitiated reader should refer to the writings of Holt [4].
As usual, places will be denoted by circles and transitions by bars in the diagrammatic representations of Petri-nets.

A Petri-net constitutes the description of an ACS only when some suitable interpretations are put on it. One way to do this would be to associate the signals of the ACS with the transitions of the Petri-net in the manner of Suhas Patil [6]. A second way to do this would be to associate the signals of the ACS with certain places of the Petri-net in a 1-1 fashion as Jack Dennis has done [7]. In this thesis, the second method has been chosen, since the author feels that this method would be more appealing to the potential designer. For example, a JUNCTION module (shown in FIG-2.2.a) can be described by the Petri-net shown in FIG-2.2.b with certain implicit rules for simulating it.

Definition

Let $G = (P, T, B^0)$ be a given Petri-net. Let $t \in T$. Then, $I(t)$ denotes the set of all places from which there is a directed arc to $t$. Similarly, $O(t)$ denotes the set of all places to which there is a directed arc from $t$. 
\[ I(t) \rightarrow \text{Input places of the transition } t \.
\]

\[ O(t) \rightarrow \text{Output places of the transition } t \.
\]

"Restriction 1"

It is assumed that the ACS under study is such that, in its Petri-net description,

1) \( \forall p \in P, \ p \in I(t_i) \) and \( \ p \in I(t_j) = t_i = t_j \)

2) \( \forall p \in P, \ p \in O(t_k) \) and \( \ p \in O(t_l) = t_k = t_l \)

A Petri-net that satisfies the two conditions stated in Restriction 1 is termed as a marked graph. A marked graph can be represented by a set of transition nodes and a set of directed arcs [1]. However, in this thesis the Petri-net representation has been retained for ease of interpretation. As mentioned earlier, Holt has carried out an exhaustive analysis on the properties of marked graphs. The results of the above work have been made use of wherever possible, though very rarely in its original form. (A summary of some of the elementary properties of marked graphs may be found in Appendix-A.)

With these preliminaries taken care of and with the understanding that the Petri-net description of the given
ACS will be of the form shown in FIG-2.2.b and will satisfy the condition stated in Restriction 1, some additional terminology can now be introduced.

**Definition**

\[ \forall p \in P, \ l(p) = \text{The label carried by the place } p. \]

If \( p \) does not contain any labels (i.e., \( p \) is an internal place) then \( l(p) = \varnothing \).

**Definition**

\[ \forall x \in S, \]

\[ p(x) = p \in P \text{ such that } l(p) = x. \]

**Definition**

\[ LP = \{ p \in P \mid l(p) \neq \varnothing \}. \]

\( (LP \rightarrow \text{labelled places.}) \)

To the reader who is familiar with Petri-nets, it will be apparent that the "simulation" of a net of the form shown in FIG-2.b does not depend solely on the stone distribution alone. Apart from the conventional rule for firing transitions, certain implicit rules for transferring stones from place \( p(3r) \) to \( p(3a) \) and \( p(1a) \) to \( p(1r) \), etc. will have to be followed. (To the uninitiated reader,
these terms and considerations will become clear after reading through the beginning part of Section 2-4, where simulation has been precisely defined.) The standard form Petri-net (SFP) will be introduced to remove such implicit rules and make the simulation of the net depend on the stone distribution alone. Yet another function of the SFP is to aid in the formulation of the C-D model and the proof of its validity. Any given Petri-net (describing an ACS) of the form shown in FIG-2.2.b is transformed into the corresponding SFP by repeatedly applying the following three rules wherever they are applicable. Before stating the three rules, we observe the following:

Because of the conventions that have been assumed about the Petri-net description of the ACS,

$$\forall x \in I, \forall t \in T \text{ such that } p(x) \in O(t).$$

(We say that $p(x)$ is isolated from the "output side".) Furthermore,

$$\forall x \in O, \forall t \in T \text{ such that } p(x) \in I(t).$$

(We say that $p(x)$ is isolated from the "input side".)
**Rule 1**

Let \( x \in \text{PI} \). Consider \( p(x) \) which will be isolated from the output side. Also \( x \in \text{PI} \Rightarrow x' \in \text{SO} \). Hence \( p(x') \) will be isolated from the input side. Introduce two new transitions and one new internal place that contains a stone initially as shown in FIG-2.3.a. \( t_1 \) and \( t_2 \) are the new transitions and \( p \) is the new place that have been introduced.

**Rule 2**

Let \( x \in \text{SI} \). Consider \( p(x) \). It will be isolated from the output side. Also, \( x \in \text{SI} \Rightarrow x' \in \text{PO} \). Hence \( p(x') \) will be isolated from the input side. Introduce a single new transition as shown in FIG-2.3.b. \( t_3 \) is the new transition that has been introduced.

**Rule 3**

Let \( t \in T \) such that,

1) \(|O(t)| > 1 \quad [|O(t)| \rightarrow \text{cardinality of the set } O(t)]\)

and

2) \( \exists p \in O(t) \) such that \( p \in \text{LP} \)

Then introduce one new internal place and one new transition as shown in FIG-2.3.c.
p is the new place and \( t_4 \) is the new transition that has been introduced. The SFP representation of the JUNCTION module derived from FIG-2.2.b is shown in FIG-2.4. The places and transitions marked with "✓" have been introduced by applying Rule 1, those carrying "x" marks through Rule 2 and the others marked with "Δ" through applying Rule 3.

For the rest of the thesis, it will be assumed that the Petri-net describing the behavior of the given ACS has been transformed into its corresponding SFP. The fact that the SFP is equivalent up to simulation to the original Petri-net is obvious.

The second major restriction imposed on the class of ACS's can now be stated. For doing this, let \( M \) be the given ACS, and SFP (M) denote its SFP specification. A marked graph is associated with \( M \) and is constructed from SFP (M) as follows:

Algorithm 1 - Step 1

Corresponding to each transition \( t_i \) define a transition node \( v_i \).

Step 2

Let \( p \in P \). Let \( t_i \in T \) such that \( p \in O(t_i) \) and \( t_j \in T \) such that \( p \in I(t_j) \). Introduce an arc that is
FIGURE-2.4
directed from \( t_i \) to \( t_j \). If (and only if) \( p \in B^0 \) put a marker on this arc.

**Step 3**

Do Step 2 for \( \forall p \in P \). The marked graph arrived at by using Algorithm 1 is denoted by \( mg(M) \).

**Restriction 2**

The ACS, say \( M \), under study is assumed to be such that \( mg(M) \) is live and safe.

\( mg \) (JUNCTION module) is shown in FIG-2.5.

2-3

As mentioned earlier, the ACS is completely characterized by,

1) The cause-effect relationship that it (together with the external world) imposes between the signals that flow into and out of the system.

2) The temporal (partial) ordering imposed on the flow of signals.

Consider the JUNCTION module whose SFP description is shown in FIG-2.4. In order to generate a "3r" the module must first receive a "1r" and a "2r", i.e., 1r and 2r in
some sense "cause" 3r to be generated. Similarly, from the point of view of the external world, in order to generate a 3a it must first receive a 3r. Thus, corresponding to each \( x \in S \), we can intuitively associate a time-invariant set of signals [which will be denoted as \( C(x) \)] that cause \( x \) to be generated.

Apart from this cause-effect relationship, the JUNCTION module also ensures that the \( n^{th} \) generation of 3r takes place iff the \( n^{th} \) reception of 1r and 2r has occurred \((n \geq 1)\). Similarly, the external world ensures (more properly, is assumed to ensure) that the \( n + 1^{th} \) generation of 2r occurs iff the \( n^{th} \) reception of 2a has occurred \((n \geq 0)\). In other words, the ACS-External world combination ensures that a constant minimal difference is maintained between \( z_i \) and \( x \) for every \( z_i \) contained in the set \( C(x) \), where \( x \) is any element of \( S \). This is the source of the temporal ordering imposed on the flow of signals.

In view of the above discussion it seems reasonable to expect that a valid alternative representation for the ACS would be to,

1) List the set \( C(x) \) for each \( x \) in the Set \( S \).

2) For each \( z \in C(x) \), list the constant minimal difference that is maintained between \( z \) and \( x \) [denoted by \( D(z,x) \)], doing this for every \( x \) in the Set \( S \).
This is the basic idea behind the C-D model formulation.

The definitions that follow (which constitute the formulation of the C-D model and the rule for "simulating" it) have been constructed, with reference to a given ACS "M", and SFP (M).

Definition

\[ \forall p_i, p_j \in P, \]

\[ (p_i, p_j) \in R_{ad} \Rightarrow (\exists t \in T) \text{ such that } 1) p_i \in I(t) \]

\[ \text{and} \]

\[ 2) p_j \in O(t) \]

Definition

A finite sequence of places \( p_1 p_2 \ldots p_n \) is a path from \( p_1 \) to \( p_n \) if,

\[ (p_i, p_{i+1}) \in R_{ad} \quad (i = 1, 2, \ldots, n-1) \]

Definition

Let \( x \in S \). Then,

\[ Cl(x) = \{ y \in S \mid \exists \text{ a path from } p(y) \text{ to } p(x) \text{ that does not contain any other labelled places except for } p(x) \text{ and } p(y) \} \]
For example, for JUNCTION module whose SFP description is given in FIG-2.4,

\[ \text{Cl}(1r) = \{1a\}; \text{Cl}(2r) = \{2a\}; \]
\[ \text{Cl}(3r) = \{1r, 2r, 3a\}; \text{Cl}(3a) = \{3r\}. \]

**Definition** \( \forall p \in P \)

\( p \) is a maxima1 place if \( p \in B^0 \).

**Remark**

One of the conventions followed in the specification of an ACS by a Petri-net is that, \( p \in R^0 = p \notin I.P \).

**Definition**

a) Let \( x \in S \) and \( y \in \text{Cl}(x) \). Then,

\[ \text{PTH}(y|x) = \begin{cases} \text{The Set of all paths from} \\ p(y) \text{ to } p(x) \text{ that do not contain} \\ \text{any other labelled places except} \\ \text{for } p(x) \text{ and } p(y) \end{cases} \]

b) An element of the set \( \text{PTH}(y|x) \) is said to be a minimal path if the number of maximal places that it contains is not greater than the number of maximal places contained in any other element in the set \( \text{PTH}(y|x) \).
c) \( N(y|x) \) denotes the number of maximal places contained in a minimal path of the set \( \text{PTH}(y|x) \).

Remark

In general \( \text{PTH}(y|x) \) can have more than one minimal path. However, since the set of integers is totally ordered, \( N(y|x) \) is unique.

Definition

Let \( x \in S \) and \( y, z \in \text{Cl}(x) \). Then,

\[(y,z) \in R_x \quad \leftrightarrow \quad \exists \text{ a path } \overline{p} \text{ of the form}
\]

\[p_1p_2 \cdots p_i p_{i+1} \cdots p_j \text{ such that,}
\]

1) \( p_1 = p(z) \) and

2) \( p_j = p(x) \) and

3) \( p_i = p(y) \) and

4) The number of maximal places contained in \( \overline{p} \leq N(z|x) \).

Thus in the case of the JUNCTION module, we have,

\[(1r, 3a), (2r, 3a) \in R_{3r} \]
**Definition**

Let \( x \in S \). Then,

\[
C(x) = \left\{ y \in Cl(x) \mid \nexists z \in Cl(x) \ (z \neq y) \right. \\
\text{such that,} \ (z,y) \in R_x \right\}
\]

**Proposition 2.1**

\( \forall x \in S, \ R_x \) is anti-symmetric.

**Proof** \ (By contradiction) \n
Let \( x \in S \) and \( y,z \in Cl(x) \) and furthermore, let \( N(y|x) = i \) and \( N(z|x) = j \). Assume \( (y,z) \in R_x \) and \( (z,y) \in R_x \) and \( y \neq z \). Then paths of the form shown in FIG-2.6 must exist. The rectangular boxes indicate the number of maximal places that are encountered in the corresponding paths.

The following equalities and inequalities hold by definitions.

\[
k_1 + k_2 \leq i \leq k_3 \quad \ldots \quad (1)
\]
\[
k_3 + k_4 \leq j \leq k_2 \quad \ldots \quad (2)
\]
From (1) and (2) we have, \( k_1 + k_4 = 0 \). But this implies that \( mg(M) \) contains a blank circuit and hence is not live, which is a contradiction and hence the proposition is proved.

**Remark**

It is easy to see (because of Restriction 2 and its implications) \( \forall x \in S, Cl(x) \neq \emptyset \). By proposition 2.1, we are assured that \( \forall x \in S, C(x) \neq \emptyset \) also, since \( Cl(x) \neq \emptyset \).

**Remark**

It is easy to show that \( \forall x \in S, R_x \) is also reflective and transitive. Since these properties of \( R_x \) are not needed in the subsequent discussions, a proof of this assertion has been omitted.

