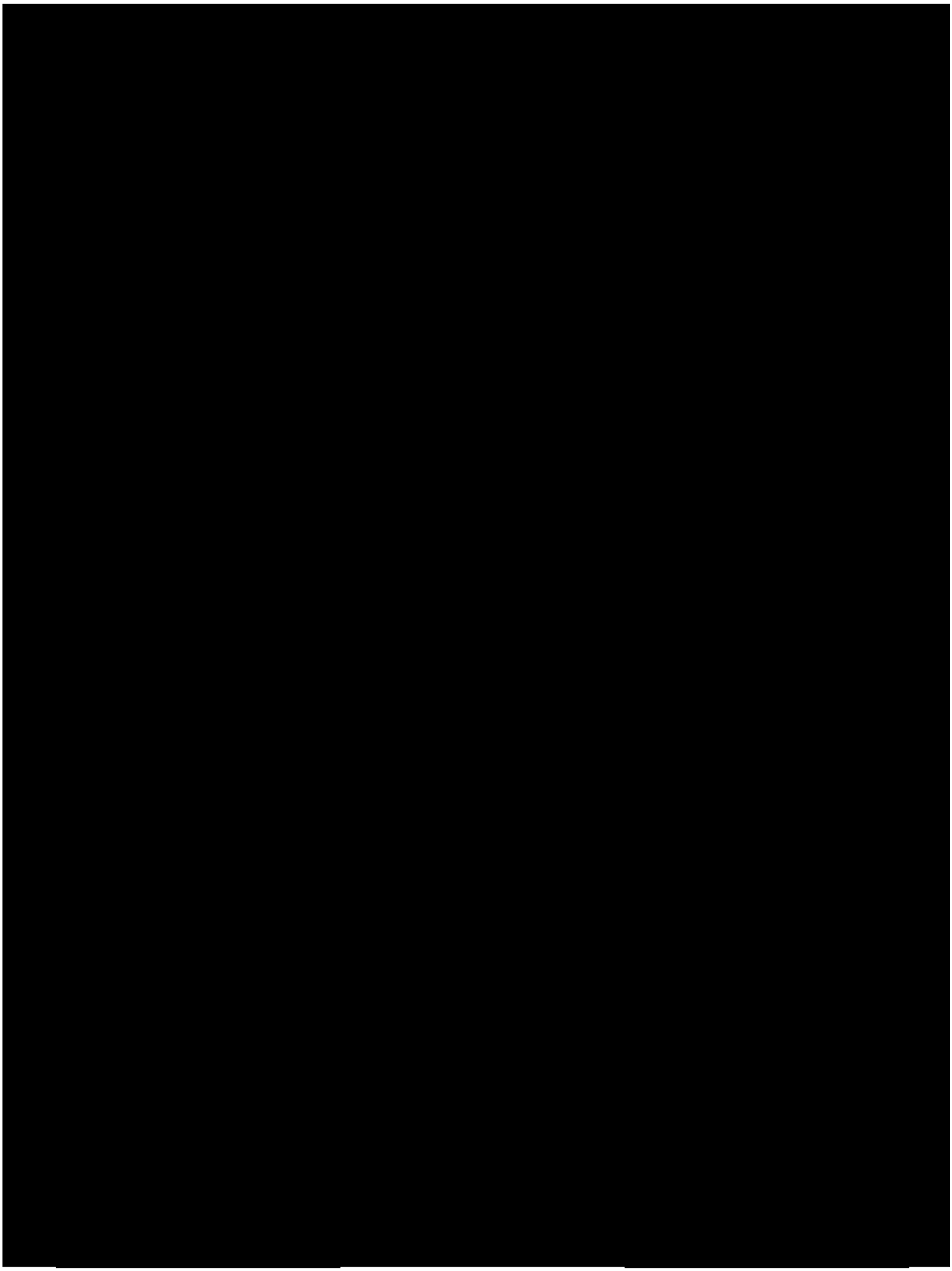


This blank page was inserted to preserve pagination.

ANALYSIS OF PRODUCTION SCHEMATA BY PETRI NETS

Michel Henri Théodore Hack

February 1972

PROJECT MAC

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Cambridge Massachusetts

ANALYSIS OF PRODUCTION SCHEMATA BY PETRI NETS

Abstract

Petri nets provide a powerful graphical tool for representing and analyzing complex concurrent systems. Properties such as hang-up freeness, determinacy, conflict, concurrency and dependency, can be represented and studied. The precise relationship between structural and behavioral properties, and between local and global properties is not well-understood for the most general class of Petri Nets. This thesis presents such results for a restricted class of Petri Nets called Free Choice Petri Nets, and for a corresponding class of Systems called Production Schemata. Results on structural constraints guaranteeing global operation, and decompositions of complex systems into meaningful parts, are also presented.

This report reproduces a thesis of the same title submitted to the Department of Electrical Engineering, Massachusetts Institute of Technology, in partial fulfillment of the requirements for the degree of Master of Science, February 1972.

ACKNOWLEDGEMENT

I wish to thank Professor Suhas Patil for his stimulating guidance throughout this research, and for introducing me to the exciting field of Petri Nets. I am very much indebted to Tolly Holt, to whom I owe many fundamental concepts in System Theory, and who communicated his enthusiasm to me. Many thanks also to Fred Commoner, with whom I spent many hours discussing theorems and proofs. I thank Jack Dennis for encouragement and helpful discussions, and for providing the research environment of the Computation Structures Group. My colleagues from this Group provided helpful comments, and I thank them for having had to endure my half-finished proofs.

Anne Rubin, who diligently typed this thesis, deserves my special gratitude.

I am grateful to Project MAC and Applied Data Research for financial assistance and the facilities they provided.

TABLE OF CONTENTS

		Page
Abstract		3
Acknowledgement		4
Table of Contents		5
Introduction and Previous Work		7
Part 1:	Description of Petri Nets and Production Schemata	9
Chapter 1:	Petri Nets	11
1.1	Definition	11
1.2	Liveness and Safeness	12
1.3	Syntactical Subclasses	13
1.4	Mathematical Properties: a first approach to the basic concepts	16
1.4.1	Overview	16
1.4.2	Liveness and Safeness in Free Choice Nets	16
Chapter 2:	Production Schemata	21
2.1	Flow of Control and Flow of Objects	21
2.2	Definition of Production Schemata: Conjunctive Elements	21
2.3	Definition of Production Schemata: Disjunctive Elements	25
2.4	Representing Production Schemata by Petri Nets	31
Part 2:	Mathematical Analysis of Free Choice Petri Nets	37
Chapter 3:	Formal Definitions and Notation	39
3.1	Petri Nets	39
3.2	Formal Definition of the Subclasses	42
3.3	Traps and Deadlocks	42

		Page
Chapter 4:	Necessary and Sufficient Conditions for Liveness and Safeness in Free Choice Petri Nets	43
4.1	Commoner's Liveness Theorem	43
4.1.1	Sufficiency Condition	43
4.1.2	Necessary Condition	49
4.2	Safeness; Live-and-Safe Markings	56
4.2.1	Definition of a Covering of a Petri Net	56
4.2.2	A Necessary Condition for a Live-and-Safe Marking in a Free Choice Net	57
4.2.3	Sufficiency Condition for Safeness in a Live Free Choice Net	59
4.3	The Live-and-Safeness Theorem	61
Chapter 5:	Decomposition of Free Choice Petri Nets	65
5.1	Well-Formedness in Free Choice Petri Nets	65
5.2	Duality, Reverse-duality; Open and Closed Consistent Subnets	65
5.3	Decomposition of a FC net Into a Covering of SCSM's	70
5.4	Decomposition of a FC Net Into a Covering of SCMG's	75
5.5	The Well-Formedness Theorem	81
5.6	Examples of Decompositions	83
Chapter 6:	Application of the Mathematical Results	91
Conclusion		109
Index		111
References		117

INTRODUCTION AND PREVIOUS WORK

The subject matter of this thesis is part of what can be called "Systemics," or System Theory, the science that analyzes and describes complex systems, patterns of interaction, communication between parts of a system, understanding of a system by understanding its parts and the interrelation of parts, and the like. Operational research has been applied to study such systems numerically, to compare operating strategies, to optimize. But our approach is structural, i.e. we are interested in the structural relations and dependencies of the system. Thus we have notions such as:

a) Operations A and B are concurrent, that is, either can precede the other, they may overlap in time, and which one of the above situations occurs is irrelevant. In some way, A and B are temporally independent.

b) Operation C must wait for both A and B to complete.

c) Operations D and E must both wait for C, but either one excludes the other, i.e. if D takes place, E cannot and vice versa. This is called a conflict situation, and related to it is the concept of decision (to resolve conflict) and branching.

d) Deadlock situation: A certain operation A must wait (depends on results of) operations B, but operation B must wait for A: The system hangs up, it is in a hang-up state, or deadlocked.

e) Unpredictability or non-determinacy: A certain operation depends on the results of either A or B, but A and B are concurrent: the final result may depend on whichever occurs first.

Petri Nets are a formal mathematical tool. They rely on a graphical representation of dependencies such as those described above, and, in a more general sense, are used to represent a system described by events whose occurrences depend on certain conditions and change those conditions. The notions of deadlock and unpredictability presented above correspond to the precisely defined properties of liveness and safeness of Petri Nets.

The mathematical analysis of Petri Nets in their full generality has not yet been very successful, but certain restricted classes are now well understood. This thesis shows important results for the class of Free Choice Petri Nets, a subclass of Petri nets, and solves the deadlock and unpredictability problem for a restricted class of systems called Production Schemata.

The concept of Systemics as a science is due to Holt (Information Systems Theory Project), who extended and applied the ideas of Petri. Petri Nets were introduced by Petri in his dissertation in 1962 [18] and modified to their present form by Holt in 1968 [10].

The idea of first studying a limited subclass of Petri Nets to obtain a better understanding of more general Petri Nets is due to Genrich [9], who introduced Marked Graphs to study concurrency.

Extensive mathematical results about a subclass of Petri Nets known as Marked Graphs have been published by Holt and Commoner [12]. In that publication, Marked Graphs have also been used to represent a subclass of Production Schemata, namely those without decision branches or conflicts.

Research on this thesis was prompted by a comparison of Rodriguez's Parallel Program Graphs [19] and Marked Graphs. Both formalisms express the same kind of determinism, but Rodriguez's Graphs allow for branching. Attempts to model branching by a method as similar in structure as possible to Marked Graphs led to the definition of Free Choice Petri Nets. The works of Karp and Miller [13], Muller and Bartky [14], Baer, Bovet, and Estrin [1], Slutz [21] were in different degrees relevant to research in the early stages of this thesis. In particular, Muller's concept of semimodularity is related to the behavior of safe Petri Nets, and the algorithms of Baer, Bovet and Estrin are of interest insofar as their "directed acyclic bilogic graphs" are structurally the same as acyclic Free Choice Petri Nets.

Among the references listed in this thesis are several other publications about Petri Nets. These include several applications of Petri Nets, notably Saint and Shapiro for representing algorithms [20], and Dennis for representing control structures in digital computers [6].

PART ONE

Description of Petri Nets and Production Schemata

CHAPTER 1
Petri Nets

1.1 Definition

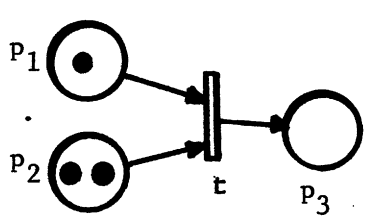
. A Petri Net is a directed bichromatic graph with an initial marking. The two distinguished types of vertices are called places and transitions. A marking is a function which associates with each place in the Petri Net a non-negative integer, called the token load of that place, or the number of tokens in it.

. A simulation of a Petri Net is a sequence of firings of transitions, only firable transitions may fire at any time, and a transition is firable if and only if all its immediate antecedent places (input places) have a positive, non-zero, load in the present marking. (A place with one or more tokens is marked, a place with no tokens is blank.) The firing of a transition changes the marking by decrementing the load of each input place by one and by incrementing the load of each immediate successor place (output place) by one.

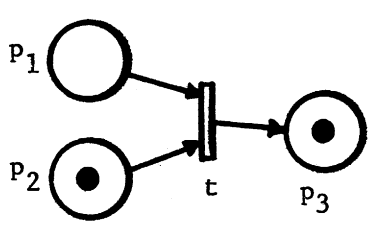
. A Marking M' is said to be reachable from marking M if there exists a firing sequence which transforms marking M into M'. The marking class of a Petri Net is the set of all markings reachable from the initial marking.

Graphically, we represent places by circles and transitions by bars. Dots in places represent the tokens of the marking.

Example:



before the firing of transition t.



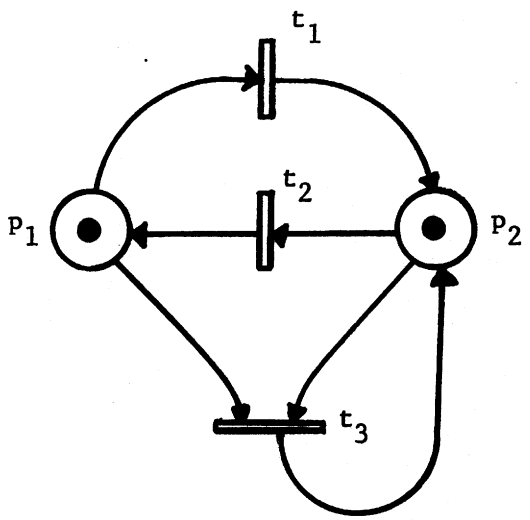
after the firing of transition t.

1.2 Liveness and Safeness

The most important properties of Petri Nets are liveness and safeness.

. A transition t is live at marking M if, for every marking M' that can be reached from M, there exists a firing sequence which fires t.