Continuing with the example of JUNCTION module,

\[
C(1r) = \{la\} ; \quad C(2r) = \{2a\} ; \quad C(3r) = \{lr,2r\} ; \\
C(3a) = \{3r\} ; \quad C(1a) = C(2a) = \{3a\} .
\]
**Definition**

Let $y, x \in S$. Then,

$$D(y, x) = \begin{cases} 
1 - N(y|x), & \text{if } y \in C(x) \\
\text{undefined}, & \text{otherwise}
\end{cases}$$

For the JUNCTION module,

$$D(1a, 1r) = D(2a, 2r) = 1 - 1 = 0$$
$$D(1r, 3r) = D(2r, 3r) = 1 - 0 = 1$$
$$D(3r, 3a) = 1 - 0 = 1$$
$$D(3a, 1a) = D(3a, 2a) = 1 - 0 = 1$$

**Remark**

From the definitions and the SFP construction it is obvious that

a) $x \in PI \Rightarrow C(x) = \{x'\}$ and $D(x', x) = 0$

b) $x \in SI \Rightarrow C(x) = \{x'\}$ and $D(x', x) = 1$.

The C-D model description of ACS will thus consist of,

1) Specification of the set $S$ (signals)

2) Specification of the set $C(x)$, $\forall x \in S$

3) Specification of $D(y, x)$ whenever it is defined.
However, any model for an ACS -- be it graphical or algebraic -- is incomplete and in fact meaningless, unless the model is accompanied by the rules for simulating it. Indeed it is only through simulation does any static model (which indeed graphical and algebraic models are in a trivial way) represent the behavior of a dynamic system, which the ACS system is. The following additional terminology will be introduced to aid in the rigorous specification of the rules for simulating the C-D model.

**Definition** (By induction)

**Basis Step**

\[ S_1 = S \]

**Induction Step**

\[ S_{i+1} = \{ xy | x \in S_i \text{ and } y \in S \} \quad (i \geq 1) \]

**Definition**

a) \[ S_+ = S_1 \cup S_2 \cup \ldots \]

b) \[ S_0 = \{ \lambda \} \quad (\"\lambda\" \text{ denotes the null sequence}) \]

c) \[ S^*_+ = S_0 \cup S_+ \]

**Remark**

The more conventional notation of \( S_+ \) and \( S^*_+ \) have been reserved for a latter use in Chapter 3.
Definition

Let \( x \in S_* \) and let \( y \in S \). Then,

\[
\#y(x) = \text{The number of times the symbol "y" appears in the sequence } x.
\]

Definition

Let \( x \in S_* \). Then,

\[
\text{En}(x) = \left\{ y \in S \mid \forall z \in C(y), \#z(x) - \#y(x) \geq D(z,y) \right\}
\]

(En \( \rightarrow \) enabled)

Remark

As usual for \( x \in S_* \), \( \log(x) \) will denote the number of non-null symbols contained in \( x \). (\( \log \rightarrow \) length)

The simulation of the C-D model consists of generating feasible signal sequences using the rule given in the following definition. As might be expected, the definition is inductive.

Definition (of feasible sequences)

Basis Step \( \lambda \) is a feasible sequence.
Induction Step

Let $\bar{x}$ be a feasible sequence such that $lg(\bar{x}) = i (i \geq 0)$. Then $\bar{xy}$ is a feasible sequence iff $y \in En(\bar{x})$.

Notation

The set of all possible feasible sequences that are generated by using the above definition is denoted by $FS^\prime$. (FS = Feasible Sequences.)

For the JUNCTION module, one possible feasible sequence using the above rule for simulation is,

$$\bar{x} = "lr 2r 3r 3a lr 2a 2r 3r 3a 2a 2r 1a 1r"$$

It is easy to see that,

1) $C(3r) = \{lr, 2r\}$;

2) $D(lr, 3r) = D(2r, 3r) = 1$;

3) $#lr(\bar{x}) = #2r(\bar{x}) = 3$;

4) $#3r(\bar{x}) = 2$.

Hence $3r \in En(\bar{x})$.

The formulation of the C-D model is now complete. It is left to show that it is a valid model and this is done in the next section.
Briefly stated, the simulation of an arbitrary
Petri-net consists of a sequence (finite or infinite) of
transition firings following certain rules. The rules are
(assuming a given initial stone distribution),
1) A transition can be fired if each one of its input
places contains at least one stone.
2) The firing of a transition consists of removing one
stone from each one of its input places and adding
one stone to each one of its output places.

Alternatively, a simulation may also be viewed as a
sequence of stone distributions. In order to make this
notion (relative to the class of Petri-nets considered in
this thesis) precise, the following definition is needed.

Definition
Let \( B^k \) be any arbitrary stone distribution in a
given Petri-net. Then
\[
\text{Et} (B^k) = \{ t \in T \mid I(t) \subseteq B^k \}
\]

(Et \( \rightarrow \) Enabled transition)

Now for the Petri-nets that occur as the SFP repre-
sentation of ACS's that satisfy Restrictions 1 and 2, a
simulation may be defined as follows:
Definition

Let the initial stone distribution be $B^0$.

Basis Step $B^0$ is a simulation.

Induction Step

Let $B^0 B^1 \ldots B^k$ be a simulation ($k \geq 0$). Then, $B^0 B^1 \ldots B^k B^{k+1}$ is a simulation if,

$$\exists t \in \text{Et}(B^k) \text{ such that } B^{k+1} = [B^k U(t)] - I(t)$$

We will frequently denote the simulation $B^0 B^1 \ldots B^k$ by $(B^0 \rightarrow B^k)$ when the intermediate stone distributions are not important.

With this brief introduction to the concept of simulation, the rationale behind the proof of the validity of the C-D module can now be stated as follows:

A stretch of behavior of the ACS is characterized by the sequence of signals that is generated by the ACS-External world combination during that stretch. Since an occurrence system (which the ACS under study is) is completely characterized by the set of all possible stretches of behavior that it can exhibit, yet another way to characterize the behavior of an ACS would be to list the set of all possible
signal sequences that it, together with the external world, can generate.

If the SFP description of an ACS is to be a faithful representation (which it will be, if the original Petri-net representation was), then corresponding to each stretch of behavior of the ACS, there must be a simulation of the SFP. Since, as already stated, a signal sequence is associated with each stretch of behavior, it must be possible to associate a particular signal sequence with each simulation of the SFP. Moreover, faithfulness of representation demands that the set of signal sequences obtained by associating one sequence with each simulation, for all possible simulations of the SFP must precisely be the set of signal sequences that characterize the behavior of the given ACS. This set of signal sequences is denoted by FS. We already have a set of signal sequences obtained through all possible simulations of the C-D model (indeed, for the C-D model, a "simulation" consists of generating a signal sequence!) which has been denoted by FS'. Then establishing the fact that FS = FS' would constitute a proof of the validity (faithfulness) of the C-D model.

However FS first has to be defined. Unfortunately the sequences obtained by simulating the SFP contain symbols from the set T also, due to the presence of "internal" transitions. However, denoting the set of sequences thus
obtained, by $FS''$, and by defining a simple "refinement" operation on the elements of this set, the required set $FS$ can be quite easily obtained. In the following definition, $B^0$ denotes the initial stone distribution of the SFP of the given ACS. Furthermore, the set $(S \cup T)_*$ denotes the set defined in the same way that the set $S_*$ was defined in Section 2-3.

**Definition (of $FS''$)**

**Basis Step**

$\lambda$ is a feasible sequence and $B^0$ is the consequent stone distribution.

**Induction Step**

Let $x \in (S \cup T)_*$ be a feasible sequence such that $\ell(x) = k$ ($k \geq 0$) and the consequent stone distribution be $B^x$. Then, if $t \in E_t (B^x)$ then $\{\text{if } [O(t) \cap LP = \emptyset]\text{ then } (xt \text{ is a feasible sequence})\}$. Else, $[\overline{x} (\ell(0(t)))$ is a feasible sequence $\}$ and the consequent stone distribution is $B^{xy}$, where,

$$B^{xy} = [B^x \cup O(t)] - I(t).$$
Remark

\[ O(t) \cap LP \neq \emptyset \quad \Rightarrow \quad \text{An output place of } t \text{ is a labelled place} \]

\[ \Rightarrow \quad \text{This is the only output place of } t \text{ by the SFP construction} \]

\[ \Rightarrow \quad \ell[O(t)] \text{ is well defined.} \]

The "refinement" operation that produces the required set \( FS \) is defined as follows.

Definition

Let \( \overline{x} \in FS'' \). Then,

\[
\text{Ref}(\overline{x}) = \begin{cases} 
\text{The sequence obtained by deleting all symbols in } \overline{x} \text{ which are contained in the set } T \text{ and preserving the order of the remaining symbols in } \overline{x}. 
\end{cases}
\]

Definition \hspace{1cm} (of \( FS \))

\[ FS = \{ \text{Ref}(\overline{x}) \mid \overline{x} \in FS'' \} \]

Remark

Because of the way the consequent stone distribution has been defined, it is obvious that \( FS \) is indeed the set of signal sequences obtained through all possible simulations.
of the SFP, starting from $B^0$.

The following two definitions introduce some additional terminology that will aid in showing that $FS = FS'$.

**Definition** (By Induction)

**Basis Step**

Let $B^0$ be a simulation and $p \in P$. Then,

$$
\#p(B^0) = \begin{cases} 
1, & \text{if } p \in B^0 \\
0, & \text{otherwise} 
\end{cases}
$$

**Induction Step**

Let $(B^0 \to B^k) = B^0 B^1 \ldots B^k$ be a simulation such that $\#p(B^0 \to B^k) = \ell$ and let $B^0 B^1 \ldots B^k B^{k+1}$ be also a simulation. Then,

$$
\#p(B^0 \to B^{k+1}) = \begin{cases} 
\ell + 1, & \text{if } p \notin B^k \text{ and } p \in B^{k+1} \\
\ell, & \text{otherwise} 
\end{cases}
$$

**Remark**

$\#p(B^0 \to B^k)$ denotes the number of times the place $p$ has received a stone in the simulation $(B^0 \to B^k)$.

We will frequently write $\#p$ when the associated simulation is clear from the context.
Definition (By Induction)

Basis Step

Let \( B^0 \) be a simulation and \( t \in T \). Then,
\[
\#t(B^0) = 0.
\]

Induction Step

Let \((B^0 \rightarrow B^k) = B^0 B^1 \cdots B^k\) be a simulation \((k \geq 0)\), \(\#t(B^0 \rightarrow B^k) = \ell\) and \(B^0 B^1 \cdots B^k B^{k+1}\) be a simulation also. Then,
\[
\#t(B^0 \rightarrow B^{k+1}) = \begin{cases} 
\ell+1, & \text{if } B^{k+1} = [B^k \cup O(t)] - I(t) \\
\ell, & \text{otherwise.}
\end{cases}
\]

Remark

\(\#t(B^0 \rightarrow B^k)\) denotes the number of times the transition \(t\) has been fired in the simulation \((B^0 \rightarrow B^k)\).

We will frequently write \(\#t\) when the associated simulation is clear from the context.

The rest of this section will be devoted to showing that \(FS = FS'\).
Lemma 2-1

Let \( B^0 \rightarrow B^k \) be a simulation. Also, let \( p \in P \) and \( t \in T \) such that \( p \in I(t) \). Then,

\[
\#p = \begin{cases} 
(#t) + 1, & \text{if, } p \in B^k \\
#t, & \text{otherwise.}
\end{cases}
\]

Proof

Follows directly from the rules for firing a transition and the properties of the given SFP.

Lemma 2-2

Let \((B^0 \rightarrow B^k)\) be a simulation. Also, let \( p \in P \) and \( t \in T \) such that \( p \in O(t) \). Then,

\[
\#p = \begin{cases} 
(#t) + 1, & \text{if, } p \in B^0 \\
#t, & \text{otherwise}.
\end{cases}
\]

Proof

Follows directly from the rules for firing a transition and the properties of the given SFP.

Lemma 2-3

Let \( \bar{x}'' \in FS'' \) and \( B^k \) be the consequent stone distribution. Then,
\[\forall y \in S, \]
\[\#_y(\overline{x''}) = \#_y[\text{Ref}(\overline{x''})] = \#_p(y)(B^0 \rightarrow B^k).\]

**Proof**

Follows directly from the definitions.

The validity of the C-D model will be established by showing that \(\text{FS}' \subseteq \text{FS}\) and \(\text{FS} \subseteq \text{FS}'\). This task becomes relatively simple when the results of two additional lemmas become available. Some additional terminology is required for stating these two lemmas in a concise manner.

**Definition**

A finite sequence of places, \(p_1p_2\ldots p_k\) is said to be a backward-directed-path from \(p_1\) to \(p_k\), if,

\[(p_{i+1}, p_i) \in R_{\text{ad}}, \quad i = 1, 2, \ldots, k-1\]

Furthermore, such a path is said to be of length \(j-1\) if \(\ell g(p_1p_2\ldots p_k) = j\).

**Remark**

In the subsequent discussions, "backward-directed-path" will be abbreviated as "b.d.p". Furthermore the b.d.p(!) from \(p_1\) to \(p_k\) will be denoted by \(p_1p_2\ldots p_k\) instead of the more conventional but also more cumbersome notation viz.
when it is clear from the context that the sequence \( p_1 p_2 \ldots p_k \) is a b.d.p from \( p_1 \) to \( p_k \).

**Definition**

\( N(p_1 p_2 \ldots p_k) \) is the number of maximal places encountered in the b.d.p from \( p_1 \) to \( p_k \), *excluding the place* \( p_k \).

**Remark**

We will simply write \( N \) when the associated b.d.p is clear from the context.

**Definition**

\( ST(p_1 p_2 \ldots p_k) \mid B^\ell \) is the number of places containing stones, *excluding the place* \( p_1 \) in the b.d.p \( p_1 p_2 \ldots p_k \), corresponding to a given stone distribution \( B^\ell \).

**Remark**

We will frequently write \( ST \), when both the associated b.d.p and the stone distribution are clear from the context.

**Lemma 2-4**

Let,

a) \( (B^0 \rightarrow B^\ell) \) be a simulation.

b) \( p_1 \in P \) such that \( p_1 \notin B^0 \).

c) \( p_1 p_2 \ldots p_k \) be a *circuit-free* b.d.p from \( p_1 \) to \( p_k \).
Then,

\[ \#p_1 = \#p_k + N - ST. \]

**Proof**

By induction on the length of the b.d.p "p_1p_2...p_k".

**Basis Step**

Let the length of the b.d.p p_1p_2...p_k = 1.

**Case 1**  
\( p_2 \in B^k \) (w.r.t FIG-2.7.a)

1) \( N = 0 \) (by definition)
2) \( ST = 1 \) (by definition)

But, \( \#p_1 = \#t (p_1 \l_1 B^0 ; \text{Lemma 2-2}) \)

\[ = \#p_2 - 1 \ (p_2 \in B^k ; \text{Lemma 2-1}) \]

Thus the lemma holds for this case.

**Case 2**  
\( p_2 \notin B^k \)

Using the same argument that was used for the previous case, we can get,

1) \( N = 0 ; \)
2) \( ST = 0 ; \)
3) \( \#p_1 = \#p_2 \)

Thus the basis step is proved.
FIGURE 2.7.a

FIGURE 2.7b
**Induction Step**

Assume the lemma to be true for all circuit-free b.d.p's of length \( j \) (\( j \geq 1 \)). Let \( P_1P_2\ldots P_{j+1}P_{j+2} \) be a circuit-free b.d.p of length \( j+1 \). Then w.r.t FIG-2.7.b,

**Case 1** \( p_{j+1} \in B^0 \) and \( p_{j+2} \in B^\ell \)

By the induction hypothesis,

\[
\#p_1 = \#p_{j+1} + N(P_1P_2\ldots P_{j+1}) - ST(P_1P_2\ldots P_{j+1}) \quad \text{(X1)}
\]

For the particular case considered,

\[
N(P_1P_2\ldots P_{j+1}) = N(P_1P_2\ldots P_{j+1}P_{j+2}) - 1 \quad \text{(X2)}
\]

\[
ST(P_1P_2\ldots P_{j+1}) = ST(P_1P_2\ldots P_{j+1}P_{j+2}) - 1 \quad \text{(X3)}
\]

Also,

\[
\#p_{j+1} = t_{j+1} + 1 \quad (p_{j+1} \in B^0 \text{ ; Lemma 2-2})
\]

\[
= \#p_{j+2} \quad (p_{j+2} \in B^\ell \text{ ; Lemma 2-1})
\]

i.e., \( \#p_{j+1} = \#p_{j+2} \quad \text{..........................(X4)} \)

Substituting (X2), (X3) and (X4) in (X1), we get,

\[
\#p_1 = \#p_{j+2} + N(P_1P_2\ldots P_{j+2}) - 1 - ST(P_1P_2\ldots P_{j+2}) + 1.
\]
Thus the induction step is proved for this case. The same type of argument may be used for proving the induction step for the three remaining cases also, which are,

**Case 2** \( p_{j+1} \in B^0 \) and \( p_{j+2} \notin B^l \).

**Case 3** \( p_{j+1} \notin B^0 \) and \( p_{j+2} \in B^l \).

**Case 4** \( p_{j+1} \notin B^0 \) and \( p_{j+2} \notin B^l \).

Thus the lemma is proved.

The rather detailed nature of the following lemma is due to the fact that it will be used for proving a part of the lemma that establishes the fact that \( FS' \subseteq FS \).

**Lemma 2-5**

1) Let \( B^0 \rightarrow B^k \) be a simulation and \( \overline{x''} \) be the element of \( FS'' \) produced by this simulation.

2) Let \( y \in S \) and \( C(y) = \{z_1, z_2, \ldots, z_\ell\} \).

3) Corresponding to the above given simulation, let \( \#p(z_i) - \#p(y) \geq D(z_i, y), \ i = 1, 2, \ldots, \ell \).

4) Let \( p(y)p_2 \ldots p_j \) be any b.d.p from \( p(y) \) to \( p_j \) such that w.r.t this b.d.p and \( B^k \), \( ST = 0 \).
Then,

a) There are only a finite number of such b.d.p's starting from p(y), each of which is of finite length (and hence circuit-free).

b) No such path contains a labelled place except for p(y).

Proof

Part a

Follows immediately from the fact that the SFP under study is finite and live.

Part b (By contradiction)

Supposing there exists such a b.d.p from p(y) that contains a labelled place, p_j, i.e., p(y)p_2...p_j is a b.d.p such that ST = 0 and p_j \in LP. Without any loss of generality assume that the b.d.p under discussion does not contain any other labelled places except for p(y) and p_j.

Case 1 \( \ell(p_j) \in C(y) \)

Let \( \ell(p_j) = z_i \ (1 \leq i \leq \ell) \). Then we have

\[ ST = 0 ; \quad N \geq N(z_i \ | y) . \]
Hence by lemma 2-4,

\[ #p(y) \geq #p(z_i) + N(z_i|y) \]

Hence,

\[ #p(y) - #p(z_i) < 1 - N(z_i|y) , \]

a contradiction, since \( 1 - N(z_i|y) = D(z_i, y) \), by definition.

\[ \text{Case 2} \quad \ell(p_j) \notin C(y) \]

Let \( \ell(p_j) = w \). Furthermore denote the b.d.p from \( p(y) \) to \( p(w) \) under discussion by \( p^{th}1 \). By definition, the existence of \( p^{th}1 \) ensures that, \( w \in Cl(y) \).

\( w \in Cl(y) \) and \( w \notin C(y) \) \( \Rightarrow \) \( \exists z_i \in C(y) \) and a b.d.p (denoted by \( p^{th}2 \)) from \( p(y) \) to \( p(w) \) passing through \( p(z_i) \) such that, \( N(p^{th}2) \leq N(w|y) \)

Consider FIG-2.8.

Since part b of this lemma has been already proved for Case 1 (the following argument will make it clear that considering first the case where \( \ell(p_j) \in C(y) \) involves no loss of generality), the portion of the b.d.p "\( p^{th}2 \)" from \( p(y) \) to \( p(z_i) \) must have at least one place that contains a stone. Hence, \( ST(p^{th}2) = k' \geq 1 \). Also, \( N(p^{th}2) \leq N(w|y) \).
Thus w.r.t. \( p_{th2} \), using lemma 2-4,

\[ \#p(y) \leq \#p(w) + N(w|y) - k' \quad \text{(Y1)} \]

Now, considering \( p_{th1} \),

\[ ST(p_{th1}) = 0 ; N(p_{th1}) \geq N(w|y) \]. Hence,

w.r.t \( p_{th1} \), by lemma 2-4,

\[ \#p(y) \geq \#p(w) + N(w|y) \quad \text{(Y2)} \]

(Y1) and (Y2) \( \Rightarrow 0 \leq -k' \)

\[ \Rightarrow 0 \geq k' \), a contradiction since \( k' \geq 1 \).

Thus part b of the lemma is also proved.

**Lemma 2-6**  \( FS' \subseteq FS \)

**Proof**  By induction on \( \lg(\bar{x}) \).

**Basis Step**

Let \( \bar{x} \in FS' \) and \( \lg(\bar{x}) = 0 \). Then the lemma is obviously true.
Induction Step

Assume the lemma to be true \( \forall x \in FS' \) such that \( \#g(x) = i \) \((i \geq 0)\). Let \( xy \in FS' \) such that \( \#g(xy) = i + 1 \).

\[ xy \in FS' \Rightarrow y \in En(x) \]

\[ = \forall z \in C(y), \#z - \#y \geq D(z, y) \]

By induction hypothesis,

\[ x \in FS' \Rightarrow \exists x'' \in FS'' \text{ such that } \text{Ref}(x'') = x. \]

Let the simulation that produces \( x'' \) be denoted by \((B^0 \rightarrow B^k)\) and the consequent stone distribution by \( B^k \).

By lemma 2-3, w.r.t the above simulation, we have, by the consequences of the induction hypothesis,

\[ \forall z \in C(y), \#p(z) - \#p(y) \geq D(z, y) \]

Now, consider the sub-Petri-net obtained by taking the union of all b.d.p's starting from \( p(y) \) for each of which (corresponding to \( B^k \)) \( ST = 0 \).* Denote this net by \( G_y \). Then, by part a of lemma 2-5, in \( G_y \), there must be a b.d.p of maximal length starting from \( p(y) \). Hence at

* This proof is based directly on the proof developed by Holt for proving Theorem E4 in [1].
least one transition in $G_Y$ must be firable. Let $t_j$ be such a transition.

Supposing we fire now $t_j$ and denote this extended simulation by $(B^0 \rightarrow B^{k+1})$. Then because by part b of lemma 2-5, $p_j \nsubseteq \text{LP}$, by the SFP construction $O(t_j) \cap \text{LP} = \emptyset$. So that the sequence generated by the simulation $(B^0 \rightarrow B^{k+1})$ is $x''t_j$ and the consequent stone distribution $B^{k+1}$ is given by

$$B^{k+1} = [B^k \cup O(t_j)] - I(t_j).$$

Suppose a new sub-Petri-net $G_{Y_1}$ is now formed w.r.t $B^{k+1}$ just in the same way that $G_Y$ was formed. It is easy to see that $G_{Y_1} \subset G_Y$. Then, by applying the same argument to $G_{Y_1}$ (that was applied to $G_Y$), yet another transition, say $t_{j-1}$ can be fired so that $x''t_jt_{j-1} \in \text{FS}''$. Continuing this process, we will eventually have,

1) $(B^0 \rightarrow B^{k+j})$ as a simulation.

2) $x''t_jt_{j-1}...t_1 \in \text{FS}''$ as the sequence generated by the above simulation and $B^{k+j}$ as the consequent stone distribution.

3) $t_0 \in \text{Et}(B^{k+j})$

So that $x''t_jt_{j-1}...t_1 \in \text{FS}''$. 
But this implies that $xy \in FS$ and hence the induction step is proved.

**Lemma 2-7** \( FS \subseteq FS' \)

**Proof** By induction on \( \lg(x) \).

**Basis Step**

Let \( x \in FS \) such that \( \lg(x) = 0 \). Then the lemma is obviously true.

**Induction Step**

Assume the lemma to be true \( \forall x \in FS \) such that \( \lg(x) = i(i \geq 0) \).

Let \( xy \in FS \) such that \( \lg(xy) = i + 1 \)

\[ xy \in FS \Rightarrow \exists x'y \in FS' \] such that \( \text{Ref}(x'y) = x \)

Let \( (B^0 \rightarrow B^k) \) be the simulation that produces \( x'' \), the consequent stone distribution being \( B^k \). Consider FIG-2.9. Then,

\[ x''y \in FS' \Rightarrow ty \in Et(B^k) \].

Also, by the induction hypothesis,

\[ x \in FS \Rightarrow x \in FS' \].
Furthermore, by lemma 2-3,

\[ \forall w \in S, \; \#w \in S \]  =  \[ \#p(w) \in B^0 \implies B^k \] (X)

Let \( z \in C(y) \). Consider a minimal path (in the sense of the definition contained in section 2-3) from \( p(z) \) to \( p(y) \) as shown in FIG-2.9. Denote the corresponding b.d.p from \( p(y) \) to \( p(z) \) by \( p^t \). Then, w.r.t \( p^t \) and \( B^k \),

\[
N = N(z|y) \quad \text{(by definition)}
\]

\[
ST \geq 1 \quad \text{[since } t \in B^k \text{]}.
\]

Hence, by lemma 2-4,

\[
\#p(y) \leq \#p(z) + N(z|y) - 1
\]

i.e., \( \#p(z) - \#p(y) \geq 1 - N(z|y) = D(z,y) \)

i.e., \( \#z(x) - \#y(x) \geq D(z,y) \text{ [by (x)]} \)

Since \( z \) was taken to be any element of \( C(y) \), we can conclude,

\[
y \in \text{En } (x).
\]

Thus \( xy \in FS' \) and the induction step is also proved.