Example:



In this example, t_1 and t_2 are live, but t_3 is not live, because if we fire t_3 we reach a marking with only one token, and no firing sequence can possibly get two tokens back on the net, hence t_3 cannot be fired again.

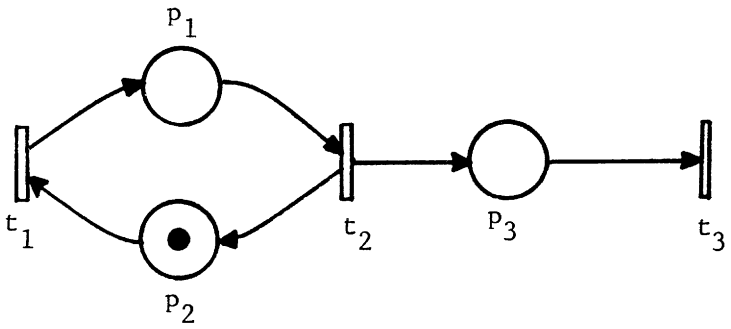
. If every transition in a Petri Net is live, the Petri Net is live.

An example of a live net is:

t_1 is live because it can fire at any time: it has no blank input place.
t_2 is live because, for any marking, $t_1 t_2$ is a firing sequence.

. A place p is safe at marking M if every marking M' that can be reached from M has at most one token on p.

Example:



p_1 and p_2 are safe; p_3 is not.

. A Petri Net is safe if every place in the net is safe.

. A Petri Net is said to be live and safe, or LS, if it is both live and safe at the initial marking.

In a safe Petri Net, a place is either blank or has one token. We can say that a place represents some condition which either holds or doesn't. A firing of a transition then terminates the holding of those conditions that enabled the transition, and begins the holding of other conditions: In this context, we say that an event, represented by the transition, occurred.

1.3 Syntactical Subclasses

The structure of Petri Nets in full generality, as defined above, is very rich, and it appears difficult to fully understand the relationships between the structure of the net (properties such as strongly connected, for example) and the behavior of the net (liveness or safeness, for example). Hence we approach the problem by analyzing first certain restricted subclasses of Petri nets.

Definitions

Presently we distinguish the following subclasses:

- State Machines (SM)
- Marked Graphs (MG)
- Free Choice Petri Nets (FC)
- Simple Petri Nets (SN)

} proper subclasses

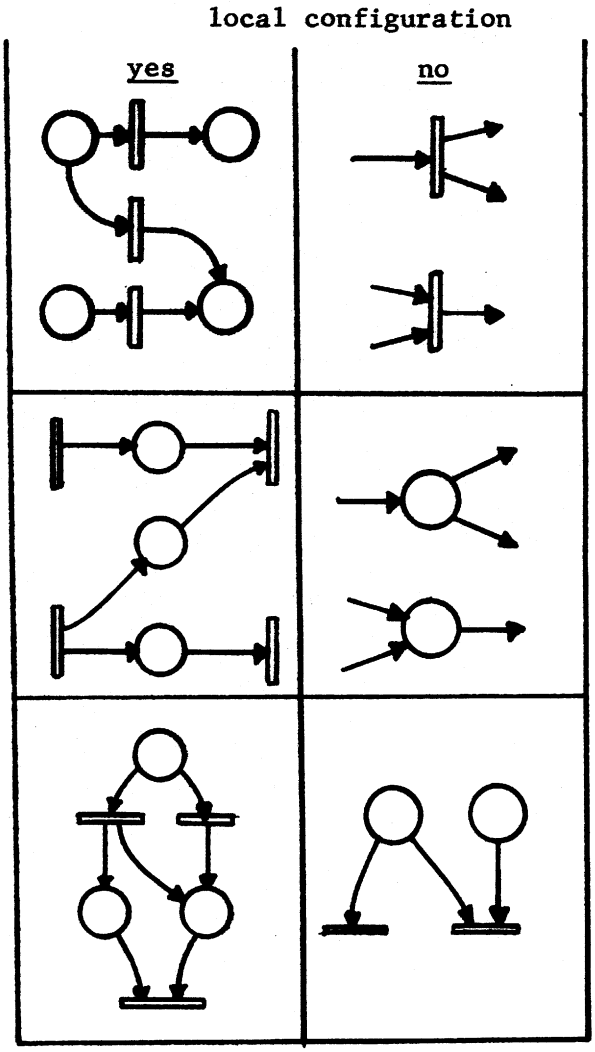
- Petri Nets (PN)

We say syntactical subclasses because of the fact that whether a given Petri Net belongs to a subclass or not is decided by the local structural configuration of the Net. In short, we have:

- SM: every transition has exactly one input place and exactly one output place.

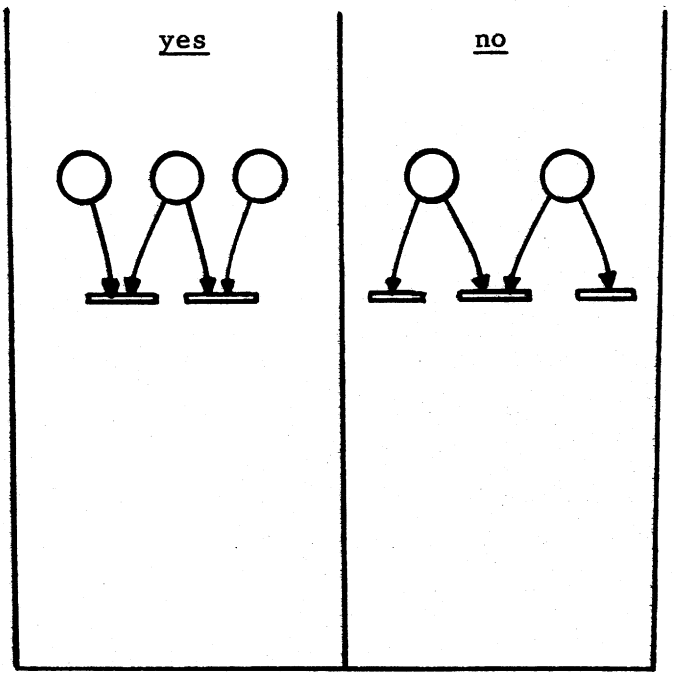
- MG: every place has exactly one input transition and exactly one output transition.

- FC: every arc from a place is either unique output of a place or unique input to a transition.

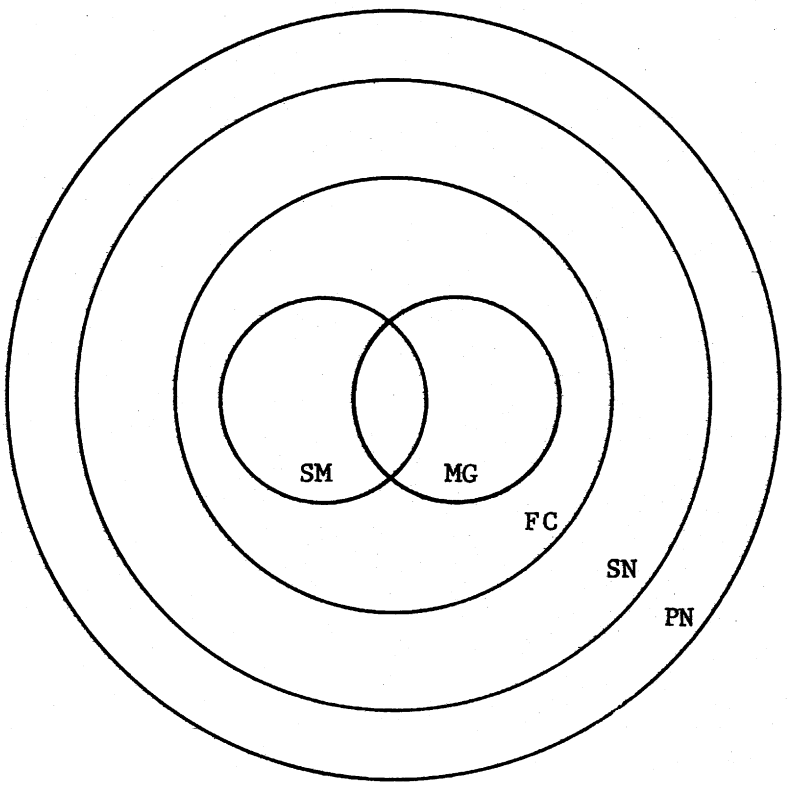


- SN: every transition has <u>at most one shared input</u> place.

- PN: no restriction.



The following figure shows the inclusion relations among the subclasses:



It can be seen that State Machines have the same structure as the familiar Finite State Automata or Sequential Machines, but uninterpreted in the sense that we do not associate input or output symbols to the transitions (state-transitions in Automata Theory language). A token in a place corresponds to the Sequential Machine being in the corresponding state, assuming there is only one token in the net.

1.4 Mathematical Properties: A first approach to the basic concepts

1.4.1 Overview

The mathematical properties of Petri Nets we are most interested in are the relations between liveness and safeness of the Net, or parts of it, and structural properties such as connectedness, covered by State Machines, decomposable into Marked Graphs.

Holt and Commoner have extensively studied the mathematics of State Machines and Marked Graphs [5, 12].

We shall focus our attention on Free Choice Nets. The most important result is a Theorem that states necessary and sufficient conditions for the existence of a live and safe marking in a Free Choice Petri Net.

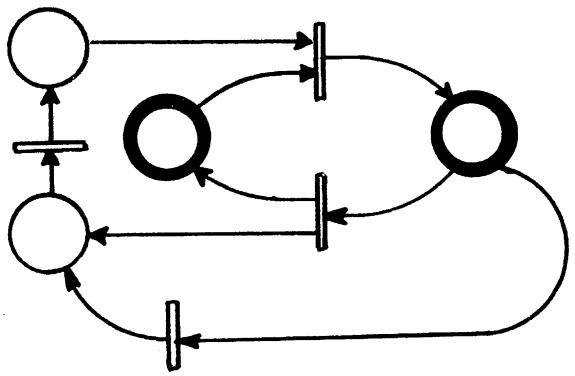
To date, Free Choice Petri Nets are the largest class of Petri Nets for which such necessary and sufficient conditions are known.

1.4.2 Liveness and Safeness in Free Choice Petri Nets

Important preliminary contributions to this topic are due to Fred Commoner, and include the definition of Traps and Deadlocks, as well as a Necessary and Sufficient Condition for Liveness of Free Choice Petri Nets.

. A Deadlock is a set of places in a Petri Net such that every transition which puts a token on some place in the set requires at least one token from some place in the set. This implies that if a deadlock is blank (i.e. contains no tokens), it will remain blank for every possible firing sequence. This is intuitively bad for liveness, since every transition having an input place in a blank deadlock will have no chance of firing.

Example:



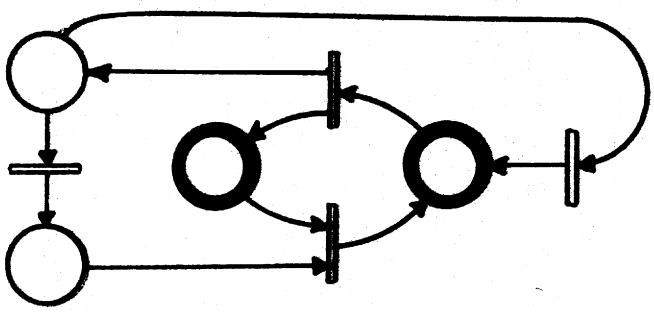
The bold face places form a deadlock.

Note that a deadlock in the Petri Net sense is a deadlock in the usual sense only if it is blank; potential deadlock might be a better name for the deadlocks defined above.

. A Trap is a set of places such that every transition which takes a token from the set puts at least one token back into the set. Hence once a Trap is marked, i.e. contains at least one token, it will always be marked, no matter what firing sequences take place.

Note that if a Deadlock contains a marked Trap, it will never become blank, and the threat to liveness described before does not exist: This is the "good" situation.

Example of a Trap (bold face)



Traps and Deadlocks are not exclusive: For example, every strongly connected Petri Net is both a Trap and a Deadlock.

Commoner has proved that a Free Choice Petri Net is live if and only if every Deadlock contains a marked Trap [4].

. Consistent Subnets: Open and Closed

A Subnet of a Petri Net is defined like a subgraph in Graph Theory [2], i.e. as a subset of vertices (places and/or transitions) and the arc relation restricted to the vertices of the subset.

Traps and Deadlocks are -- strictly speaking -- subnets by themselves, but such a collection of places without the transitions that are connected to them is not very meaningful by itself. Thus we introduce the concept of a Consistent Subnet.

. A Consistent Subnet of a given Petri Net is

either: a subnet consisting of a set of places and all transitions pointing to or from these places, called a Closed Consistent Subnet.

or: a subnet consisting of a set of transitions and all places pointing to or from these transitions, called an Open Consistent Subnet.

The distinction between Closed and Open comes from the fact that one type is connected to the rest of the net by sharing certain transitions, and the other by sharing certain places. We assume a place is more "open" than a transition hence an Open subnet has an "open" boundary of places, and a Closed subnet has a "closed" boundary of transitions.

Deadlocks and Traps can be conveniently viewed as Closed Consistent Subgraphs, because they are defined as a set of places. We shall henceforth take this point of view.

The union of Consistent Subnets is defined in the obvious way, so is the Covering of a Petri Net by a set of Consistent Subnets. Unless the Petri Net is very peculiar (having transitions without any input nor output places for example), if the union of the places of Closed Subnets is the set of all places of the Petri Net, the union of the Subnets is the whole Petri Net. In this sense we can speak of a Petri Net being covered by State Machines or by Marked Graphs.

Let a minimal Deadlock be a Deadlock that does not properly contain any non-empty deadlock.

We shall prove that a Free Choice Petri Net has a live and safe marking if and only if it is covered by strongly connected State Machines and every minimal Deadlock is a strongly connected State Machine.

This empty page was substituted for a blank page in the original document.

CHAPTER 2
Production Schemata

2.1 Flow of Control and Flow of Objects

In the introduction we described Systems in very general terms. We spoke of operations and dependencies of events on each other. One way to describe dependencies dynamically is to speak in terms of flow. We may, in general, speak of two sorts of flow: flow of control and flow of objects.

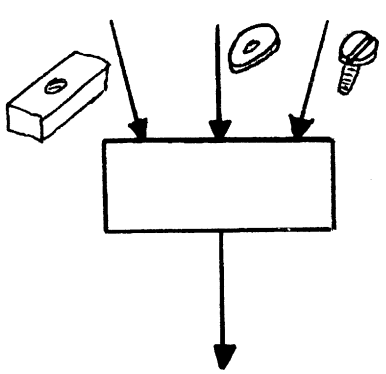
Flow of control often has a very complex structure because it describes situations such as gathering information in different parts of the system and directing one course of action instead of another. To model flow of control by Petri Nets, we need at least the structural complexity of Simple Nets.

Flow of objects, on the other hand, can be represented and analyzed by Free Choice Nets. We describe flow of objects in a System by Production Schemata.

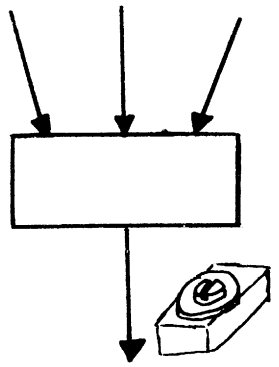
2.2 Definition of Production Schemata: Conjunctive Elements

A Production Schema is a model for representing the flow of objects in a System. It describes operations on objects, and branching or merging of flow.

An assembly operation takes as inputs all the parts needed to assemble an object: The operation takes place only when all inputs have arrived; there is one path of flow per object.

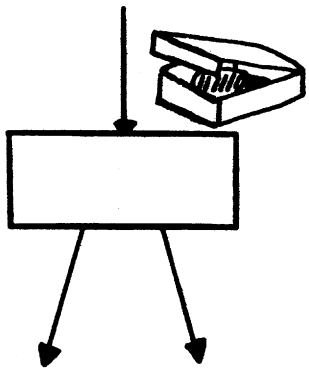


before assembly

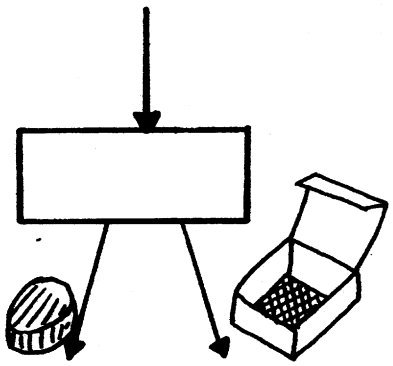


after assembly

We also have a disassembly operation:

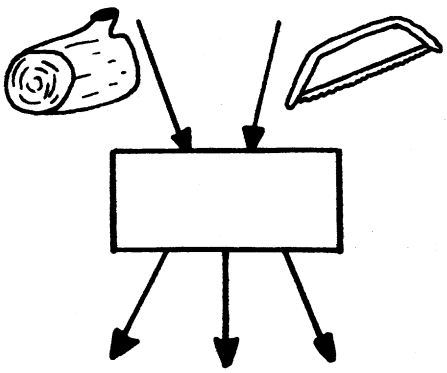


before disassembly

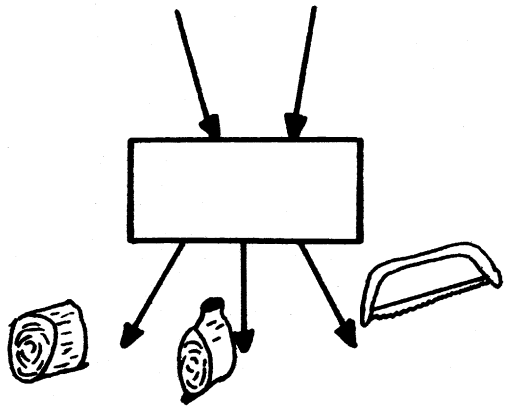


after disassembly

In a more general sense, we have operations with several inputs and several outputs:



before



after

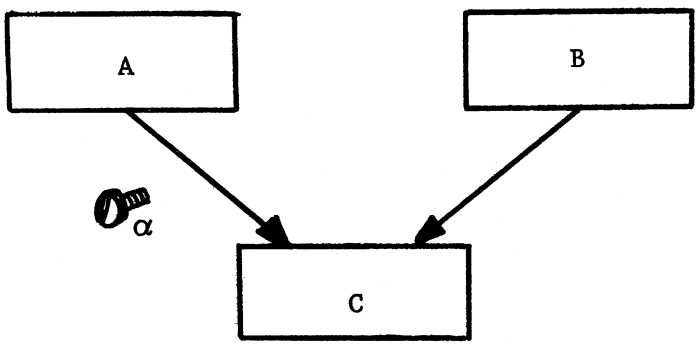
These operations are described by conjunctive nodes because input flow and output flow are conjunctive: all input objects are needed to initiate the operation, and all output objects are produced each time the operation terminates.

Before we present more elements of Production Schemata, we shall emphasize two points: Timing, and accumulation of several objects in one place (input arc to an operation, for example).

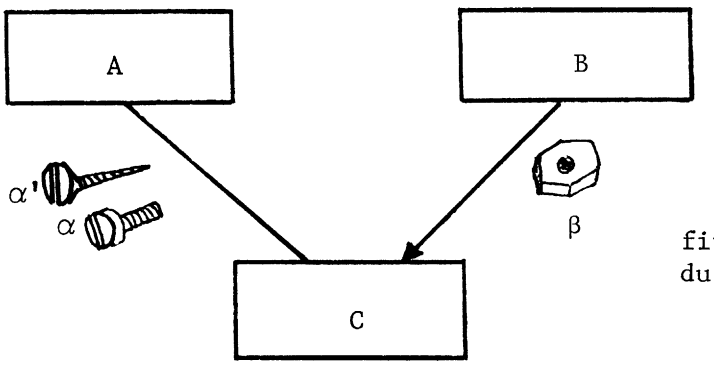
Timing, in the usual sense of a description of the upper and lower bounds of delays, is a "bad word" in our context. We wish to represent all constraints structurally in our model. This means that if a certain system contains timing constraints, these will show up as structural constraints in the model which is itself strictly asynchronous. This is possible because we can model the flow of metered time by a "clock," a certain event which happens, by definition, every t seconds. The structure of the model is then such that if a certain event must (by specification) occur between, say, the a^{th} tick (since some time origin) and the b^{th} tick, that event depends (structurally) on the a^{th} tick, but the b^{th} tick depends on it. This way we can model situations like: "If item A has not been used after four hours, discard it."

Had we chosen a synchronous model, with metered time, it would be very difficult indeed to represent asynchronous systems, and the cause and effect relationship among events. Moreover, it seems that even in the case of synchronous systems, we gain more insight into the system by explicitly representing all constraints on the events in the system in an asynchronous model.

Now consider the following situation:



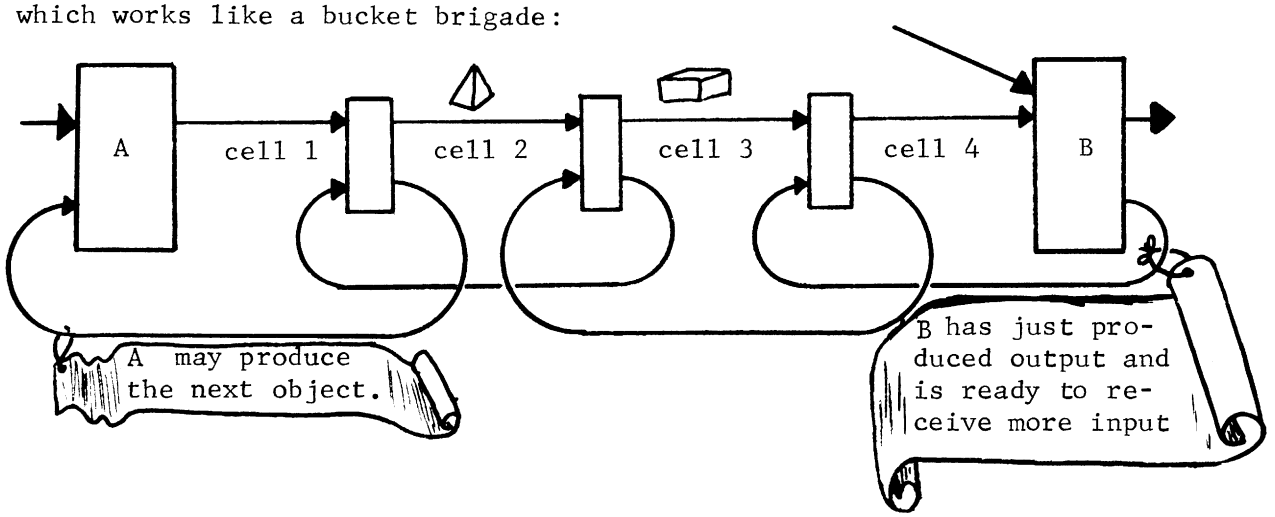
Operation C gets its inputs from A and B. One object, α, has arrived from A, and C is now waiting for an object from B to proceed. But before this happens, A produces another object, α'.



finally, B produces object β.

Now, should C use α and β, or α' and β? If α and α' were undistinguishable it would not matter, but we intend to keep our model as general and uninterpreted as possible and must assume that all objects are distinguishable (cf "free interpretation" in program schemata [15]). We could require the link to preserve order (and hence mate β to α), but this can be modeled independently by a pipeline, which we shall introduce below. We therefore let this situation be undesirable, i.e. express a malfunction of the system, and shall analyze it as such. It reminds us of course of unsafeness in Petri Nets, and, in most systems, can be thought of as a malfunction leading to unpredictability and non-determinacy.

To represent a system where one part may produce at times more objects than are consumed by another, we need a buffer, or pipeline, and usually the capacity is specified; in particular we do not expect infinite queues. Then, a pipeline that can hold up to, say, 4 items and deliver them in order, can be represented by the following arrangement, which works like a bucket brigade:

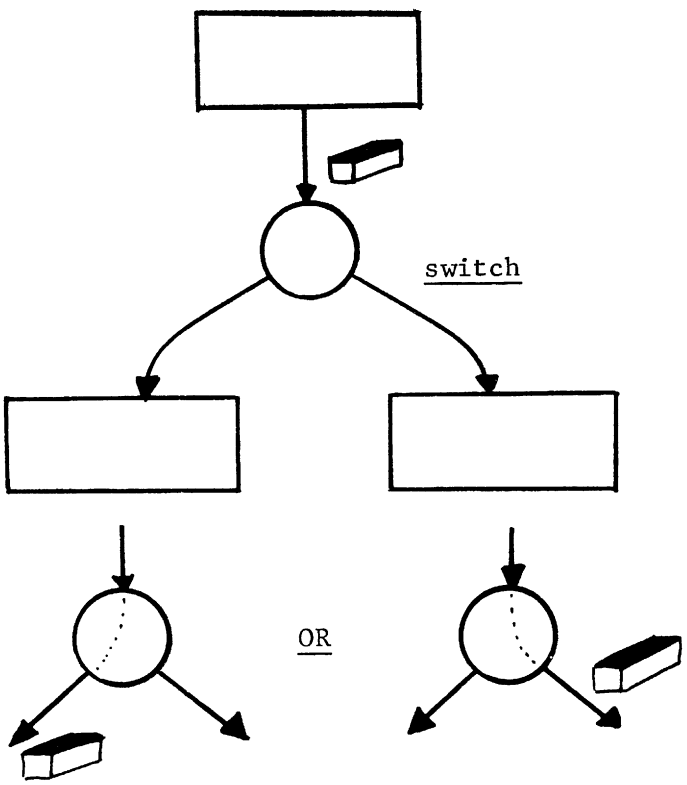


We have 4 cells. Each cell either contains an object on the top link, or a message on the bottom link. The message says actually two things, depending on the point of view: "Ready to receive another object," and "Previous object has just been delivered." These messages constitute what Holt calls "backflow" in Marked Graph models for Production Facilities [12]. It is of course debatable whether we should consider this flow of messages as flow of objects rather than flow of control; but in some systems all objects might effectively be messages, and, more importantly, we may consider a warehouse as an operation taking as input an order form, and giving the requested object as output. This approach obviates the need for special input or output nodes: An input node is an operation which produces an object upon receiving a request, and an output node is an operation which produces a receipt, or acknowledgement, upon delivering to the "outside world" an object received as input. The important fact is that such messages are treated in a strictly local manner, just like other objects, and only the producing and receiving operations are "aware" of its existence, as opposed to control information described in 2.1.

So far, we have described exactly the same class of Systems as have been represented by Marked Graphs in "Events and Conditions," by Holt and Commoner [12]. We present next those elements which introduce decisions, switches, and permit the representation of a larger class of Systems.