*Theorem 2-1* \( FS = FS' \)
Proof

Follows at once from lemma 2-6 and lemma 2-7.

Thus, the validity of the C-D model is now established.
In this chapter the second algebraic model called the poset model is formulated and its validity is established. As the name suggests the poset model consists of representing the given ACS (that satisfies the two restrictions stated in Chapter 2) as a set (an infinite set) and a single partial ordering relation imposed on the elements of this set. In Section 3-2, the poset model is formulated and in Section 3-3, its validity is established.

One way to interpret the C-D model of an ACS would be as follows: the ACS is completely characterized by the temporal ordering that it imposes (together with the external world) on the flow of signals. For $x \in S$, $C(x)$ is the set of "immediate predecessors" of the signal "x". Hence an alternative model that captures all the aspects the temporal ordering imposed by the ACS should also be a valid model. The poset model that will be formulated below indeed carries out this function in a valid manner. The starting point for the formulation of this model will be the C-D model specification of the ACS.
Definition

a) \( S^i = \{ x^i \mid x \in S \} \quad i = 1, 2, \ldots \)

The element \( x^1 \in S^1 \) will be frequently denoted by \( x \).

b) \( S^0 = \{ \lambda \} \) where "\( \lambda \)" as usual is the null symbol.

Definition

a) \( S^+ = S^1 \cup S^2 \cup \ldots \)

b) \( S^* = S^0 \cup S^+ \)

\( S^* \) is the infinite set that was mentioned in Section 3-1. Let \( x \in I \). Then \( x^i \) will be interpreted as the \( i \)th reception of the signal "\( x \)" to the ACS. \( (i \geq 1) \). Similarly, if \( y \in O \), then, \( y^j \) will be interpreted as the \( j \)th generation of the signal "\( y \)" by the ACS.

The partial ordering relation imposed on the elements of the set \( S^* \) defined below attempts to capture the temporal ordering relation imposed on the flow of signals by the ACS-external world combination.

Definition

Let \( z, y \in S \) such that \( z \in C(y) \). Let \( D(z, y) = k \). Then,

1) \( y^{2-k} \leq z^1 \)

2) \( \forall x^i, y^j \in S^*, x^i \leq y^j \Rightarrow x^{i+1} \leq y^{j+1} \)
3) \( \forall x^i \in S^* \), \( x^i \leq x^i \)

4) \( \forall x^i, y^j, z^k \in S^* \), \( x^i \leq y^j \) and \( y^j \leq z^k \Rightarrow x^i \leq z^k \)

\( (S^*, \leq) \) will be the poset model representation of the given ACS.

From the above definition, it is obvious that the relation "\( \leq \)" is reflexive and transitive. Hence we only need to show that "\( \leq \)" is antisymmetric in order to establish the fact that it is a partial ordering relation.

**Proposition 3-1**

"\( \leq \)" as defined in this section is anti-symmetric.

**Proof** (By contradiction)

Let \( x^i, y^j \in S^* \) such that, \( x^i \leq y^j \) and \( y^j \leq x^i \).

**Case 1** \( i > j \)

Let \( i = j + k \), \( (k \geq 1) \)

\[ y^j \leq x^i = y^1 \leq x^{i-(j-1)} \]  (Part 2 of DEF of "\( \leq \)"

\[ = y^1 \leq x^{k+1} \]  \( (k + 1 \geq 2) \).

But \( y^1 \leq x^{k+1} \) could have been only generated by using Part 1 and Part 4 of the definition of "\( \leq \)". But this is impossible since in Part 1 of the definition, the superscript of the variable on the left hand side is always greater than or
equal to that of the variable on the right-hand side
[D(z,y) ≤ 1 by definition] and Part 4 of the definition
does not alter the values of the superscripts. Hence,

\[ i > j = y^j \leq x^i \quad \text{and hence the proposition.} \]

Case 2  \[ j > i \]

The proof is the same as that for Case 1 with "x"
and "y" interchanged in the argument.

Case 3  \[ i = j \quad \text{and} \quad x \neq y \]

Assume \( x^i \leq y^j \quad \text{and} \quad y^j \leq x^i \).

\[ i = j \Rightarrow x^1 \leq y^1 \quad \text{and} \quad y^1 \leq x^1. \]

\( x \leq y = \exists x_1, x_2, \ldots, x_n \in S, \text{ such that} \)

1) \( x_1 = x \)

2) \( x_n = y \)

3) \( x_{i+1} \in C(x_i) \quad i = 1, 2, \ldots, n-1 \)

and \( D(x_{i+1}, x_i) = 1 \) (by definition)

This implies that \( \exists \) a path from \( p(x) \) to \( p(y) \) that does not
contain any maximal places. Similarly, \( y \leq x = \exists \) a path
from \( p(y) \) to \( p(x) \) that does not contain a maximal place.

Therefore, \( x \leq y \) and \( y \leq x = \exists \) a circuit in the SFP that
does not contain a maximal place. This implies that in the corresponding marked graph, there exists a circuit that does not contain any markers, a contradiction since this implies that the graph is not live.

Hence \( x^i \leq y^j \) and \( y^j \leq x^i \) both hold only when \( x = y \) and \( i = j \) and the proposition is proved.

**Convention**

"\( \lambda \)" is the greatest element of \( \langle S^*, \leq \rangle \). As an example, the poset model representation of a JUNCTION-module is shown in FIG-3.1 in terms of the Hasse diagram of the poset \( \langle S^*, \leq \rangle \). We note that, \( C(3r) = \{1r, 2r\} \) and \( D(1r, 3r) = D(2r, 3r) = 1 \).

Hence \( 3r^1 \leq 1r^1 \) and \( 3r^1 \leq 2r^1 \).

Also, \( C(1r) = \{1a\} \) and \( D(1a, 1r) = 0 \)

Hence, \( 1r^{2-0} = 1r^2 \leq 1a^1 \) etc.

In the Hasse diagram, \( \lambda \) will not be shown and the superscripts of the various elements of \( S^* \) will be deleted, for the sake of convenience.
FIGURE-3.1
Definition

Let \( x^i \in S^* \). Then,

\[
\text{Pred} (x^i) = \{ y^j \in S^* \mid 1) y^j \neq x^i \quad \text{and} \\
\quad 2) x^i \leq y^j \}
\]

(Pred \( \rightarrow \) Predecessors)

Definition

Let \( \bar{x} \in (S^*)_* \) [(\( S^* \))_* is defined in the same manner as \( S_* \) was defined in Chapter 2.]

Let \( \bar{x} = x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \)

Then \( \bar{x} = \{ x_1^{i_1}, x_2^{i_2} \ldots x_n^{i_n} \} \).

Definition

Let \( \bar{x} \in (S^*)_* \). Then,

\[
\text{En} (\bar{x}) = \{ y^j \in S^* \mid 1) y^j \notin \bar{x} \\
\quad 2) \text{Pred} (y^j) \subseteq \bar{x} \}
\]

As was pointed out in Chapter 2, the poset model is meaningful only when the rules for "simulating" it are given. As was done in the case of C-D model, the simulation rule is specified in terms of specifying the rules for generating feasible signal sequences from poset model.
specification. Once again, the specification of the rule is inductive.

**Basis Step**

"λ" is a feasible sequence.

**Induction Step**

Let $\bar{x}$ be a feasible sequence, such that $\ell_g(\bar{x}) = i \ (i \geq 0)$. Then $\bar{xy}^j$ is a feasible sequence iff $y^j \in \text{En}(\bar{x})$.

The set of all possible signal sequences generated by the above rule is denoted by $\text{FS}^{II}$. However, if the poset model is to be a faithful representation, we must be able to associate a signal sequence consisting of symbols from the set $S$ alone, with each simulation of the model, i.e., with each signal sequence is the set $\text{FS}^{II}$. This association is achieved by defining the following "refinement operation".

**Definition**

Let $\bar{x} \in \text{FS}^{II} = x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$

Then $\text{REF}(\bar{x}) = x_1 x_2 \ldots x_n$. (i.e., the superscripts are deleted)
Definition

\[ \text{FS}_1 = \text{Ref (FS}_{11}) = \{\text{Ref (x)} \mid x \in \text{FS}_{11}\} \]

The formulation of the poset model is now complete and in the next section, its validity is established.

3-3

In 2-4, it was shown that FS' corresponds to all possible signal sequences generated by the ACS through all possible stretches of behavior. Each element of FS\(_1\) corresponds to a particular simulation of the poset model. If the poset model is to be a valid model, corresponding to every possible stretch of behavior of the ACS, there must be a simulation of the poset model and vice-versa. Hence it is necessary and sufficient to show that FS\(_1\) = FS' to establish the validity of the poset model. We start by proving the following useful lemmas.

**Lemma 3.1** \( \forall x \in S, \ x^{i+1} \leq x^i \) \( (i = 1, 2, \ldots) \)

**Proof** (by cases)

**Case 1** \( x \in \text{PI} \)

\( p(x) \) will occur in the SFP representation as shown in FIG-3.2.a.
FIGURE-3.2.a

FIGURE-3.2.b
In the corresponding marked graph there will be three arcs, \(a_1, a_2\), and \(a_3\), corresponding to the places \(p(x'), p\) and \(p(x)\) respectively and two vertexes \(v_1\) and \(v_2\) corresponding to the transitions \(t_1\) and \(t_2\) respectively as shown in FIG-3.2.b. However, in a live marked graph, an arc is safe iff it is contained in a basic circuit. 

Hence the arc \(a_1\) (FIG-3.2.b) must be contained in a circuit that contains one and only one marker. But it is easy to see that any circuit that contains the arc "\(a_1\)" must also contain the arcs \(a_2\) and \(a_3\). Since "\(a_2\)" contains a marker, there must be a path from \(a_3\) to \(a_1\) that does not contain any markers. In terms of FIG-3.2.a then, there is a path from \(p(x)\) to \(p(x')\) that does not contain any maximal places. Denote this path by, \(p_1p_2\ldots p_n\) where \(p_1 = p(x)\) and \(p_n = p(x')\). Considering the facts that, 1) the SFP of the given ACS is finite, and 2) the \(R_x\) relation defined in Chapter 2 has been shown to be antisymmetric, we can conclude that there exists a maximal-place-free path from \(p(x)\) to \(p(x')\) of the form,

* The reader who is unfamiliar with marked graph should refer to Appendix-A.
\( P_{i1} \ldots P_{i2} \ldots P_{ij} \ldots P_{i\ell} \) such that,

a) \( P_{i1} = p(x) \)

b) \( P_{i\ell} = p(x') \)

c) \( P_{ij} \in \mathbb{L}_P, \quad j = 1, 2, \ldots, \ell \)

d) \( k(p_{ij}) \in C[\ell(p_i(j+1))] \), \( j = 1, 2, \ldots, \ell-1 \)

Let \( k(p_{ij}) = w_j \), \( j = 1, 2, \ldots, \ell \).

Then we also have, by definition,

e) \( D(w_j, w_{j+1}) = 1 \) \( (j = 1, 2, \ldots, \ell-1) \).

\( d) \) and \( e) \)

\( w_2 \leq x, \ w_3 \leq w_2, \ldots, x' \leq w_{\ell-1} \)

\( = x' \leq x \) (by transitivity of \( \leq \))

But \( x \in \mathbb{P}_I \Rightarrow x' \in C(x) \) and \( D(x', x) = 0 \)

\( = x^2 \leq x' \)

Hence, \( x^2 \leq x^1 \) and the lemma follows immediately by the definition of \( \leq \).
Case 2 \( x \in SO \)

\[
x \in SO \Rightarrow x' \in PI
\]

\[
\Rightarrow x \leq x' \quad \text{(by the proof of Case 1)}
\]

\[
\Rightarrow x^2 \leq (x')^2 \quad \text{(by the definition of } \leq \text{)}
\]

But \( x' \in PI \Rightarrow (x')^2 \leq x \)

Hence, \( x^2 \leq x \) and the lemma follows.

Case 3 \( x \in SI \)

We can use exactly the same arguments that were used for proving the lemma for Case 1. The only difference being, in this case, there will be one maximal place in the appropriate path from \( p(x) \) to \( p(x') \) and none from \( p(x') \) to \( p(x) \) so that we can finally obtain

\[
(x')^2 \leq x
\]

But, \( x \in SI \Rightarrow C(x) = \{x'\} \) and \( D(x', x) = 1 \)

\[
\Rightarrow x \leq x'
\]

\[
\Rightarrow x^2 \leq (x')^2
\]

Hence \( x^2 \leq x \).
Case 4 \( x \in PO \)

The lemma is proved for this case using the proof for Case 3 just in the same way that Case 2 was proved using the proof for Case 1.

Lemma 3.2

Let \( \bar{x} \in FS^{II} \) and \( y^j \in En(\bar{x}) \).

Then

1) \( y^1, y^2, \ldots, y^{j-1} \in \bar{x} \)

2) \( \# y [\text{Ref}(\bar{x})] = j-1 \)

Proof

Follows immediately from,

1) Lemma 3-1,

2) Transitivity of \( \leq \),

3) Definition of Pred \((y^j)\),

4) Definition of \( En(\bar{x}) \), and

5) Definition of \( \text{Ref}(\bar{x}) \).
**Definition**

Let $y^j \in S^*$. Then,

$$\text{Cov} (y^j) = \{ x^i \in \text{Pred} (y^j) \mid \not\exists z^k \text{ such that },$$

1) $y^j \leq z^k \leq x^i$,

2) $z^k \neq x^i$ and

3) $z^k \neq y^j \}$$

**Remark**

$\forall \overline{x} \in FS^{II}$ and $y^j \in S^*$,

$$\text{Pred} (y^j) \trianglelefteq \overline{x} \Rightarrow \text{Cov} (y^j) \trianglerighteq \overline{x}$$

**Lemma 3-3**

Let $y^j \in S^*$. Then,

$$z^i \in \text{Cov} (y^j) \iff \begin{cases} 1) z \in C(y) \text{ and } \\ 2) D(z,y) = i + 1 - j \end{cases}$$

**Proof**

$z^i \in \text{Cov}(y^j) \Rightarrow y^j \leq z^i \ (j \geq i)$

$\Rightarrow y^{j-i+1} \leq z$

Hence, $z \not\in C(y) = \exists w^k \in S^* \text{ such that }$

$y^{j-i+1} \leq w^k \leq z$
Thus \( z \in C(y) \). Furthermore,

\[
\begin{align*}
z \in C(y) \quad \text{and} \quad y^{j-i+1} \leq z & \Rightarrow 2 - D(z,y) = j-i+1 \\
& \Rightarrow D(z,y) = i+1-j
\end{align*}
\]

\[
\begin{align*}z \in C(y) \quad \text{and} \quad D(z,y) = i+1-j & \Rightarrow y^{j-i+1} \leq z \\
& \Rightarrow y^j \leq z^i
\end{align*}
\]

Supposing \( z^i \notin \text{Cov}(y^j) \). Then (using the conventional terminology associated with posets), assume without any loss of generality that \( w^k \) is the only other element in the interval \([y^j, x^i]\). But,

\[
y^j \leq w^k \leq x^i = y^{j-k+1} \leq w \quad \text{and} \quad w^{k-i+1} \leq z
\]

\[
\begin{align*}
& \Rightarrow 1) \ w \in C(y) \ ; \ 2) \ D(w,y) = k-j+1 \\
& \quad 3) \ z \in C(w) \ ; \ 4) \ D(z,w) = i-k+1
\end{align*}
\]

\[
\begin{align*}
& \exists \ \text{a path from} \ p(w) \ \text{to} \ p(y) \ \text{containing} \\
& 1-(k-j+1) \ \text{maximal place and} \ \exists \ \text{a path from} \\
& p(z) \ \text{to} \ p(w) \ \text{containing} \ 1-(i-k+1) \ \text{maximal places}
\end{align*}
\]

\[
\begin{align*}
& \exists \ \text{a path from} \ p(z) \ \text{to} \ p(y) \ \text{passing} \ p(w) \\
& \ \text{containing} \ j-i \ \text{maximal places.}
\end{align*}
\]
But \( D(z,y) = i + l - j \Rightarrow N(z,y) = j - i \)

Hence the assumption that \( z^i \in \text{Cov}(y^j) \) must be false since it leads to the contradiction that \( z \notin C(y) \). Thus the lemma is proved.

A similar argument may be extended to the case where the interval \([y^j, z^i]\) is assumed to contain more than one element (other than \( y^j \) and \( x^i \)), by breaking up the interval into sub-intervals containing only one intermediate element. In each case the contradiction that \( z \notin C(y) \) can be arrived at.

The validity of the poset model can now be established.

\textbf{Lemma 3-4} \quad \text{FS}' \subseteq \text{FS}^I

\textbf{Proof} \quad \text{By induction on } \ell g(\overline{x}) .

\textbf{Basis Step}

Let \( \ell g(\overline{x}) = 0 \). Then the lemma is obviously true.

\textbf{Induction Step}

Assume the lemma to be true for all \( \overline{x} \in \text{FS}' \) such that \( \ell g(\overline{x}) = i \) \( (i \geq 0) \). Let \( \overline{xy} \in \text{FS}' \) such that \( \ell g(\overline{xy}) = i + 1 \) and \( \# y(\overline{x}) = j \).
\[ \log(xy) = i+1 \Rightarrow \log(x) = i \text{ and } y \in \text{En}(x) \]
\[ = \forall z \in C(y), \#z(x) - \#y(x) \geq k, \]
where \( D(z,y) = k \)
\[ = \forall z \in C(y), \#z(x) \geq k + j \]

By induction hypothesis,
\[ \overline{x} \in FS' \Rightarrow \overline{x} \in FS^I \]
\[ = \exists (\overline{x})'' \in FS^{II} \text{ such that } \text{Ref}[(\overline{x})''] = \overline{x} \]
\[ = y^j, z^{k+j} \in (\overline{x})'' \text{ (by lemma 3.2)}. \]

But by lemma 3-3,
\[ \text{Cov}(y^{j+1}) = \left\{ z^{j+k} \mid z \in C(y) \text{ and } D(z,y) = k \right\} \]

Hence,
\[ \overline{xy} \in FS' \Rightarrow (\overline{x})'' \in FS^{II} \text{ and } \text{Cov}(y^{j+1}) \subseteq (\overline{x})'' \]
\[ = (\overline{x})'', y^{j+1} \in FS^{II} \]
\[ = \overline{xy} \in FS^I \]

**Lemma 3-5** \[ FS^I \subseteq FS' \]

**Proof** \[ By \text{ induction on } \log(\overline{x}). \]
Basis Step

Let \( \log(\overline{x}) = 0 \). Then the lemma is obviously true.

Induction Step

Assume the lemma to be true for all \( \overline{x} \in \mathcal{FS}^I \) such that \( \log(\overline{x}) = i \) \((i \geq 0)\).

Let \( \overline{xy} \in \mathcal{FS}^I \) such that \( \log(\overline{xy}) = i+1 \)

\[ \overline{xy} \in \mathcal{FS}^I = \exists (\overline{x'})y^j \in \mathcal{FS}^{II} \text{ such that,} \]
\[ \text{Ref } [(\overline{x})', y^j] = \overline{xy} \]
\[ \Rightarrow \#y(\overline{x}) = j-1 \text{ and } \text{Cov}(y^j) \subseteq \overline{x} \]
\[ \Rightarrow \forall z \in C(y), z^{k+j-1} \in \overline{x} \]
(by lemma 3-3), where \( D(z,y) = k \)
\[ \Rightarrow \overline{x} \in \mathcal{FS}' \text{ (by the induction hypothesis) and,} \]
\[ \forall z \in C(y), \#z(\overline{x}) - \#y(\overline{x}) = k \geq D(z,y) \]
\[ \Rightarrow \overline{xy} \in \mathcal{FS}' \]

Theorem 3-1 \( \quad \mathcal{FS}' = \mathcal{FS}^I \)

Proof

Follows at once from lemmas 3-4 and 3-5. Thus, the validity of the poset model is now established.
In this chapter, as an example of the application of the two algebraic models that have been formulated, a simple scheme for obtaining a physical realization for the class of ACS's under study is demonstrated.

In section 4-2, a straight-forward method for implementing a precedence graph is developed. The concept of a standard-form-realization is also introduced.

In section 4-3, making use of the fact that the Hasse diagram of the poset model may be viewed as an infinite precedence graph, the proposed scheme of implementation is developed.

Section 2-4 establishes the fact that the implementation obtained by using the above scheme is a speed-independent and faithful realization of the corresponding ACS.

A precedence graph is an acyclic finite directed graph that represents the temporal ordering imposed on the occurrences of a set of events, where each event in the set occurs only once. Thus each vertex of the precedence graph
will have an event associated with it.

The precedence graph may be viewed as the Hasse diagram of a partial ordering relation (which is what the temporal ordering is) imposed on a set (the elements of this set will be events). Denoting this partial ordering relation by \( \leq \), we have the following definition.

**Definition**

Let \( E_i \) be a vertex of a precedence graph. Then,

\[
\text{Cov} (E_i) = \left\{ E_j \mid 1) \ E_j \text{ is a vertex of the precedence graph} \\
2) E_i < E_j \text{ and} \\
3) \nexists E_k \text{ in the vertex set of the precedence graph such that } E_i \leq E_k \leq E_j \right\}
\]

The physical implementation of a precedence graph will have for its input a distinguished signal called "START" (which will cause action to begin) and one signal corresponding to the termination of each event of the precedence graph. Its output signals will consist of a distinguished signal called "END" (which will signify the ceasing of action) and one signal corresponding to the initiation of each event of the precedence graph.
The following definitions will aid in stating the proposed scheme for implementing precedence graphs in a precise manner.

**Definition**

Let $E$ be an event of the precedence. (i.e., $E$ is the event associated with the vertex of a precedence graph) Then $E \cdot b$ is the signal corresponding to the initiation of the event $E$. Also, $E \cdot t$ is the signal corresponding to the termination of the event $E$.

**Definition**

Let $E_i$ be an event of a precedence graph. Then,

$$ En(E_i) = \{ E_j \cdot t \mid E_j \in Cov (E_i) \} . $$

Using this terminology, it is easy to see that in the physical implementation of a precedence graph, the signal $E \cdot b$ should be generated by the implementation iff it has received the set of signals $En(E)$. Using Muller's C-elements* to "sense" $En(E)$ for each event, a straightforward implementation for a precedence graph can be easily

* A nice discussion on Muller's C-elements may be found in [9].
obtained. The method is best illustrated through an example. A precedence graph is shown in FIG-4.1 and its implementation is shown in FIG-4.2.

Signals are assumed to be represented by changes in levels rather than the levels themselves. The blocks marked with "C" in FIG-4.2 are the C-elements. That the structure shown in FIG-4.2 is a faithful implementation of the precedence graph shown in FIG-4.1 is obvious.

If in a precedence graph, two ordered vertexes correspond to the two occurrences of the same event then above implementation procedure can still be used with the addition of UNION modules [7] at the output side that would encode the outputs generated by the original implementation in a suitable fashion. For example, in the precedence graph shown in FIG-4.1 let the occurrences of the events A, B and F correspond to three occurrences of the same event $E_1$ and let the occurrences of the events D, G and H correspond to three occurrences of the same event $E_2$. Then the modified implementation that achieves the required encoding at the output side is shown in FIG-4.3.

The blocks marked with "U" represent UNION modules and $M'$ is the original implementation shown in FIG-4.2.
FIGURE-4.3
For the rest of this chapter for a precedence graph that contains multiple, ordered occurrences of the same event, by "the implementation of the precedence graph" we would mean a structure of the form shown in FIG-4.3.

The following definition will aid in specifying the inputs to the UNION modules in a precise manner.

**Definition**

Let PG be a precedence graph and \( \{E_1, E_2, \ldots, E_i\} \) be the set of events associated with PG. Then given an interpretation,

\[
E^x \sim E^y \quad \text{if and only if} \quad E^x \text{ and } E^y \text{ correspond to different occurrences of the same event } E'.
\]

**Remark**

\( R_E \) is an equivalence relation.

We conclude this section by noting that the proposed implementation for an arbitrary precedence graph can be represented in a standard form as shown in FIG-4.4.

In the figure mentioned above, the "START" and "END" signals have been omitted. The vertex set of the precedence graph has been assumed to be the set \( \{E_1, E_2, \ldots, E_i\} \). Furthermore, it has been assumed that \( R_E \) partitions the vertex set into \( j \) blocks (\( j \leq i \)).
"INPUT SELECTOR" is a component-free structure that simply fans out the incoming wires to the appropriate places. "OUTPUT ENCODER" will consist of UNION modules and there will be as many UNION modules as there are blocks in the partition defined by $R_E$.

4-3

The principle behind the proposed scheme for implementing the ACS under study may be explained as follows:

Consider an output link of an ACS. If an event is associated with this link, then the ready signal sent out on this link may be interpreted as the initiation of the associated event and the acknowledge signal received on this link may be interpreted as the termination of the associated event.