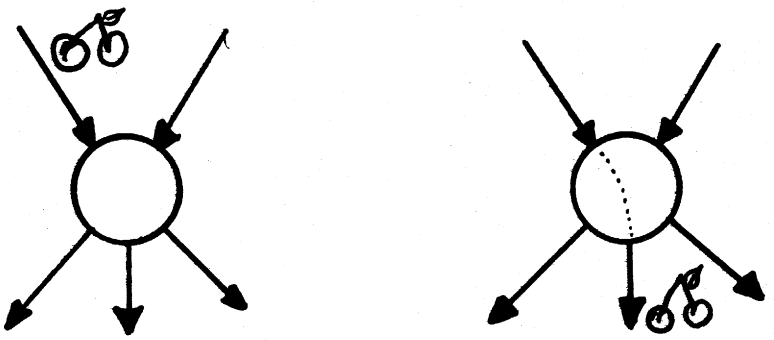
2.3 Definition of Production Schemata: Disjunctive Elements

If we want to represent a situation where an object produced by A flows either to B or to C, depending on circumstances (nature of the object, for instance), we need a new element whose outputs are disjunctive: It acts as a switch:



Also, if a certain operation gets its inputs from exactly one of several possible sources, we need an element with disjunctive input, sort of a reverse switch, or collector:

Of course, nothing a priori forbids us to consider a more general form of a switch:

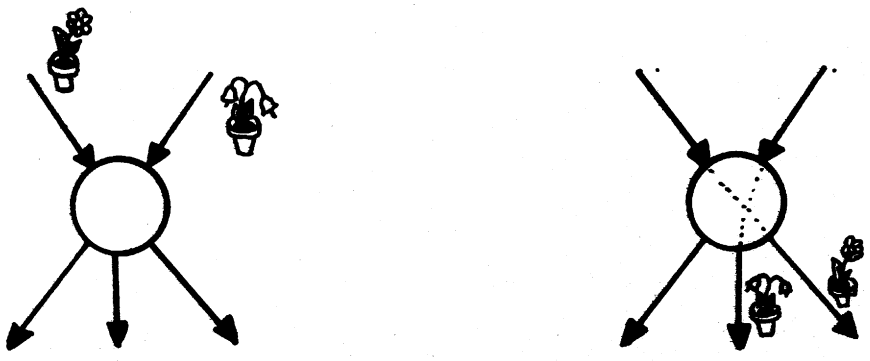


before after

These elements differ from operation elements by the fact that:

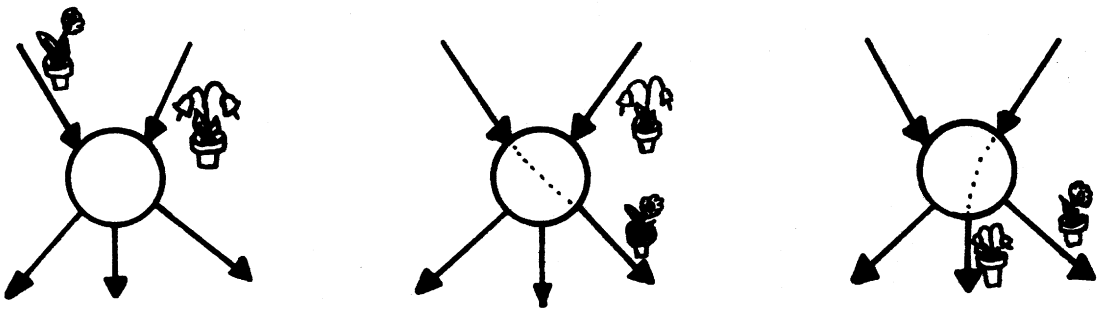
- they have disjunctive input and output,
- there is only one object flowing through at a time,
- the object flows through unchanged.

In particular, this means that the following transformation cannot take place in one step.



phase 1 step 1 (incorrect) phase 2

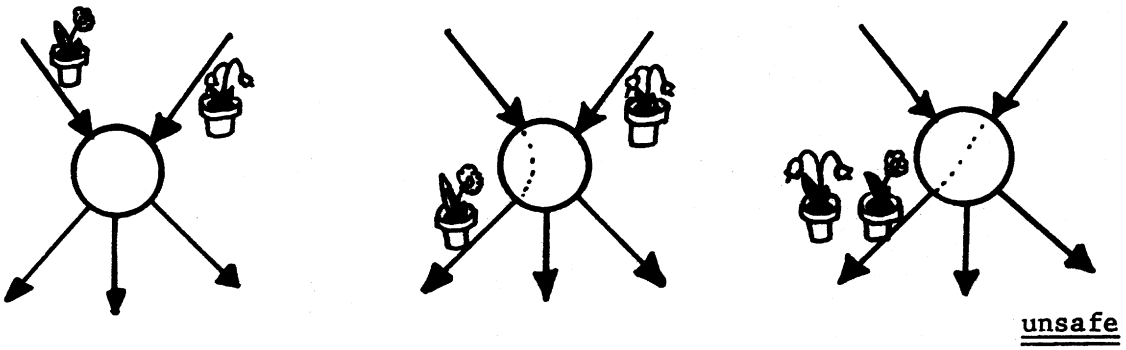
Instead, it involves two steps, which can occur in either order:



step 1 step 2

phase 1 phase 2 phase 3

But we could also have the following: (and in the free interpretation we must consider this along with all other possibilities).

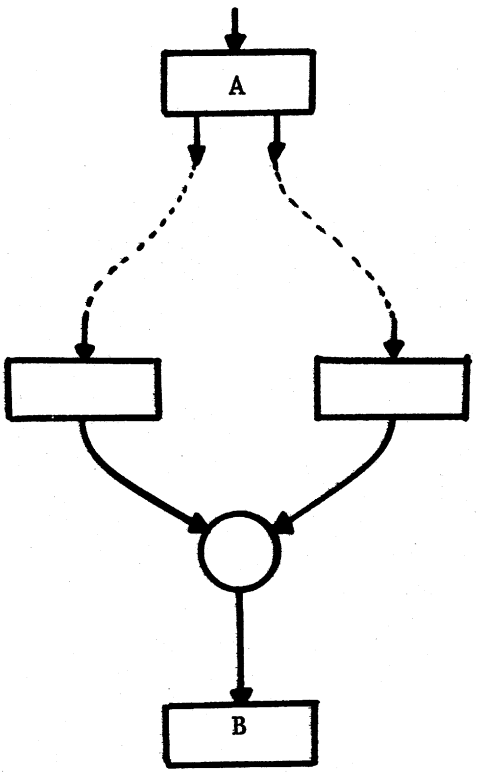


unsafe

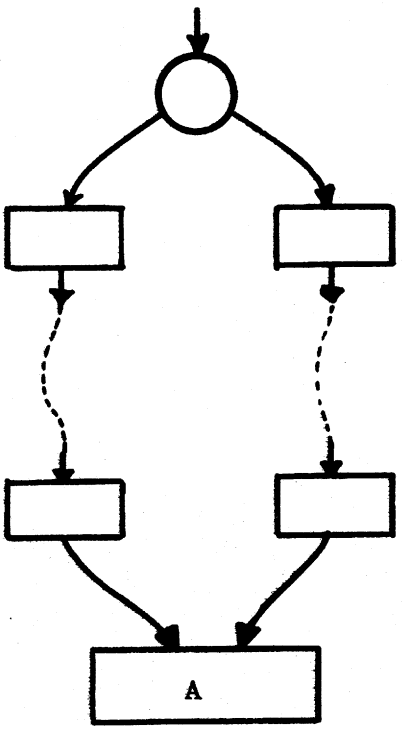
1 2 3

This leads to a situation we chose to consider a malfunction, possibly leading to non-determinacy. One of the objectives of this thesis is to guarantee structures such that if a collector element receives an object on one input, no object can possibly show up on any other input until the first object has been delivered to the next element following the collector.

We shall conclude this section by giving an example of a structure leading to deadlock, a structure leading to unsafeness, and an example of a structure without malfunctions.

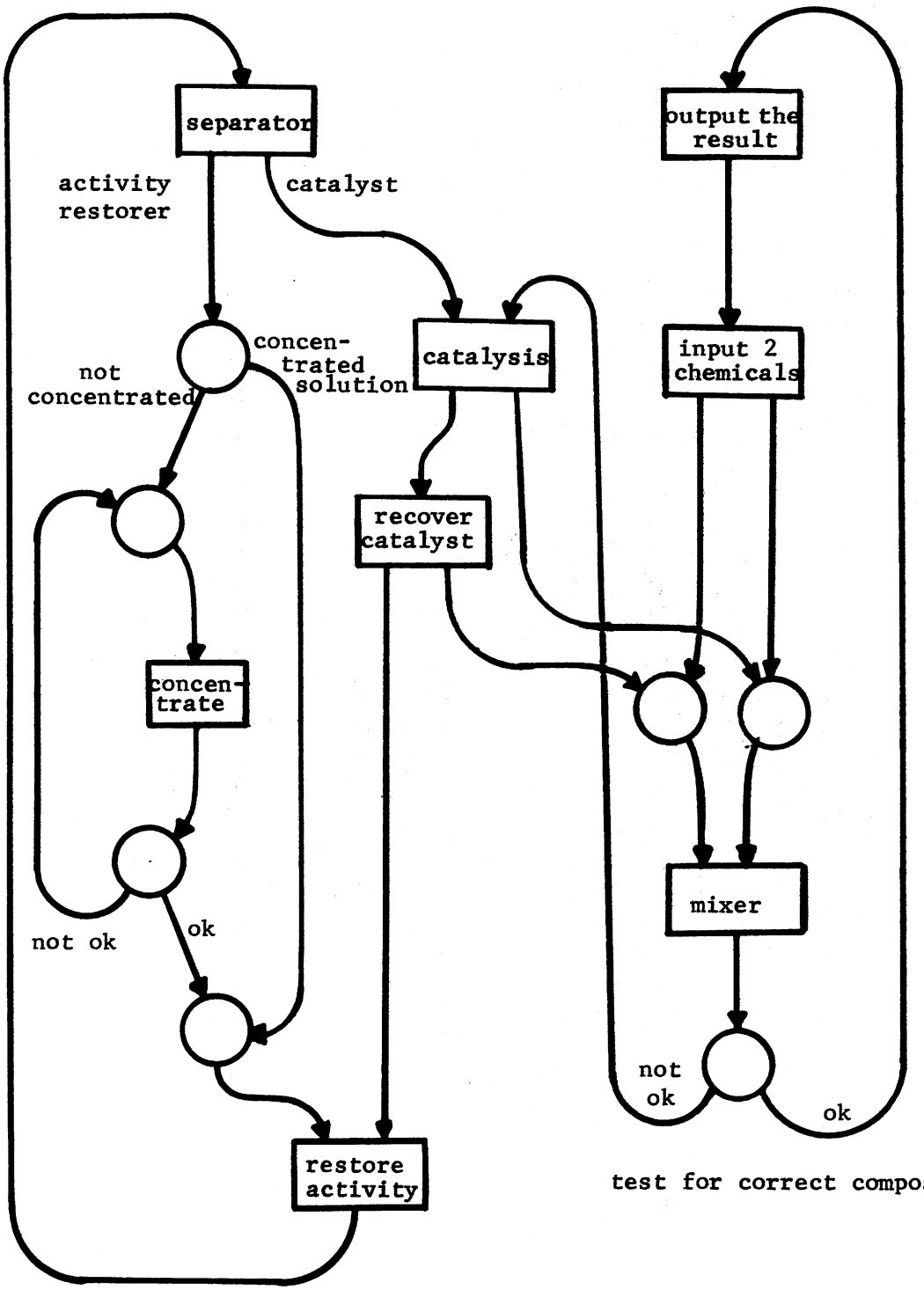


Example 1: Two paths, originating conjunctively and joining disjunctively, create possible unsafeness at the input to B.



Example 2: Two paths, originating disjunctively and joining conjunctively can lead to hang-up on A: If all objects are switched down the left path, the right input will never get an object, and A cannot operate.

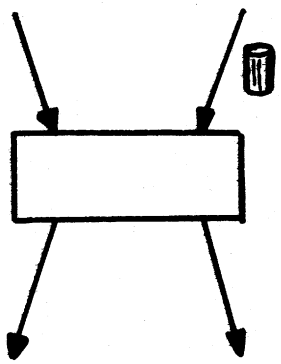
Example of a Well-Formed Production Schema.

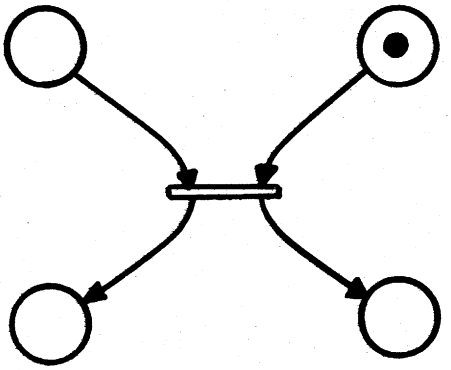


2.4 Representing Production Schemata by Petri Nets

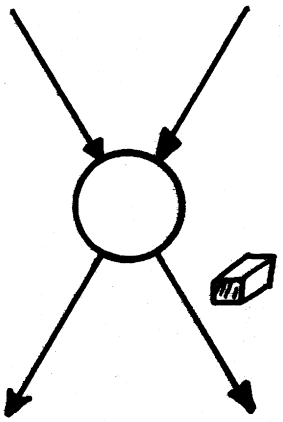
At this point, the reader has certainly noticed the similarity between objects and tokens, operations and transitions, links between elements and one-input-one-output places, and disjunctive elements and multiple arc places. The correspondence is straightforward:

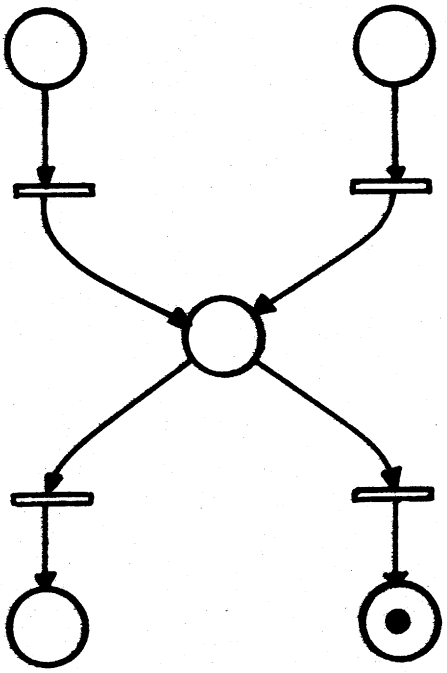
a)





b)





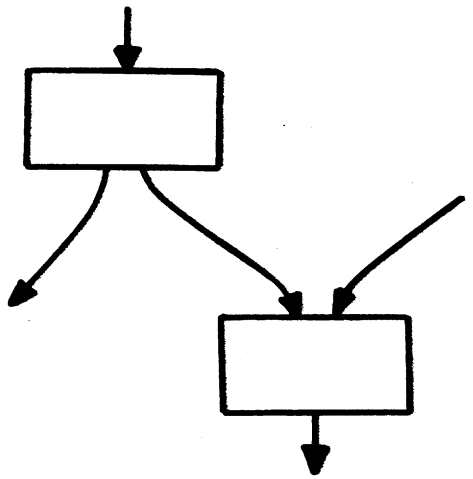
Production Schema

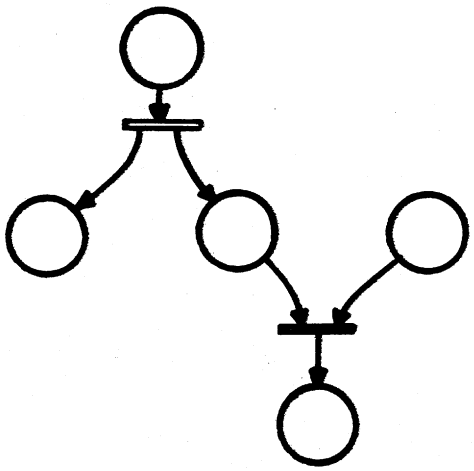
Petri Net

Production Schema

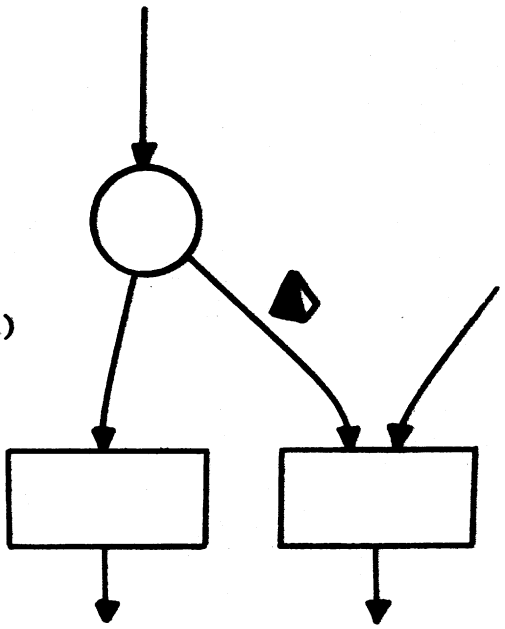
Petri Net

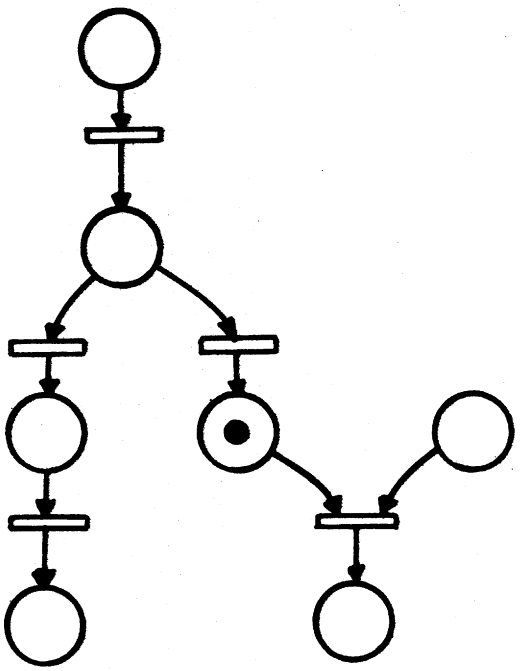
c)





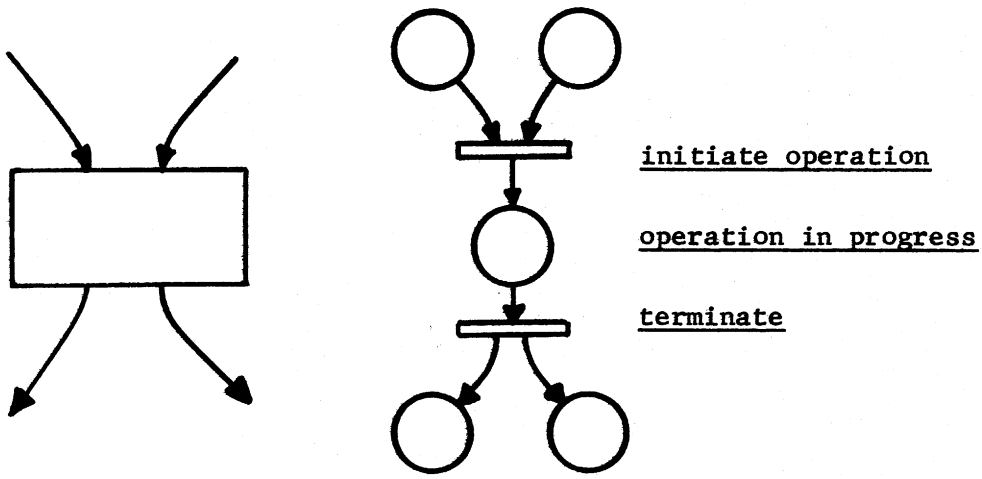
d)





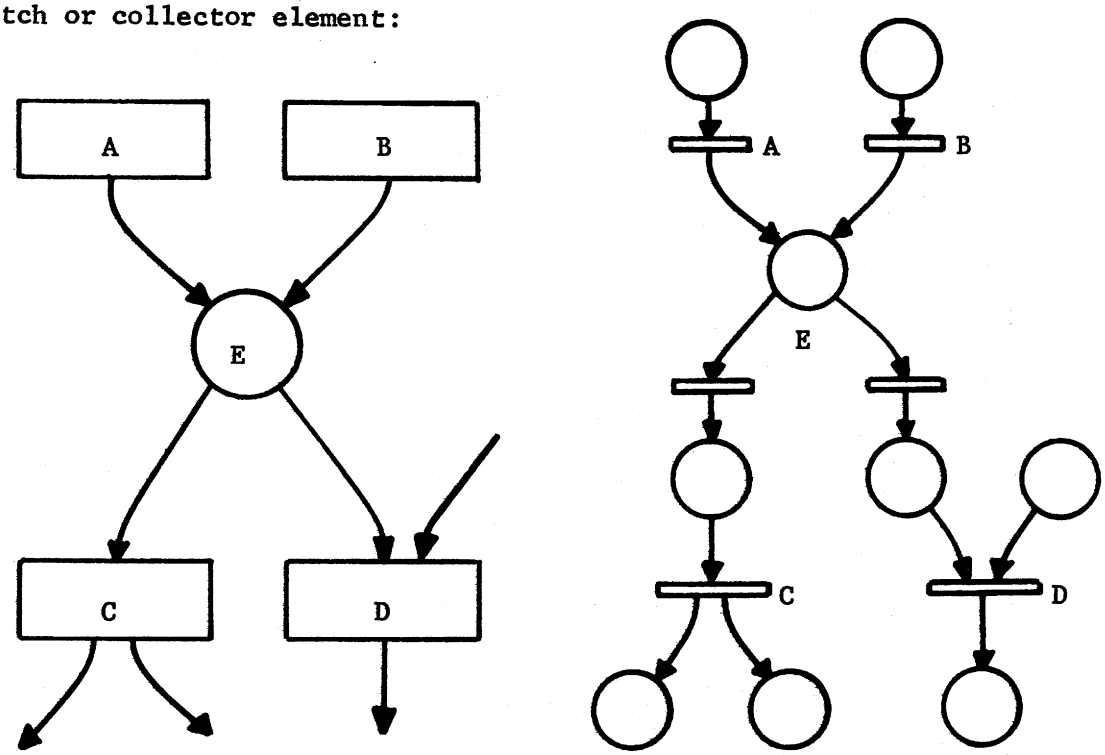
We note that, in Production Schemata, objects (tokens) are on the links, but in the Petri Net, tokens are always on places. This is especially illustrated in example d). There are two Petri Net firings associated with the switch (or collector) element, and there seems to be an intermediate step where the object is "inside" the switch. This is perfectly acceptable, and the switch or collector element could well

have been defined that way. We could also model an operation as follows, if we wish:

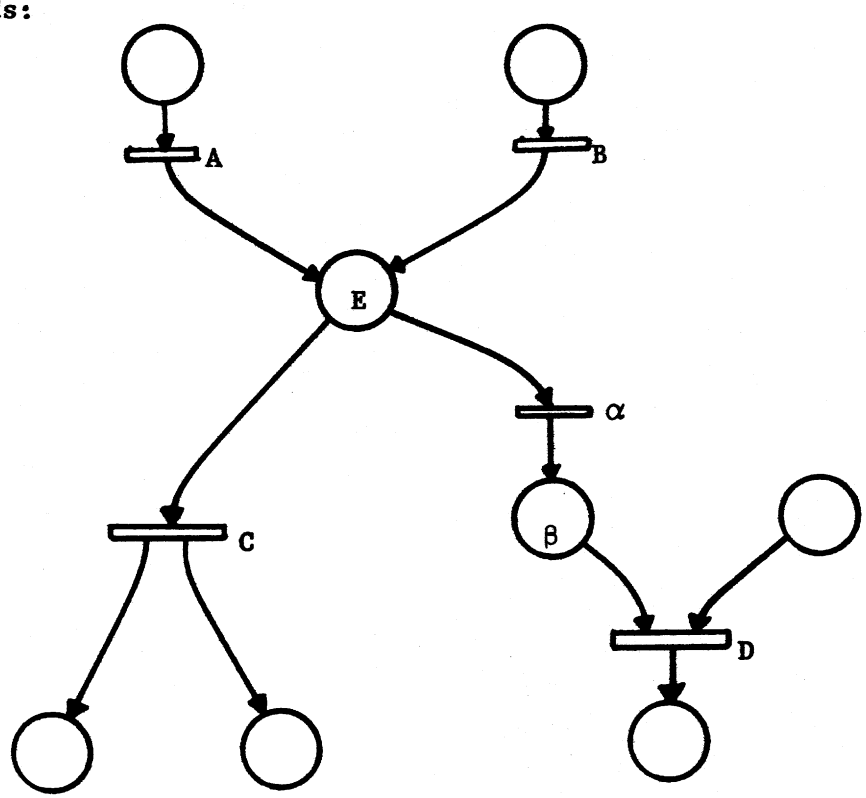


Semantically, this is even quite attractive, but it does not in any way change the structure of dependencies that we wish to analyze.

On occasion, we might wish to contract the representation of the switch or collector element:



We could even go one step further, though it may be questionable on semantic grounds:



But in no case can we suppress the "auxiliary" transition α and place β, because this would make the structure essentially different. As long as α and β are there, a token can be switched towards D and, after that, will have to wait for D to receive its other input, and fire. If, however, we remove α and β, the token could at any time be "stolen" or leaked away towards C; the switching decision would not be necessarily final as in the original net. This distinction is fundamental to the concept of Free Choice Petri Nets:

> Every Production Schema can be represented by a Free Choice Petri Net.

Conversely, every Free Choice Petri Net represents a Production Schema, if we allow contractions as discussed above.

The desirable properties for a Production Schema are:

- determinacy, predictability
- no hang-up states under any conditions of operation.

The first property has been associated with unsafeness in Petri Nets by definition of our formalism, the second property is clearly related to liveness in the representing Petri Net. We therefore define:

A Well-Formed Production Schema is a Production Schema represented by a Live and Safe Free Choice Petri Net.

This empty page was substituted for a blank page in the original document.

PART TWO

Mathematical Analysis of Free Choice Petri Nets

This empty page was substituted for a blank page in the original document.

CHAPTER 3

Formal Definitions and Notation

This chapter provides the formalism for the concepts introduced in Chapter 1.

3.1 Petri Nets

Definition: A Petri Net is a triple $\langle \Pi, \Sigma, \cdot \rangle$ where

Π is a non-empty set of places

Σ is a non-empty set of transitions

$\cdot$ is a relation; it corresponds to the arcs in the directed bichromatic graph; the set of vertices is $\Pi \cup \Sigma$. We have: $\cdot \subseteq (\Pi \times \Sigma) \cup (\Sigma \times \Pi)$

Notation: $\langle x, y \rangle \in \cdot$ is written as $x \cdot y$

$\{y \mid x \cdot y\}$ is written as $x^{\cdot}$

$\{y \mid y \cdot x\}$ is written as ${}^{\cdot}x$

We also apply the dot notation to designate the successor set of a set of places or transitions.

Example: $P \subseteq \Pi \qquad P^{\cdot} = \{x \mid \exists y \in P \text{ and } y \cdot x\}$

Def. A Marking is a function $M : \Pi \to \mathbb{N}$ (non-negative integers)

Def. A Firing is a partial function from markings to markings. There is a firing associated with every transition $t \in \Sigma$; t is said to be firable if its firing function is defined at the given marking M of the net, and the firing yields marking M'. We write this: $M[t\rangle M'$. The firing associated with $t \in \Sigma$ is such that:

$$\left.\begin{array}{ll} \forall p \in {}^{\cdot}t - t^{\cdot} & M'(p) = M(p) - 1 \\ \forall p \in t^{\cdot} - {}^{\cdot}t & M'(p) = M(p) + 1 \\ \forall p \in {}^{\cdot}t \cap t^{\cdot} & M'(p) = M(p) \end{array}\right\} \quad \text{defined only if: } \forall p \in {}^{\cdot}t \;\; M(p) > 0$$

Def. A firing sequence σ is a string over transition names and, as a function over markings, the composition of the firings of the transitions in the order they appear in the string.