In the case of an input link the situation is slightly more complicated. However, if we assume that, for an input link, operation begins by sending out an acknowledge signal on the input link (the first of which, the external world must be trained to ignore) then an event can be associated with an input link also. The ACS initiates this event by sending out an acknowledge signal on the corresponding input link and senses the termination of this event when it receives a ready signal on the same input link.
Under this interpretation, the ACS may be viewed as the implementation of an infinite precedence graph in which a set of \( m+n \) events occur repeatedly and endlessly. It is easy to see that the poset model description faithfully displays the temporal ordering imposed on the repeated occurrences of these \( m+n \) events. Hence, starting from the poset-model description, the corresponding infinite precedence graph implemented by the ACS can be easily obtained. Furthermore, due to the time-invariant nature of \( C(x) \) and \( D(y,x) \) (\( \forall x, y \in S \)), the infinite precedence graph thus obtained will be "periodic" in some sense. Consequently, if we implement only one "period of operations" of this precedence graph and provide suitable means for setting the implementation back to "appropriate initial states" in a speed-independent fashion, then we should have an implementation of the ACS under study.

The following developments formalize the notions introduced in the above, informal discussion.

**Remark**

The Hasse diagram of the poset model description of the ACS "M" will be denoted by \( \text{HSD}(M) \).

For the sake of convenience the form of the ACS is once again displayed in FIG-4.5.
FIGURE-4.5
Definition \[ S^0 = \{1a^0, 2a^0, \ldots, na^0\} \]

Definition \[ S^i = \left( \bigcup_{j=1}^{i} S^i \cup S^0 \right) - S^i \quad (i \geq 1) \]

Remark \[ \leq_i \] will denote the relation \( \leq \) defined on the elements of \( S^* \), restricted to the elements of \( S^i \).

Definition \[ \leq_i = \leq \cup \{(jr^1, ja^0) \mid j=1,2,\ldots,n\} \]

Remark \[ \langle S^i, \leq_i \rangle \] will be referred to as a "finite portion" of \( HSD(M) \). Furthermore, we will frequently write \( \leq \) instead of \( \leq_i \) when there is no scope for confusion.

Definition

Let \[ \langle S^i, \leq \rangle \] be a finite portion of \( HSD(M) \). Then the corresponding precedence graph is denoted by \( PG(i) \) and is defined by,

\[ PG(i) = \langle E, \leq \rangle \quad \text{where,} \]

\[ E = \left\{ E_{x,y} \mid y=1,2,\ldots,i \right\} \]

\[ x=1,2,\ldots,n,(n+1),\ldots,(n+m) \]
≤_g is defined by cases as follows.

Let \( E_x^Y, E_n^V \in \mathcal{E} \). Then,

**Case 1** \( x, u \leq n \)

\[
E_x^Y \leq_g E_u^V = (xa)^{y-1} \leq (ur)^V
\]

**Case 2** \( x \leq n \); \( n < u \leq n+m \)

\[
E_x^Y \leq_g E_u^V = (xa)^{y-1} \leq (ua)^V
\]

**Case 3** \( n < x \leq n+m \); \( u \leq n \)

\[
E_x^Y \leq_g E_n^V = (xr)^Y \leq (ur)^V
\]

**Case 4** \( n < x \); \( u \leq n+m \)

\[
E_x^Y \leq_g E_u^V = (xr)^Y \leq (ua)^V
\]

**Remark**

We will frequently write \( \leq \) instead of \( \leq_g \).

The above definition becomes much more meaningful when the signals of PG(i) are associated with the elements of \( S^i \).
Definition

\[ \text{Sig}[PG(i)] = \left\{ E_x^y \cdot b, E_x^y \cdot t \mid y=1,2,\ldots,i \right\} \]

\[ x=1,2,\ldots,n, \quad n+1,\ldots,n+m \]

Definition

The correspondence between the signals of \( PG(i) \) and \( S^i \) is established by the function, \( f : \text{Sig}[PG(i)] \rightarrow S^i \)

which is defined by cases as follows.

Case 1 \( x \leq n \) Then,

\[ f(E_x^y \cdot b) = (xa)_y^{-1} , \quad y=1,2,\ldots,i \]

\[ f(E_x^y \cdot t) = (xr)_y , \quad y=1,2,\ldots,i \]

Case 2 \( n < x \leq n+m \) Then,

\[ f(E_x^y \cdot b) = (xr)_y , \quad y=1,2,\ldots,i \]

\[ f(E_x^y \cdot t) = (xa)_y , \quad y=1,2,\ldots,i \]

Remark

\( f \) is obviously a 1-1 correspondence.
Remark

Since the relationship between the elements of $\text{Sig}[\text{PG}(i)]$ and the elements of $S^1$ is quite intimate, we will frequently write $(xa)^{Y-1}$ instead of $E_x^Y \cdot b$ and write $(xr)^Y$ instead of $E_x^Y \cdot t$ whenever $x \leq n$. A similar remark applies for the case when $n < x \leq n+m$.

A more detailed outline of the proposed scheme of implementation may now be given.

1) A value $K (K \geq 1)$ will be determined from the SFP specification.

2) Corresponding to this value of $K$, PG($K$) will be formed and implemented in the form shown in FIG-4.4.

3) At this stage, the inputs to the C-element, the output of which is identified as $E_x^Y \cdot b$ will be the set of signals defined by $E_x^Y$. In the proposed scheme, there will be an additional input to the C-element (for certain $E_x^Y$, this additional input could be absent) which will be generated as shown in FIG-4.6.

The NOR gate shown in FIG-4.6 can be used to advantage to set all the signal levels to "0" initially by simply holding the START signal level alone at "1" and temporarily breaking all the feed-back paths. The START signal will be
represented by a single \( l \to 0 \) change after which, it will be held at "0".

In the definitions that follow the set \( \mathbb{E}_x^y \) is precisely defined as also the value of \( K \).

**Definition**

\[ \forall x, y \in S, \text{ a path from } p(x) \text{ to } p(y) \text{ is said to be a minimal path if the number of maximal places contained in this path is not greater than that contained in any other path from } p(x) \text{ to } p(y). \]

**Definition**

\[ \forall x, y \in S, \]

\[ N'(x|y) = \text{The number of maximal places contained in a minimal path from } p(x) \text{ to } p(y). \]

**Remark**

\( mg(M) \) is finite and live and hence is strongly connected.

Thus, \( \forall x, y \in S, \) \( N'(x|y) \) is well-defined.

**Definition**

\[ \forall x, y \in S, \]

\[ CN(x,y) = N'(x|y) + N'(y|x). \]

**Definition**

\[ K = \max \{ CN(x,y) \mid x, y \in S \} \]
Definition

Let $E_i^\ell$ be an event of $PG(K)$. Denote $f(E_i^\ell \cdot b)$ by $(ix)^{j\star}$. Then

$\forall y \in C(ix),$

If $(2 - j - k) > 0$ then $y^{K+k+j-1} \in En(E_i^\ell)$, where,

$k = D(y, ix)$

Remark

In the above definition, we should actually write $f^{-1}(y^t)$ instead of $y^t$. However, as mentioned before, since the relationship between $S^K$ and $PG(K)$ is transparent, this should cause no confusion. In fact we will frequently write $En(ia)^{k-1}$ and $En(ia)^{k-1}$ instead of $En(E_i^\ell)$ and $En(E_i^\ell)$, when $i \leq n$. A similar remark applies for the case where $n < i \leq n+m$.

The proposed scheme of implementation may now be stated precisely as follows:

1) Compute the value of $K$ for the given SFP.
2) Implement $PG(K)$ in the standard form shown in FIG-4.3.

* "x" = "a" and $j = \ell-1$ if $i \leq n$.

* "x" = "r" and $j = \ell$ if $n < i \leq n+m$. 
3) Use the equivalence relationship defined by
\[ E^Y_x \sim E^V_u \Rightarrow x = u \]
to design the output encoder part of the implementation.

4) Compute \( \text{En}(E^Y_x) \) for every \( E^Y_x \) that is an event of \( \text{PG}(K) \).
Then generate the additional input to the C-element
whose output is identified as \( E^Y_x \cdot b \), as shown in
FIG-4.6.

As an example, consider the ACS whose SFP description
is given in FIG-4.7. By inspection,
\[ K = \text{CN}(2a, 1a) = 3 \]
The Hasse diagram of \( (S^3, \leq) \) for this ACS is shown
in FIG-4.8. In order to illustrate the principle behind
the proposed scheme part of \( \text{HSD}(M) \) not contained in \( (S^3, \leq) \)
is also shown. The dotted line indicates the separation
between \( (S^3, \leq) \) and the remaining part of \( \text{HSD}(M) \).

In FIG-4.9, PG(6) is shown and the separation between
PG(3) (which is only of interest) and the remaining portion
of PG(6) is again shown by a dotted line. The labels A,
B, C, D and E have been used to identify \( \text{En}(E^Y_x) \) whenever
present for each \( E^Y_x \) in PG(3).

In FIG-4.10 the actual implementation is shown. The
details about the input selector and output encoder have
not been shown. \( f(E^Y_x \cdot b) \) and \( f(E^Y_x \cdot t) \) have been
FIGURE-4.9
FIGURE 4.10
indicated in the diagram rather than $E_x^y \cdot b$ and $E_x^y \cdot t$.
The C-modules marked with "x" can be deleted since they have only one input each. Finally, the range of values for $J$ and $L$ are,

$$J = 0, K, 2K, \ldots, \text{etc.}$$

$$L = 0, 1, 2, 3, \ldots, \text{etc.}$$

4-4

In the first part of this section, the speed-independence of the implementation obtained by using the scheme developed in the previous section is established. This is done by showing that a marked graph that may be associated with the implementation has the appropriate properties. In the second part of this section, that the implementation is a faithful realization is shown.

Notation

Given an ACS $M$, $\text{PGI}(M)$ will denote the precedence graph implementation of $M$ that was developed in the previous section.

Remark

$K$ and $\text{PG}(K)$ will have the same connotations that they had in the previous section.
The following lemmas will be used for specifying the marked graph that will be associated with PGI(M) and for showing that it has the appropriate properties.

**Lemma 4-1**

Let $z, y \in S$ such that $z \in C(y)$. Then,

$$2 - D(z, y) = 1 + N(z|y).$$

**Proof**

Follows immediately from the definition of $D(z, y)$.

**Lemma 4-2**

Let $z, y \in S$. Then,

$$N'(z, y) = J = y^{1+J} \leq z.$$

**Proof** By induction on $J$.

**Basis Step** Let $J = 0$.

Due to the finiteness of SFP(M), \( \exists z_1, z_2, \ldots, z_j \in S \) such that

1) $z_1 = z$; 2) $z_j = y$

3) $z_i \in C(z_{i+1})$ \( i = 1, 2, \ldots, j-1 \)

4) $D(z_i, z_{i+1}) = 1$ \( i = 1, 2, \ldots, j-1 \)
(1), (2), (3) and (4) = \[ z_{i+1} \leq z_i \quad i=1,2,\ldots,j-1 \]
\Rightarrow y \leq x

**Induction Step**

Assume the lemma to be true \( \forall J \leq L \) \( (L \geq 0) \). Let \( J = L + 1 \). Again due to the finiteness of \( \text{SFP}(M) \),
\[ \exists z_1,z_2,\ldots,z_k,z_{k+1}\ldots z_j \in S \quad \text{such that}, \]
1) \( z_1 = z \); 2) \( z_j = y \)
3) For some \( k+1 \leq j \), \( N'(z_k | z_{k+1}) > 0 \)
4) \( N'(z_\ell | z_{\ell+1}) = 0 \), \( \ell = 1,2,\ldots,k-1 \).
5) \( z_i \in C(z_{i+1}) \), \( i = 1,2,\ldots,j-1 \).

(4) and (5) \( \Rightarrow z_k \leq z \) (by the proof of the basis step)

\[ z_k \in C(z_{k+1}) \Rightarrow N'(z_k | z_{k+1}) = N(z_k | z_{k+1}) \]

Let \( N(z_k | z_{k+1}) = u [u > 0, \text{ by (3)}] \). Then,

we have, by definition, \( z_{k+1}^{l+u} \leq z_k \).
u > 0 and (4) = \( N'(z_{k+1} | y) = L+1-u < L+1 \)

\[ = L+1-u \leq L \]

\[ = y^{L+1-u+1} \leq z_{k+1}, \text{ by the induction hypothesis} \]

\[ = y^{(L+1)+1} \leq z_{k+1} \leq z \]

**Lemma 4-3**

Let \( x^i, y^j \in S^* \). Then,

\[ x^i \in \text{Cov}(y^j) \Rightarrow x^{i+K} \leq y^j \]

**Proof**

\[ x^i \in \text{Cov}(y^j) \Rightarrow 1) \ x \in C(y) \]

\[ 2) D(x,y) = i-j+1 \text{ by lemma 3-3.} \]

\[ D(x,y) = i-j+1 \Rightarrow N(x|y) = N'(x|y) = j-i, \]

by lemma 4-1.

Let \( N'(y|x) = K_1 \). Then by lemma 4-2, \( x^{1+K_1} \leq y \)

and hence \( x^{K_1+j} \leq y^j \).