We shall say $t \in \sigma$ if t is fired at least once in σ. We say that M leads to M' via σ, and write $M[\sigma\rangle M'$, or $M' = M[\sigma\rangle$ if σ, as a partial function, is defined for M. The set of firing sequences is denoted by Σ^*.

Def. The forward Marking Class $\vec{M}$ of a marking M is the set of markings which can be reached from M via some firing sequence:

$\vec{M} = \{M' \mid \exists\sigma \in \Sigma^* \text{ and } M[\sigma\rangle M'\}$

The concepts of liveness and safeness are defined as follows for Petri Nets:

Def. A transition t is live in a given marking if and only if for every marking in the marking class there exists a firing sequence which fires that transition.

$t \in \Sigma$ live at $M \overset{\Delta}{\Leftrightarrow} (\forall M' \in \vec{M})(\exists\sigma \in \Sigma^*)$ such that:
$M'[\sigma\rangle$ exists (i.e. σ, as a function, is defined at M') and $t \in \sigma$.

Def. A marking is live if and only if every transition is live at that marking.

Def. A place p is safe if and only if for every marking in the given marking class the load on p is not greater than one.

$p \in \Pi$ safe at $M \overset{\Delta}{\Leftrightarrow} \forall M' \in \vec{M} \quad M'(p) \leq 1$

Def. A marking is safe if and only if every place is safe at that marking.

Corollary: If a transition is live at marking M, it is live at any $M' \in \vec{M}$. If a place is safe at marking M, it is safe at any $M' \in \vec{M}$.

Def. A subnet of a Petri Net $\langle \Pi, \Sigma, \cdot \rangle$ is a Petri Net $\langle \Pi', \Sigma', o \rangle$ such that:

$$\Pi' \subseteq \Pi$$

$$\Sigma' \subseteq \Sigma$$

(o is the restriction of $\cdot$)

$$o = \cdot \cap (\Pi' \times \Sigma' \cup \Sigma' \times \Pi')$$

Short notation for a Petri Net $\langle \Pi, \Sigma, \cdot \rangle$: $\langle \Pi, \Sigma \rangle$

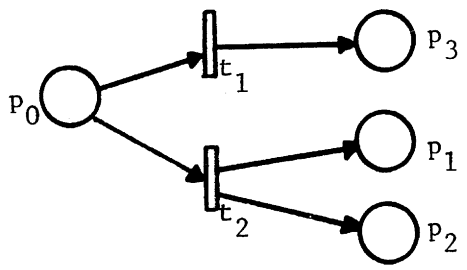
This can be used whenever $\cdot$ is clear from context. Thus, if we say that $\langle \Pi', \Sigma' \rangle$ is a subnet of $\langle \Pi, \Sigma \rangle$, it is understood that the arc relation for $\langle \Pi', \Sigma' \rangle$ is the restriction of the relation for $\langle \Pi, \Sigma \rangle$ to the set of vertices $\Pi' \cup \Sigma'$.

Example: $\langle \Pi, \Sigma, \cdot \rangle$ where:

$$\Pi = \{p_0, p_1, p_2, p_3\}$$

$$\Sigma = \{t_1, t_2\}$$

$$\cdot = \{\langle p_o, t_1 \rangle, \langle p_o, t_2 \rangle, \langle t_1, p_1 \rangle, \langle t_1, p_2 \rangle, \langle t_2, p_3 \rangle\}$$



expressed as: $p_o \cdot t_1$ $\quad t_1 \cdot p_1$

$p_o \cdot t_2$ $\quad$ etc.

also: $p^{\cdot} = \{t \mid p \cdot t\}$ $\quad p \in \Pi,\ t \in \Sigma$

${}^{\cdot}p = \{t \mid t \cdot p\}$

$t^{\cdot} = \{p \mid t \cdot p\}$

if $P \subseteq \Pi$, then $P^{\cdot} = \{t \mid \exists\, p \in P$ and $p \cdot t\}$

Hence, in example above: $p_o^{\cdot} = \{t_1, t_2\}$

$\{t_1, t_2\}^{\cdot} = \{p_1, p_2, p_3\}$

3.2 Formal Definition of the Subclasses

Definition: A **State Machine** (SM) is a Petri Net $\langle \Pi, \Sigma, \cdot \rangle$ such that: $\forall t \in \Sigma \; |{}^\bullet t| = |t^\bullet| = 1$. ($|A|$, where A is a set, is the cardinality of the set A). In other words, each transition has exactly one input place and one output place. (cf Chapter 1)

Definition: A **Marked Graph** (MG) is a Petri Net $\langle \Pi, \Sigma, \cdot \rangle$ such that: $\forall p \in \Pi: \; |{}^\bullet p| = |p^\bullet| = 1$

Definition: A **Free Choice Petri Net** (FC) is a Petri Net $\langle \Pi, \Sigma, \cdot \rangle$ such that: $(\forall p \in \Pi)(\forall t \in \Sigma)$: $p.t \Rightarrow p^\bullet = \{t\}$ or ${}^\bullet t = \{p\}$, i.e. an arc from a place p to a transition t either is the unique output arc of p or the unique input arc to t.

3.3 Traps and Deadlocks

In a Petri Net $\langle \Pi, \Sigma \rangle$,

Definition: A **Trap** is a subset of places $T \subseteq \Pi$ such that $T^\bullet \subseteq {}^\bullet T$, i.e. every transition having an input place in T must have an output place in T.

Definition: In a Petri Net $\langle \Pi, \Sigma \rangle$ a **Deadlock** is a subset of places $D \subseteq \Pi$ such that ${}^\bullet D \subseteq D^\bullet$, i.e. every transition having an output place in D must have an input place in D.

In a strongly connected Petri Net $\langle \Pi, \Sigma \rangle$, it is clear that we have ${}^\bullet\Pi = \Pi^\bullet = \Sigma$, hence it is both a trap and a deadlock.

Terminology: a set of places $P \subseteq \Pi$ in a Petri Net $\langle \Pi, \Sigma \rangle$ with marking M is said to be

- **blank**, if no place contains a token: $\forall p \in P: M(p) = 0$
- **marked**, if some place contains a token: $\exists p \in P: M(p) \geq 1$
- **empty**, if it is the empty set: $P = \emptyset$

CHAPTER 4
Necessary and Sufficient Conditions for Liveness and Safeness in Free Choice Petri Nets

4.1 Commoner's Liveness Theorem

Commoner solved the problem of deciding whether a given marking in a Free Choice Petri Net is live by proving that a necessary and sufficient condition for liveness is that every deadlock contain a marked trap. The proof we give here follows very closely the original proof of the theorem.

4.1.1 Sufficiency Condition

First, we prove the sufficiency condition, namely that if every deadlock contains a marked trap, then the marking is live. Lemma 1 establishes the influence of blank deadlocks on possible firings, and can be regarded as a mere technical preliminary to Lemma 2. Lemma 2 is phrased in a way as to directly lead to a proof by induction on the size of a subset of transitions. If the subset includes all transitions, Lemma 1 is applicable and provides the basis for the inductive proof. If the subset contains only one transition, the lemma expresses a sufficient liveness condition for that transition. Theorem 1, the sufficiency condition for liveness in Free Choice nets, follows immediately from Lemma 2.

Lemma 1: In a Petri Net $\langle \Pi, \Sigma \rangle$, let $M^o \subseteq \Pi$ be the set of blank places, and $M^+ \subseteq \Pi$ be the set of marked places ($\Pi = M^o \cup M^+$).

Let $W \subseteq \Sigma$ be a subset of transitions.

Then ${}^\bullet({}^\bullet W \cap M^o) \subseteq W \Rightarrow$ either: some t firable in W (i.e. $\exists t \in W: {}^\bullet t \subseteq M^+$)

or: $\exists$ blank deadlock D: $W \subseteq D^\bullet$

Proof: Assume no t firable in W: $\neg(\exists t \in W \text{ and } {}^\bullet t \subseteq M^+)$

i.e: $\forall t(t \notin W \text{ or } {}^\bullet t \not\subseteq M^+)$

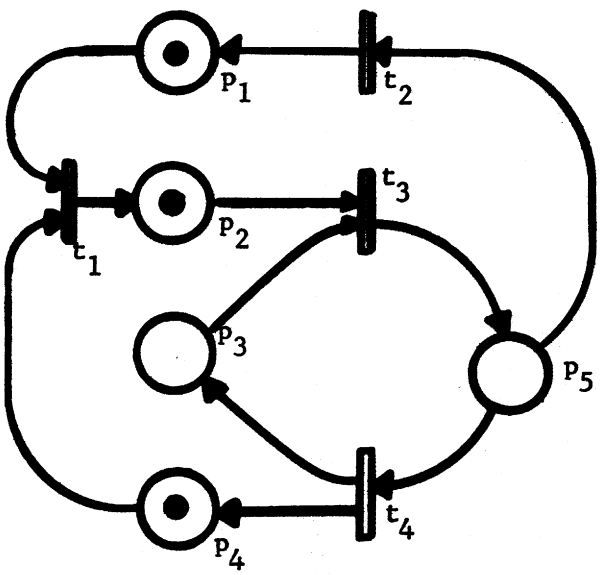
then we get $\forall t$: $t \in W \Rightarrow {}^\bullet t \cap M^o \neq \emptyset$

$\forall t$: $t \in W \Rightarrow t \in ({}^\bullet t \cap M^o)^\bullet$

hence $W \subseteq ({}^\bullet W \cap M^o)^\bullet$

But ${}^\bullet({}^\bullet W \cap M^o) \subseteq W$ by hypothesis: $({}^\bullet W \cap M^o)$ is a blank deadlock.

Example:



$W = \{t_2, t_3, t_4\}$

${}^\bullet({}^\bullet W \cap M^o) = \{t_3, t_4\}$

blank deadlock: $\{p_3, p_5\}$

Lemma 2: In a Free Choice Petri Net $\langle \Pi, \Sigma \rangle$, with marking M, let $W \subseteq \Sigma$ be a subset of transitions such that no firing sequence fires any transition in W. Then there exists a marking M' reachable from M such that there is a blank deadlock $D \subseteq M'^o$ and $W \subseteq D^\bullet$.

Proof: By induction on the size of $(\Sigma - W)$.

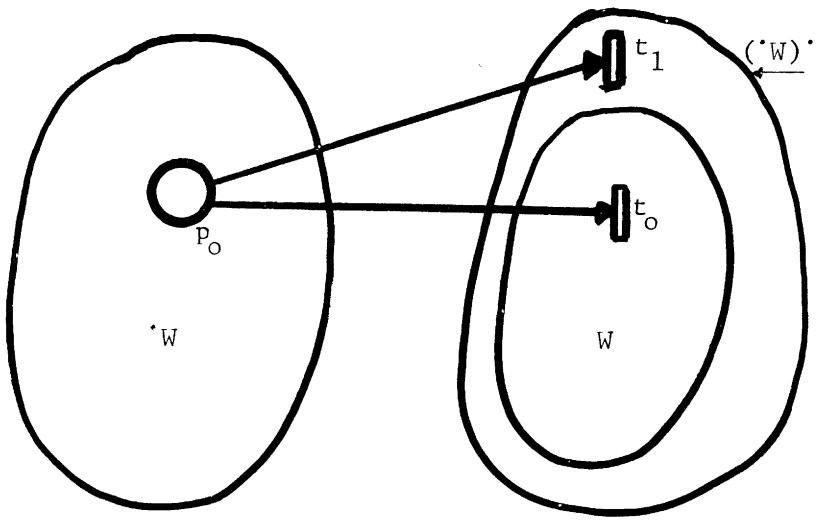
Basis: $|\Sigma - W| = 0$

Then $W = \Sigma$. Since Σ is the set of all transitions in the net, $\cdot(\cdot W \cap M^{o}) \subseteq W$ is trivially true. Therefore Lemma 1 applies directly to show that, if no transition can be fired in W, there must be a blank deadlock D such that $W \subseteq D^{\cdot}$.

Inductive Step: $|\Sigma - W| > 0$

Let the initial marking be $M_o = M$. We shall construct a firing sequence leading successively to the markings $M_1, M_2, \ldots M_i, \ldots M'$ such that, at M', we have a blank deadlock $D \subseteq M'^{o}$ and $W \subseteq D^{\cdot}$.

a) We shall show that no firing sequence fires any transition in $(\cdot W)^{\cdot}$. For suppose there is a transition $t_o \in W$ and a place $p_o \in \cdot t_o$ such that some transition $t_1 \in p_o^{\cdot}$ can be fired by some firing sequence. Since no firing sequence fires t_o by hypothesis, we must have $t_1 \in p_o^{\cdot} - W$:

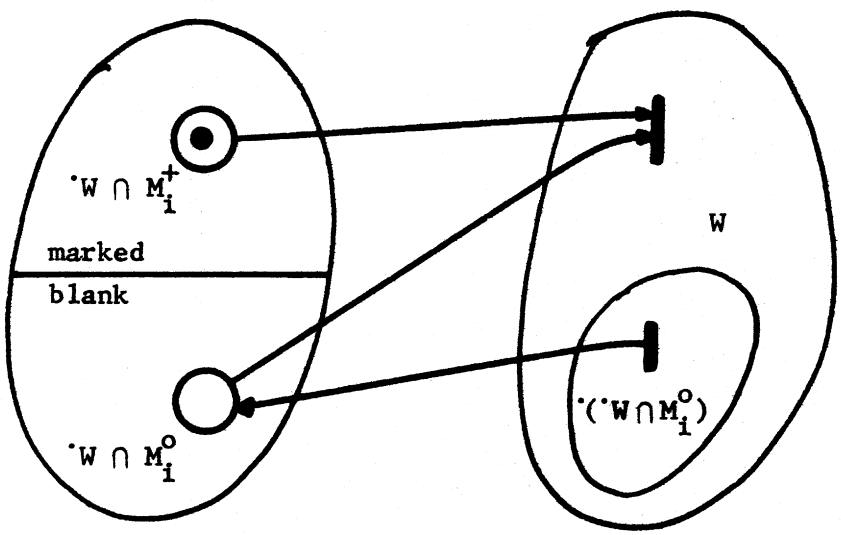


But then p_o has several output transitions, and by the Free Choice hypothesis, if t_1 can be fired t_o can also be fired, which contradicts the hypothesis that no firing sequence fires any transition in W:

- No firing sequence fires any transition in $({}^{\bullet}W)^{\bullet}$.

b) Let the present marking be M_i. There are two cases:

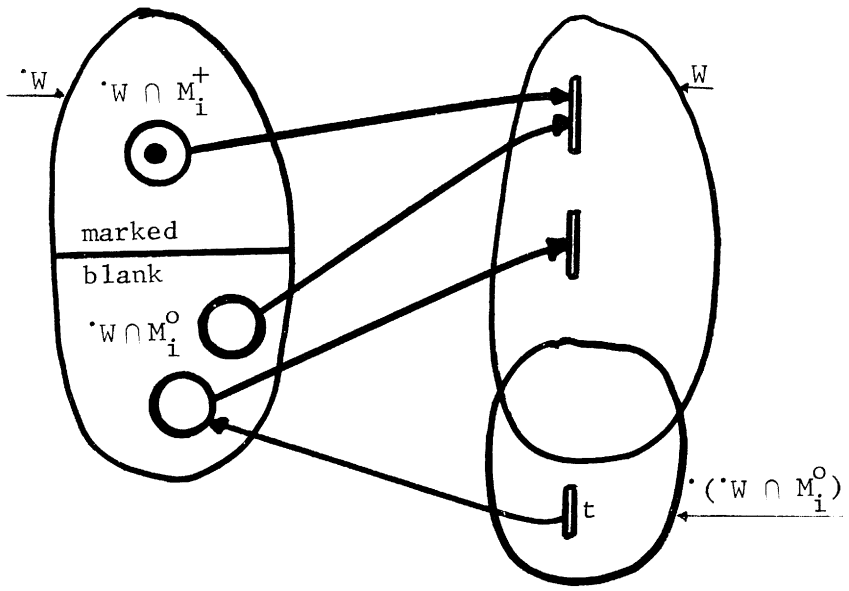
Case 1: ${}^{\bullet}({}^{\bullet}W \cap M_i^o) \subseteq W$



In this case Lemma 1 applies. Since, by hypothesis, no firing sequence fires in W, there must be a blank deadlock $D = {}^{\bullet}({}^{\bullet}W \cap M_i^o)$ such that $W \subseteq D^{\bullet}$, which proves Lemma 2 with $M' = M_i$.

Case 2: $\dot{}(\dot{}W \cap M_i^o) \not\subseteq W$

Then there exists a transition $t \in \dot{}(\dot{}W \cap M_i^o) - W$:



There are two subcases:

Case 2.1: No firing sequence fires t.

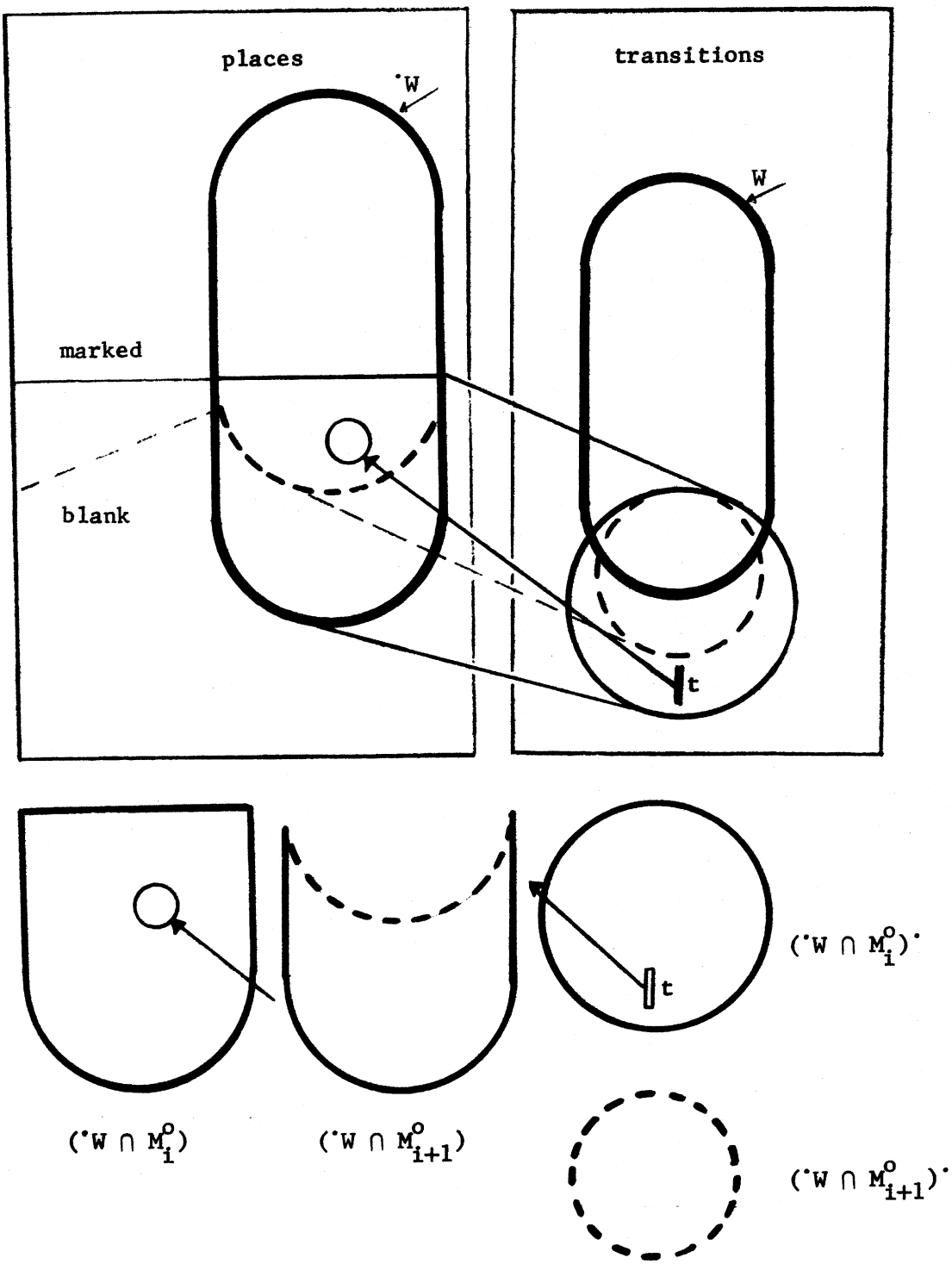
Then, let $W' = W \cup \{t\}$. No firing sequence fires any transition in W'. But $|\Sigma - W'| = |\Sigma - W| - 1$: By the inductive hypothesis, there must exist a firing sequence σ leading to a marking $M' = M_i[\sigma\rangle$ such that there is a blank deadlock $D \subseteq M'^o$ and $W' \subseteq D^{\dot{}}$. Then, since $W \subseteq W'$, we have proved Lemma 2 with marking M' and deadlock D.

Case 2.2: There exists a firing sequence σ which fires t.

Let $M_{i+1} = M_i[\sigma\rangle$. Since, because of a), σ does not fire any transition in $(\dot{}W)^{\dot{}}$, we have: $(\dot{}W \cap M_i^+) \subseteq (\dot{}W \cap M_{i+1}^+)$. Then, since t fires into $\dot{}W \cap M_i^o$ and σ does not fire in $(\dot{}W)^{\dot{}}$, we have: $|\dot{}W \cap M_{i+1}^o| < |\dot{}W \cap M_i^o|$.

We repeat the argument at marking M_{i+1}. Since, each time we have to apply case 2.2, the size of $(\dot{}W \cap M_i^o)$ decreases, we must eventually terminate at case 1 or case 2.1.

q.e.d.


places
transitions
•W
W
marked
blank
t
t
(•W ∩ M_i^o)•
(•W ∩ M_i^o)
(•W ∩ M_{i+1}^o)
(•W ∩ M_{i+1}^o)•

Case 2.2

From Lemma 2 we deduce that if no deadlock can ever be blank, there must always be a firing sequence that fires any given transition. (Take W = {t}). But if a deadlock contains a marked trap, since the trap will always contain at least one token, the deadlock cannot become blank:

Theorem 1. If in a FC net every deadlock contains a marked trap, then the net is live.
(Sufficient condition for liveness)

4.1.2 Necessary Condition.

We want to prove that in a live FC net, every deadlock must contain a marked trap, i.e. if the maximal trap in some deadlock is blank, there must exist a killing sequence, that is, a firing sequence leading to a marking where some transition can never be fired again.

Such a killing sequence can be obtained by making a certain choice ahead of time of the exit of multiple-output places: This selection is called an allocation. More precisely, we shall define an allocation on a set of places as a function which associates exactly one of the place's output transitions with the place. An allocation is circuit-free if there is no closed path through allocated transitions only.

Definition:

- An allocation A on a set of places S is a function:

$$A: \; S \rightarrow S^{\bullet}$$

such that $\forall p \in S: \; A(p) \in p^{\bullet}$

- An allocation A is circuit-free if there does not exist a path $p_o, t_o, p_1, t_1, \ldots p_n, t_n$ of places and transitions such that:

$$A(p_i) = t_i \qquad p_{i+1} \in t_i^{\bullet} \qquad p_o \in t_n^{\bullet}$$

- The set of allocated transitions is $\{t \mid \exists p \in S \text{ and } t = A(p)\}$, denoted by A(S)

- The set of excluded transitions is

$$\{t \in S^{\cdot} \mid \forall p \in {}^{\cdot}t \quad p \in S \Rightarrow t \neq A(p)\}$$

denoted by $\bar{A}(S)$

Note that $A(S) \cap \bar{A}(S) = \emptyset$

$$A(S) \cup \bar{A}(S) = S^{\cdot}$$

Hence $\bar{A}(S) = S^{\cdot} - A(S)$

The objective of the proof is to show that if some deadlock contains a blank trap, we can construct a killing sequence that does not put a token on the trap. First, we show the existence of an allocation that prevents the trap from getting a token, then we prove that this allocation permits us to kill the net.

Lemma 3: Given a set of places $Q \subseteq \Pi$ and the maximal trap T in Q, there is a circuit-free allocation $A: (Q - T) \to (Q - T)^{\cdot}$ of $Q - T$ that does not allocate into the trap, i.e.:

$$\forall p \in (Q - T) : A(p) \notin {}^{\cdot}T, \quad \text{or:} \quad A(Q - T) \cap {}^{\cdot}T = \emptyset$$

The maximal trap is the largest trap, or the union of all traps, in Q. It may be the empty trap, i.e. there may be no trap in Q.

Proof: By induction on $|Q - T|$.

- if $Q = T$, the empty allocation $\emptyset \to \emptyset$ satisfies the conditions trivially.

- assume $|Q - T| > 0$: $\left.\begin{array}{l} \exists p_o \in Q - T \\ \exists t_o \in p_o^{\cdot} \end{array}\right\} t_o^{\cdot} \cap Q = \emptyset$

since p_o is not in the maximal trap.

Hence, T is the maximal trap in $Q' = Q - \{p_o\}$. By inductive hypothesis there exists a circuit-free allocation A' of $Q' - T$ such that

$$A'(Q' - T) \cap {}^{\cdot}T = \emptyset$$

Let $A : (Q - T) \to (Q - T)^{\cdot}$ be the allocation whose restriction to $Q' - T$ is A', and which assigns t_o to p_o:

$$\forall p \in Q - T: \quad p \neq p_o \Rightarrow A(p) = A'(p)$$

$$p = p_o \Rightarrow A(p) = t_o$$

$$A(Q - T) = A'(Q' - T) \cup \{t_o\}$$

Since $A'(Q' - T) \cap {}^{\cdot}T = \emptyset$

and $t_o^{\cdot} \cap Q = \emptyset \Rightarrow t_o \notin {}^{\cdot}T$

we have $A(Q - T) \cap {}^{\cdot}T = \emptyset$

A does not allocate into T. Now suppose A is not circuit-free. Then, since A' is circuit-free, any circuit of A must contain the arc $p_o \cdot t_o$. But $t_o^{\cdot} \cap Q = \emptyset$: the arc $p_o \cdot t_o$ is not part of any circuit in Q, hence in $Q - T$.

Allocation A satisfies the conditions of Lemma 3.

q.e.d.

Lemma 4: If the maximal trap T in any deadlock D of a Free Choice net is blank, there exists a firing sequence which leads to a marking where no transition of $D^{\cdot}$ is live.

Proof: Let $A : (D - T) \to (D - T)^{\cdot}$ be a circuit-free allocation of $D - T$ such that $A(D - T) \cap {}^{\cdot}T = \emptyset$. Such an allocation exists by Lemma 3.