Now, \( \text{CN}(x,y) \leq K \) by definition of \( K \). But
\[ CN(x, y) \leq K \Rightarrow K_1 \leq K - N(x|y) \]

\[ = K_1 + j \leq K - j + i + j \]

\[ = x^{K+i} \leq x^{K_1+j} \leq y^j \]

and the lemma is proved.

**Lemma 4-4**

Let \( E_x^Y \) be an event of \( PG(K) \). Then,

\[ (ix)^j \in E_n (E_x^Y) \Rightarrow (ix)^j \leq f(E_x^Y \cdot b) \]

**Proof** By cases.

**Case 1** \( i \leq n \) and \( x \leq n \)

\[ x \leq n \Rightarrow f(E_x^Y \cdot b) = (xa)^{y-1} \]

\[ i \leq n \Rightarrow (ix)^j = (ix)^j \cdot \]

\[ (ir)^j \in E_n [(xa)^{y-1}] \Rightarrow 1) \ ir \in C(xa) \]

\[ 2) j = K+k+(y-1)-1 \] where

\[ D(ir, xa) = k, \quad i.e., \]

\[ k = j - [(y-1)+K] + 1 \]
We claim that \((ir)^j \in Cov(xa)^{y-1+K}\). To this end, we only need to show that

1) \(ir \in C(xa)\);

2) \(D(kr, xa) = j - [(y-1) + K] + 1\),

by lemma 3-3.

But we already have shown (1) and (2) to be trivially true and hence the claim is indeed true. But,

\[(ir)^j \in Cov(xa)^{y-1+K} = (ir)^{j+K} \leq (xa)^{y-1+K}\]

by lemma 4-3

\[= (ir)^j \leq (xa)^{y-1} .\]

The proof for the three remaining cases is identical and hence has been omitted.

**Lemma 4-5**

Let \(E_x^Y\) be an event of \(PG(K)\). Then,

\[(ix)^j \in En(E_x^Y) \Rightarrow (ix)^j \in \overline{S^K} .\]

**Proof**  By cases.

**Case 1** \(i \leq n\) and \(x \leq n\)

Then, we have,

\[(ir)^j \in En(xa)^{y-1} .\] But,
(ir)\(^j\) \in E_n(xa)^{y-1} \Rightarrow j = K+k+(y-1)-1\) and 
\[2-k-(y-1) > 0\] by definition, 
where, \(k = D(ir,xa)\)
\[\Rightarrow j = K+k+y-2\] and \(k+y-2 < 1\)
\[\Rightarrow j \leq K.\]

By lemma 4-4, \(j \geq 1\), and hence the lemma is true for this case. An identical argument may be used for showing the lemma to be true for the three remaining cases also.

The following definition establishes yet another (redundant but useful) link between \(\overline{S}^K\) and \(PG(K)\).

**Definition** \{of \(E : \overline{S}^K \rightarrow \) [The set of events of \(PG(K)\)]\}

The function \(E\) is defined by cases.

**Case 1** \(k \leq n\)

\[E[(ka)^j] = E_k^{j+1}, j = 0,1,\ldots,K-1\]

\[E[(kr)^j] = E_k^j, j = 1,2,\ldots,K\]

**Case 2** \(n < k \leq n+m\)

\[E[(kr)^j] = E[(ka)^j] = E_k^j, j = 1,2,\ldots,K.\]
**Remark**

We will frequently write $E(y^j)$ instead of $E(y^j)$.

**Lemma 4-6**

Let $x^i, y^j \in S^*$ such that,

1) $x^i \in S^K$

2) $y^j \notin S^K$

3) $x^i \in \text{Cov}(y^j)$

Then,

a) $2 - D(x, y) - (j-K) > 0$

b) $K + D(x, y) + (j-K) - 1 = i$

c) $x^i \in E_n[E(y^{j-K})]$ .

**Proof**

We begin by observing that from the definition of $S^K$ and $\text{PG}(K)$, (1) and (2) => $x \in I$ and hence part-c of the above lemma is a meaningful assertion at the worst. Now for proving the lemma,

$x^i \in S^K = i \leq K$ . But

$x^i \in \text{Cov}(y^j) = 2 - D(y,x) = j-i+1$

$= 2 - D(y,x) - (j-K) = K+1 - i > 0$ ,

Since $K \geq i$.

Hence part-a of the lemma is proved.
Part-b follows directly from the fact that,
\[ 2 - D(y|x) = j-i+1. \]

Part-c follows at once from the definitions.

With these preliminaries taken care of, the marked graph associated with PGI(M) can now be introduced. Let \( E \) be an event in \( PG(K) \). From the structure of PGI(M) it is obvious that the \( J^{th} \) production of the signal \( E \cdot b \) occurs iff the \( J^{th} \) reception of every signal in the set \( En(E) \) and the \( J-1^{th} \) reception of every signal in the set \( En(E) \) has occurred (\( J \geq 1 \)). Let
\[
En(E) = \left\{ E_{11} \cdot t, \ldots, E_{1i} \cdot t \right\} \quad \text{and} \quad En(E) = \left\{ E_{21} \cdot t, \ldots, E_{2j} \cdot t \right\}
\]

If, corresponding to each event of \( PG(K) \), we have a unique vertex in a marked graph and interpret each firing of a vertex as the initiation of the event associated with that vertex followed by the termination of that event, then for the event \( E \) under discussion, in terms of a marked graph, we will have the situation shown in FIG-4.11.

Using this idea, the marked graph associates with PGI(M) -- denoted by \( mg[PGI(M)] \) or simply \( mg \) -- may be defined as follows.
**Definition**  (of mg)

1) \( V = \) The set of vertices = \( \{E|E\) is an event of \( PG(K)\} \).

2) \( \forall E_x, E_y \in V \),

\[
(E_x, E_y) \in A^* \iff E_x \cdot t \in E(E_y) \cup E(E_y).
\]

\( A' \) is the set of arcs.

3) \( \forall (E_x, E_y) \in A \),

\[
M[(E_x, E_y)] = 1 \iff E_x \cdot t \in E(E_x).
\]

Otherwise, \( M[(E_x, E_y)] = 0 \).

As an example, mg for the example worked out in the previous section is shown in FIG-4.12.

Now that mg has been defined, the properties that mg must have so that PGI(M) will be speed-independent can be arrived at through the following considerations:

PGI(M) consists of an interconnected network of C-elements, NOR gates and UNION modules (excluding the links that communicate with the external world which is

* Since in a Hasse diagram, multiple arcs between two vertexes are not present, this method of specifying arcs is valid.
FIGURE 4.12
assumed to function in a speed-independent manner).
These components function in a speed-independent manner only when certain constraints imposed on the inputs to these elements are satisfied.

To begin with, from the structure of PGI(M), and lemma 4-4 it is obvious that the NOR-gates after the appearance of the START signal will act as delay elements and hence will always function in a speed-independent fashion in the present application.

Secondly, in the case of the C-element, it functions in a speed-independent manner when it is ensured that the \( i+1 \) th input (i.e., the \( i+1 \) th change in the level of the input wire) is received only after the \( i \) th output has been generated. Now consider the C-element used for synchronizing the set of signals \( En(E_x) \) for some event \( E_x \) in PG(K). From the structure of the implementation, it is easy to see that its proper operation is assured if the \( i+1 \) th reception of \( E_y \cdot t \) occurs only after the \( i+1 \) th production of \( E_x \cdot b \) has taken places for every \( E_y \cdot t \in En(E_x) \). In terms of mg this constraint can be translated as "the vertex \( E_y \) should be firable for the \( i \) th time iff the vertex \( E_x \) has been fired for the \( i \) th time."

Since the arc from \( E_y \) to \( E_x \) will initially contain a marker, we can also translate the constraint on the C-element as "the arc from \( E_y \) to \( E_x \) should be safe." An
identical argument for the case of the C-element used for synchronizing the signals in the set \( \text{En}(E_x) \) will yield the fact that such a C-element functions in a speed-independent fashion iff the arc from \( E_z \) to \( E_x \) is safe for every \( E_z \in \text{En}(E_x) \). Hence, in effect, the C-elements used in PGI(M) function in a speed-independent manner iff \( mg \) is safe.

Finally, in the case of the UNION module, its proper functioning is assured if no two input links become active at the same time. Simple considerations about \( mg \) will show that this is ensured iff, in \( mg \), the set of vertices contained in a single block of \( \pi_E \) are contained in a basic circuit.

The following lemma will be useful for showing that \( mg \) is safe.

**Lemma 4-7**

Let \( x^i, y^j \in S^* \) such that \( x^i \leq y^j \). Then,

a) \( y \in SO \) and \( x \in I = (x')^i \leq (y')^{j+1} \)

b) \( y \in PO \) and \( x \in I = (x')^i \leq (y')^j \)

**Proof**

Follows immediately from the SFP construction and the definitions.
Lemma 4-8 \[ \text{mg is safe.} \]

Proof \[ \text{By cases.} \]

Let \((E_u, E_v)\) be an arc of \(mg\) and let \(f(E_u \cdot t) = x^i\) and \(f(E_v \cdot b) = y^j\).

Case 1 \[ M[(E_u, E_v)] = 1 \]

Then we have, \(x^i \in En(y^j)\). But,

\[ x^i \in En(y^j) \Rightarrow x^i \leq y^j, \text{ by lemma 4-4} \]

\[ \Rightarrow E(x^i) = E_u \leq E_v = E(y^j), \text{ by lemma 4-7, and the definitions} \]

\[ \Rightarrow \exists \text{ a marker-free path from } E_v \text{ to } E_u \]

in \(mg\), by the definition of \(PG(K)\) and \(mg\)

\[ \Rightarrow (E_u, E_v) \text{ is contained in a basic circuit and hence is safe.} \]
Case 2 \[ \text{M}[(E_u,E_v)] = 0 \]

Then we have, \( x^i \in \text{En}(y^j) \). But,

\[ x^i \in \text{En}(y^j) = x^i \in \text{Cov}(y^j), \] by definition

\[ = x^{i+K} \leq y^j, \] by lemma 4-3 and

\[ x^{i+K} \not\in S^K, \] by lemma 4-5

\[ = \exists z^k, w^\ell \in S^* \text{ such that,} \]

1) \( x^{i+K} \leq w^\ell \leq z^k \leq y^j \)

2) \( z^k \in S^K \)

3) \( w^\ell \not\in S^K \)

4) \( z^k \in \text{Cov}(w^\ell) \)

\[ = 1) x^i \leq w^{\ell-K} \]

2) \( \exists \) a marker-free path from \( E(y^j) = E_v \) to \( E(z^k) \) in mg

by lemma 4-7

3) \( E(z^k).t \in \text{En}[E(w^{\ell-K})] \) by lemma 4-6

\[ = 1) \exists \) a marker-free path from \( E_v \) to \( E(z^k) \) in mg

2) \( \exists \) an arc containing a marker from \( E(z^k) \) to \( E(w^{\ell-K}) \) in mg

3) \( \exists \) a marker-free path from \( E(w^{\ell-K}) \) to \( E(x^i) = E_u \) in mg
\[ (E_u, E_v) \text{ is contained in a basic circuit.} \]

\[ (E_u, E_v) \text{ is safe.} \]

Hence \( mg \) is safe.

Lemma 4-9

Let \( \mathcal{C} \) be a block of \( \pi_{R_E} \). Then there exists a basic circuit in \( mg \) that contains every element in \( \mathcal{C} \).

Proof

To begin with, because of the way \( PG(K) \) and \( R_E \) have been defined, it is obvious that every block of \( \pi_{R_E} \) will contain \( K \) elements. Let,

\[ \mathcal{C}' = \{ E^{x_1}_K, E^{x_2}_K, \ldots, E^{x_K}_K \} \]

Furthermore it is easy to see that if \( z^k \) is a maximal element of \( S^K \), then \( z^{k+K} \) will be a maximal element of \( S^* - \overline{S^K} \). The actual proof of the lemma can now be given by considering two cases.
Case 1 \(x \leq n\)

From the arguments that were used in the proof of lemma 3-2, we have, \((xa)^i \leq (xr)^i\) \(i = 1, 2, \ldots\)

Hence, \(E_{x}^{i+1} \leq E_{x}^{i}\) \(i = 1, 2, \ldots, K-1\)

Thus \(\exists\) a marker-free-path from \(E_{x}^{1}\) to \(E_{x}^{K}\) that does not contain any markers. We also have,

1) \((xa)^K \leq (xr)^K\)

2) \((xa)^0\) is a maximal element of \(\overline{S}^K\) by definition and hence \((xa)^K \notin \overline{S}^K\) is a maximal element of \(S^* - \overline{S}^K\).

Hence by the arguments that were used for proving case-2-part of lemma 4-6, \(\exists\ y^j \in \overline{S}^K\) such that,

3) \(y^j \leq (xr)^K\)

4) \(y^j \in \text{Cov}[(xa)^K]\) \(\text{[by (2)]}\)

But \(y^j \in \overline{S}^K\), (2), (4) \(\Rightarrow y^j \in \text{En}(E_{x}^{1})\) by lemma 4-6 and the definitions

\(\Rightarrow \exists\) an arc containing a marker from \(E(y^j)\) to \(E_{x}^{1}\) and \(\exists\) a marker-free-path from \(E_{x}^{K}\) to \(E(y^j)\).

Since \(y^j \leq (xr)^K\).
Since we have already shown the existence of a marker-free-path from $E_x^1$ to $E_x^K$, the lemma has been shown to hold for this case.

The proof for the case where $n < x \leq n+m$ is identical except for some minor modifications and has been omitted.

Since $mg$ has now been shown to have the required properties, we have,

Theorem 4-1

$\text{PGI}(M)$ is speed-independent.

The following lemmas will be useful for showing that $\text{PGI}(M)$ is a faithful implementation of $M$.

Lemma 4-10

Let $E$ be an event of $\text{PG}(K)$ and $f(E) = x^i$. ($0 \leq i \leq K$)

Then,

$$\text{Cov}(x^{i+K}) = \{ y^{j+K} \mid y^j \in \text{En}(x^i) \} \cup \text{En}(x^i).$$

Proof

For the sake of convenience, denote

$$\{ y^j \mid y^j \in \text{En}(x^i) \}$$

by $B$.

To begin with,

$$y^j \in \text{En}(x^i) \Rightarrow y^j \in \text{Cov}(x^i), \text{ by definition}$$

$$= y^{j+K} \in \text{Cov}(x^{i+K}), \text{ by definition}$$
Hence $B \subseteq \text{Cov}(x^{i+K})$.

Also,

$y^j \in \text{En}(x^i) \Rightarrow$

1) $y \in C(x)$

2) $j = K+k+i-1$ where,

$D(y,x) = k$.

i.e., $k = j-(K+i) + 1$

$\Rightarrow y^j \in \text{Cov}(x^{i+K})$ by lemma 3-3.

Hence $\text{En}(x^i) \subseteq \text{Cov}(x^{i+K})$. In fact,

$\text{En}(x^i) \cup B \subseteq \text{Cov}(x^{i+K})$. For showing the containment the other way,

$y^j \in \text{Cov}(x^{i+K}) \Rightarrow$

1) $y \in C(x)$

2) $D(y,x) = k = j-i-K+1$

Case 1 $2-k-i > 0$

Then, $y \in C(x) \Rightarrow y^{K+i+k-1} \in \text{En}(x^i)$

$\Rightarrow y^j \in \text{En}(x^i)$, from (2).

Thus for this case,

$\text{Cov}(x^{i+K}) \subseteq \text{En}(x^i)$
Case 2 \( 2-k-i \leq 0 \)

\[
2-k-i \leq 0 \Rightarrow 2-j+i+K-1-i \leq 0
\]

\[
\Rightarrow j > K
\]

\[
\Rightarrow j-K \in S^* \quad \text{and} \quad y^{j-K} \in \text{Cov}(x^i)
\]

Since we already have,
\[
y^j \in \text{Cov}(x^{i+K})
\]

\[
\Rightarrow y^{j-K} \in \text{En}(x^i)
\]

\[
\Rightarrow y^j \in B
\]

Hence for this case \( \text{Cov}(x^{i+K}) \subseteq B \) and the lemma is proved.

At this stage, it should be recalled that for \( x \leq n \), the first appearance of the signal \( (x_a) \) will be ignored (i.e., "unobserved"). Assuming this convention we have,

Lemma 4-11

Let \( E \) be an event of \( \text{PG}(K) \) and let \( f(E \cdot b) = x^i \) \((0 \leq i \leq K)\). Then the \( J^{th} \) production of the signal \( x^i \) by \( \text{PGI}(M) \) will be observed as the \([i + (J-1)K]^{th}\) production of the signal \( x \) at the output of the appropriate UNION module.
Proof

Follows at once from the definition of PG(K), RE, the structure of PGI(M) and the above mentioned convention.

Definition

\( \text{FS}^{\text{III}} \) is the set of all observable signal sequences that are generated by PGI(M).

Remark

For the sake of completeness, the START signal will be, in the present discussion, interpreted as the reception of the signal λ!

Theorem 4-2 \( \text{FS}^{\text{III}} = \text{FS}^{\text{I}} \)

Proof

By induction on \( \ell g(\bar{x}) \) where \( \bar{x} \in S_\ast \).

Basis Step

Let \( \ell g(\bar{x}) = 0 \). Then the theorem is obviously true.

Induction Step

Assume the theorem to be true \( \forall \bar{x} \) such that \( \ell g(\bar{x}) = L \).

Let \( \bar{xy} \in \text{FS}^{\text{III}} \) such that \( \ell g(\bar{xy}) = L+1 \).
Furthermore, let the symbol \( y \) appear in the sequence \( \bar{x} \), \( j+(J-1)K-1 \) times where

1. \( 1 \leq j \leq K \)
2. \( J \geq 1 \).

**Case 1** \( y \in \text{PI} \)

\( \bar{xy} \in \text{FS}^{\text{III}} \) \( \Rightarrow \) \( y' \) has appeared in the sequence \( \bar{x} \), \( j+(J-1)K-1 \) times (by the convention that we have adopted)

1. \( \exists \bar{x}'' \in \text{FS}^{\text{II}} \) such that \( \text{Ref}(\bar{x}'') = \bar{x} \), by the induction hypothesis
2. \( (y')^{j+(J-1)K-1} \in \bar{x}'' \), by lemma 3-2

\( \Rightarrow \) \( (\bar{x}'') \ y^{j+(J-1)K} \in \text{FS}^{\text{II}} \), since \( \text{Cov}(y^2) = (y')^1 \) by the argument used for proving lemma 3-1

\( \Rightarrow \bar{xy} \in \text{FS}^{\text{I}} \)

**Case 2** \( y \in \text{SI} \)

By using an argument identical to the one used for the previous case, with slight modifications, the theorem can be shown to be true for this case also.
Case 3 \( y \in 0 \)

\[ \overline{xy} \in \text{FS}^{III} \Rightarrow y^j \] can be generated for the \( j^{th} \) time, by lemma 4-11

\( \Rightarrow \) Every signal in the set \( \text{En}(y^j) \) has been received for the \( j^{th} \) time and every signal in the set \( \text{En}(y^j) \) has been received for the \( j-1^{th} \) time. \(*

1) \( \exists \overline{x}'' \in \text{FS}^{II} \) such that

\[ \text{Ref}(\overline{x}'') = \overline{x} \], by the induction hypothesis

2) \( \text{Cov}(y^{j+(J-1)K}) \subseteq \overline{x}'' \), by lemma 4-10 and the fact that the START signal is interpreted as \( \lambda \) when required

\( \Rightarrow \) \( \overline{xy} \in \text{FS}^{I} \).

Hence the theorem is proved and the faithfulness of the realization obtained by the proposed scheme is now established.

\(*\) The 0\( ^{th} \) reception of a signal in the set \( \text{En}(y^j) \) is to be interpreted as the reception of the START signal and the reception of this signal can be taken for granted.
CHAPTER 5

CONCLUSIONS

5-1

This chapter consists of a discussion of the areas of future research interest pointed to by the results that have been obtained so far in the study of asynchronous control structures.

5-2

Algebraic models for dynamic, information processing systems serve their purpose by,

1) Enabling one to formulate significant questions regarding the structure and behavior of the particular class of systems that have been modelled in a precise and rigorous fashion.

2) Making possible the application of the tools of mathematics to find elegant answers to such questions.

The algebraic models that have been proposed for the class of ACS's considered in this thesis are no exceptions to this rule.

The first problem (and the simplest) of interest concerning an ACS is the analysis problem, i.e., given the descriptions of the individual subsystems, the problem of
arriving at the description of the composite system obtained by interconnecting these subsystems. An important related problem endemic to ACS's (which may also be thought of as a part of the analysis problem), is the so called "dead-lock" problem.* i.e., the problem of deriving a set of necessary and sufficient conditions that the interconnection pattern between the individually-dead-lock-free subsystems must satisfy, so that the network of these subsystems is itself dead-lock-free. The C-D model appears to be a suitable tool for solving these two interrelated problems for marked-graph-describable ACS's. The work of Bruno and Altman leading to the concept of a "well-formed net" is certainly a step in the right direction for solving the dead-lock problem. However, a satisfactory, general solution to this problem that does not appeal to the particular implementation procedure used for realizing the subsystems of the network, is yet to appear.

A second problem of interest concerning an ACS is the equivalence problem, i.e., the problem of, firstly establishing whether it is possible to decide and secondly

* An ACS is said to be in dead-lock if all its input links are active and output links are inactive and the system is unable to generate any output signals.
(assuming that it is indeed decidable) providing the means for deciding whether two ACS's are equivalent or not, according to some well-defined criterion for equivalence. The poset-model, due to its well-defined and simple structure relative to which morphisms can be easily defined, appears to be a most suitable starting point for solving this problem for marked-graph-describable ACS's. Since a solution to the equivalence problem usually gives rise to a canonical representation, the poset-model is expected to be of considerable aid in the search for canonical representations for marked graphs.

A third problem of interest is the synthesis (realization) problem. A synthesis problem becomes interesting (and in fact meaningful) only when the constraints imposed on the realization are also stated. In the present context, some of the constraints that might be imposed are that the realization be modular, minimal in terms of the number of components used, cellular, etc. However, in the case of an ACS, independent of any of the constraints mentioned above, any solution to the realization problem, should also lead up to the demonstration of the fact that the proposed realization is speed-independent.

The technique of converting the problem of establishing the speed-independence of a realization to that of
showing that a corresponding marked graph (or a Petri-net in the more general case) has the appropriate properties, as has been done in this thesis [12], appears to be a useful one. A study of such techniques would establish the link (a sorely needed link in the author's opinion) between the research that has been carried out on Petri-nets and their realizations so far, and Muller's work on speed-independent circuits.

Finally, considering the question of algebraic models per se, an obvious (and important) extension of the problem that has been considered in this thesis would be to study the problem of formulating valid, useful and elegant algebraic models for the general class of ACS's, i.e., for the general class of Petri-nets. A mathematical representation of the important concept of conflict would be the major and crucial step in the solution to such a problem. The three classes of problems mentioned in the above discussion, when addressed to a general ACS, can be attacked in a systematic manner when an algebraic model for a general Petri-net has been formulated. As a final note, it seems to be safe to assert that most of the work on this problem of formulating an algebraic model for Petri-nets need to be concentrated on safe Petri-nets alone, at least for some time to come.
APPENDIX A

We present here a brief summary of those portions of Holt's work on marked graphs that are relevant to this thesis. The proofs for the theorems stated below may be found in [1]. Minor changes in the notations used by Holt to describe and analyze marked graphs have been made for the sake of convenience.

Definition

A marked graph $M_g$ is defined by

$$M_g = \langle (V, A, f_{in}, f_{o}), M \rangle$$

where

1) $V$ is an at most denumerably infinite set of vertices.
2) $A$ is an at most denumerably infinite set of arcs.
3) $f_{in}$ and $f_{o}$ are functions from $A$ to $V$.

$f_{in}(a)$ is called the input vertex of the arc $a$ and $f_{o}(a)$ is called the output vertex of the arc $a$. Also, if $a \in f_{in}^{-1}(v)$, then $a$ is called an output arc of $v$. If $a \in f_{o}^{-1}(v)$, then $a$ is called an input arc of $v$.

A vertex may have only a finite number of input and output arcs.
4) \( M \) is a function from \( A \) to the set of non-negative integers. \( M \) is called the marking of \( M_g \).

5) \( \langle V, A, f_{in}, f_{o} \rangle \) is the graph of \( M_g \).

**Remark**

A marker on an arc is indicated by placing a small circle on the arc.

A vertex whose all incoming arcs contain at least one marker may be fired by removing one marker from all its input arcs and adding one marker to all its output arcs.

**Definition**

a) A vertex \( x \) in a marked graph is **live** if there exists a sequence of vertex firing that contains \( x \).

b) A marked graph is live if all its vertices are live.

**Definition**

A set of arcs in a marked graph is said to be blank if each of the arcs in the set contain no markers.

**Theorem**

A vertex is live iff it is not contained in a blank circuit or on a blank path from a blank circuit.
Definition

a) An arc in a marked graph is said to be **safe** if the maximum number of markers that may ever appear (through all possible sequences of vertex firings) on the arc is 1.

b) A marked graph is said to be safe if all its arcs are safe.

Definition

A circuit in which exactly one of the arcs contained in the circuit has a marker on it is said to be a **basic circuit**.

Theorem

An arc in a **live** marked graph is safe iff it is contained in a basic circuit.
BIBLIOGRAPHY


10) Altman, Stanely M. and P. J. Denning (1970), "Decompositions of Control Networks", Record of the Project MAC Conference on Concurrent Systems and Parallel Computation, ACM.


12) Jump, Robert J. (1972), (Private Communications).