Let us call a firing sequence that does not fire any excluded transitions an A-sequence:

$$\sigma \text{ is an A-sequence} \overset{\Delta}{\Leftrightarrow} \forall t \in \sigma \qquad t \notin \bar{A}(D - T)$$

Then: - no A-sequence puts tokens on T: T remains blank (A does not allocate into T and D is deadlock).
- no A-sequence fires in $(D - T)^{\bullet} - A(D - T)$ [excluded transitions $\bar{A}(D - T)$]
- no A-sequence fires in $T^{\bullet}$ since T remains blank.

hence: no A-sequence fires in $T^{\bullet} \cup [(D - T)^{\bullet} - A(D - T)]$.
Let B be a set of places in $D - T$: $B \subseteq D - T$.

claim: The only firings in an A-sequence that put tokens on B are those that fire in $A(D - T)$:

For B to receive a token, the sequence must fire in ${}^{\bullet}B$. But $B \subseteq D$ and ${}^{\bullet}D \subseteq D^{\bullet}$, hence ${}^{\bullet}B \subseteq D^{\bullet}$. Since $T \subseteq D$ we have : $D^{\bullet} = (D - T)^{\bullet} \cup T^{\bullet}$.

Hence ${}^{\bullet}B \subseteq T^{\bullet} \cup (D - T)^{\bullet}$

But an A-sequence does not fire in $T^{\bullet} \cup ((D - T)^{\bullet} - A(D - T))$, hence any firing of an A-sequence in ${}^{\bullet}B$ must be in $A(D - T)$.

Now let $B_o = \{p \in D - T \mid \nexists\, p' \in D - T: p \in (A(p'))^{\bullet}\}$,

i.e. B_o is the set of "heads" of the circuit-free allocation. Since ${}^{\bullet}B_o \cap A(D - T) = \emptyset$ by construction, no A-sequence puts tokens on B_o , hence there is a bound on the number of times any A-sequence can fire in $B_o^{\bullet}$.

Now let $B_{i+1} = \{p \in D - T \mid \nexists\, p' \in (D - T) - B_i : p \in (A(p'))^{\bullet}\}$.

Assume $t \in {}^{\bullet}B_{i+1} \cap A(D - T)$

Then, we have: $\exists p \in B_{i+1} : p \in t^{\bullet}$

$\exists p' \in D - T : t = A(p')$

This implies $p \in (A(p'))^{\bullet}$

Hence, by the definition of B_{i+1}:

$$p' \notin (D - T) - B_i$$

This implies that every such t must be in B_i.

Hence: $\dot{B}_{i+1} \cap A(D - T) \subseteq A(B_i)$

We know that any A-sequence can fire only a bounded number of times in $B_o^{\cdot}$. Assume (inductive hypothesis) that any A-sequence can fire only a bounded number of times in $B_i^{\cdot}$. It follows from $\dot{B}_{i+1} \cap A(D - T) \subseteq A(B_i)$ that any A-sequence can put only a bounded number of tokens (cumulatively) on B_{i+1}, and hence can fire only a bounded number of times in $B_{i+1}^{\cdot}$.

Now, we show that $B_i \subseteq B_{i+1}$.

Assume $B_i \not\subseteq B_{i+1}$: There must be a place $p \in D - T$ such that:

$p \notin B_{i+1}$, i.e.: $\exists\, p_o \in (D - T) - B_i : p \in (A(p_o))^{\cdot}$

$p \in B_i$, i.e.: $\nexists\, p' \in (D - T) - B_{i-1} : p \in (A(p'))^{\cdot}$

Hence, we must have: $p_o \notin B_i$

$p_o \in B_{i-1}$

That is to say: $B_{i-1} \not\subseteq B_i$

By repeating the argument for decreasing values of i, we get:

$$B_i \not\subseteq B_{i+1} \Rightarrow B_o \not\subseteq B_1$$

But this leads to a contradiction: There must be a place $p \in D - T$ such that:

$p \notin B_1$, i.e.: $\exists\, p_o \in (D - T) - B_o : p \in (A(p_o))^{\cdot}$

$p \in B_o$, i.e.: $\nexists\, p' \in D - T : p \in (A(p'))^{\cdot}$

which implies both $p_o \in D - T$ and $p_o \notin D - T$.

This permits us to rewrite the definition of B_{i+1} as:

$$B_{i+1} = B_i \cup \{p \in (D - T) - B_i \mid \nexists p' \in (D - T) - B_i : p \in (A(p'))^\bullet\}$$

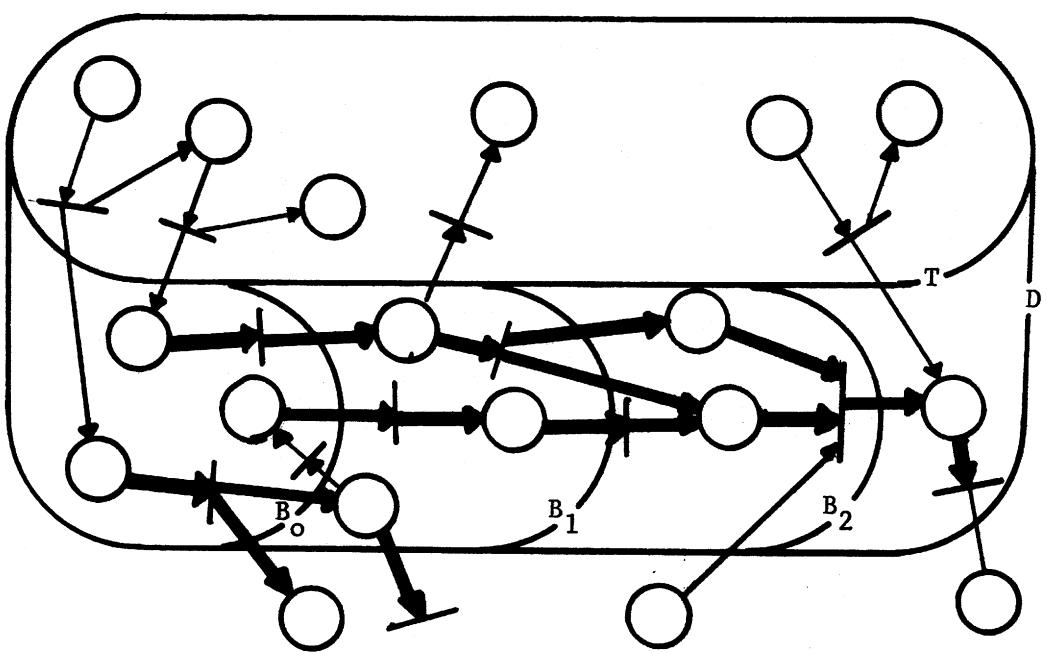
Then $B_{i+1} - B_i = \emptyset \Leftrightarrow$ $\left|\begin{array}{l} (D - T) - B_i = \emptyset \text{ or} \\ \forall p \in (D - T) - B_i \quad \exists p' \in (D - T) - B_i : p \in (Ap'))^\bullet \end{array}\right.$

But the second alternative is impossible since A is circuit-free. Hence, since $B_i \subseteq B_{i+1} \subseteq D - T$

$$B_{i+1} = B_i \Leftrightarrow B_i = D - T$$

This implies that the sequence B_i grows strictly until it covers all of $D - T$. In particular, $D - T$ is some B_i, and hence, by induction:

- Any A-sequence can fire only a bounded number of times in $(D - T)^\bullet$. Since no A-sequence fires in $T^\bullet$, and $(D - T)^\bullet \cup T^\bullet = D^\bullet$, we have:
- There is an upper bound on the number of times any A-sequence can fire in $D^\bullet$. Hence, there exists an A-sequence which leads to a marking $\underline{M}$ such that no A-sequence starting at M can fire in $D^\bullet$.



The circuit-free allocation is shown in bold.

So far, we have not used the Free-Choice Hypothesis. Now we show that, in a Free-Choice net, every firing sequence starting at M is an A-sequence, and hence does not fire in $D^{\cdot}$.

Assume there is a firing sequence σt_o that starts at M and is not an A-sequence, but σ is an A-sequence, i.e. σt_o is the shortest non-A-sequence from marking M. Hence, we must have $p \in {}^{\cdot}t_o$ such that:

$$p \in D - T$$

$$A(p) = t_1 \neq t_o$$

But then, by Free-Choice hypothesis: ${}^{\cdot}t_o = \{p\}$

$${}^{\cdot}t_1 = \{p\}$$

and (t_o firable at $M[\sigma\rangle$) $\Rightarrow$ (t_1 firable at $M[\sigma\rangle$) . But σt_1 is an A-sequence and $t_1 \in D^{\cdot}$: this contradicts our hypothesis that no A-sequence starting at M can fire in $D^{\cdot}$.

This proves Lemma 4.

Lemma 4 immediately implies:

Theorem 2: If a Free Choice net is live, every deadlock contains a marked trap.

Proof: If some deadlock does not contain a marked trap, its maximal trap must be blank: apply Lemma 4.

From Theorems 1 and 2 follows

Commoner's Liveness Theorem (Theorem 3): A Free-Choice Net is live if and only if every deadlock contains a marked trap.

4.2 Safeness; Live-and-Safe Markings

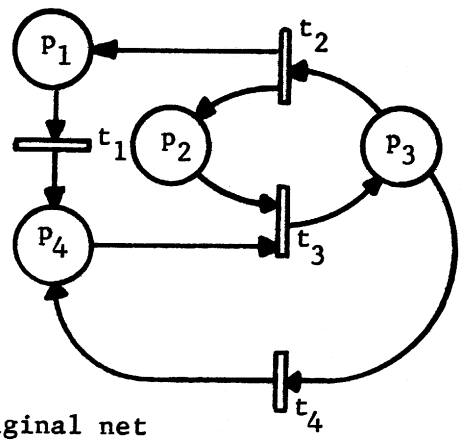
For our purposes, it is not very interesting to study safeness in non-live nets. For example, ~~every~~ Petri Net that has no zero-input transitions has at least one safe marking: The blank marking. Hence, the concept of Live-and-Safe is studied rather than safeness for its own sake.

4.2.1 Definition of a Covering of a Petri Net

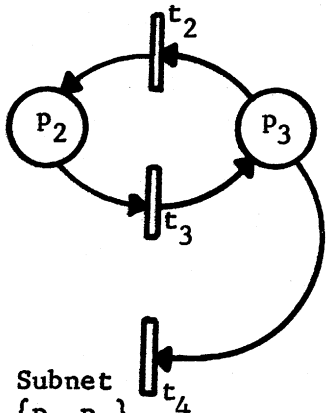
Deadlocks and traps have been defined as sets of places. However, we also use sets of transitions associated with such sets of places, both in the definition $^{\bullet}D \subseteq D^{\bullet}$ and in applications: cf. proofs seen so far. So, we define the concept of a consistent subnet defined by a set of places Q:

Definition: A consistent subnet of a Petri Net $\langle \Pi, \Sigma \rangle$ defined by a set of places $Q \subseteq \Pi$ is the Petri Net $\langle Q, {}^{\bullet}Q \cup Q^{\bullet} \rangle$, i.e. the net consisting of Q and all transitions directly connected to Q.

Example:



Original net



Consistent Subnet defined by $\{p_2, p_3\}$

We also define the union of two consistent subnets defined by $Q \subseteq \Pi$ and $Q' \subseteq \Pi$ as the consistent subnet defined by $Q \cup Q'$.

Definition: A Petri Net is covered by a collection of consistent subnets if the union of these consistent subnets over the collection is the whole net, or equivalently, if every place is in some consistent subnet of the collection.

We say that these subnets form a covering of the original net.

Note that if Q is | a deadlock, it's consistent subnet is $\langle Q, Q^{\bullet} \rangle$
| a trap, it's consistent subnet is $\langle Q, {}^{\bullet}Q \rangle$.

4.2.2 A Necessary Condition for a Live-and-Safe Marking in a Free Choice Net

The prototype of a live and safe net is a net where there is always only one token. Strongly connected State Machines, where every transition has exactly one input and one output place, have such one-token live and safe markings. We will show that the concept of one-token Strongly Connected State Machine (SCSM) is central to the discussion of Live and Safe Free Choice Nets.

We shall first prove that if a Free Choice Net is live and safe, there must exist a covering of one-token SCSM's.

First we note that if the net is live and safe at marking M, the marking M' obtained by removing one token from M is not live. For if it were, we could get another stone on the place where the previous stone was removed, and hence the marking M would have been unsafe. (We must exclude here nets that have isolated places, i.e. not connected to any transition, this should not be a severe restriction however.)

Theorem 4: If a Free-Choice net is Live and Safe, there is a covering by one-token Strongly Connected State Machines:
LSFC $\Rightarrow$ covered by one-token SCSM's.

Proof: a) Live and Free Choice $\Rightarrow$ every deadlock contains a marked trap.

Live and Safe: If we take one token away, the net is non-live, and some deadlock has a blank maximal trap.

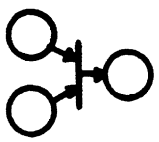
(We need both the necessary and sufficient condition for liveness.)

Hence: LSFC ⇒ every token is the unique token of the maximal trap in some deadlock.

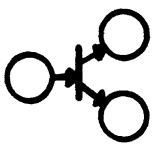
b) Suppose such a deadlock is not minimal. Then the token of the maximal trap will be in the maximal trap of some smaller deadlock. (There is only one token available, every deadlock must contain a marked maximal trap, and the maximal trap of the smaller deadlock is contained in the maximal trap of the containing deadlock.)

Hence: LSFC ⇒ Every token is the unique token of the maximal trap in some minimal Deadlock.

c) In a FC net, the consistent subnet defined by a minimal deadlock does not contain a transition with more than two input places. If there were such a transition, its input places would have no other output transition (Free Choice). But then we could take away all but one input place and still have a deadlock: The deadlock was not minimal. Therefore, the number of tokens in the maximal trap of a minimal deadlock in a FC net may not decrease by any firing sequence.

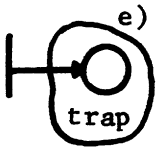


d) Now suppose the consistent subnet defined by the maximal trap in the minimal deadlock has a transition with two output places. If the net is live, every firing of this transition increases the number of tokens on the trap. But it cannot decrease: unbounded, hence unsafe.



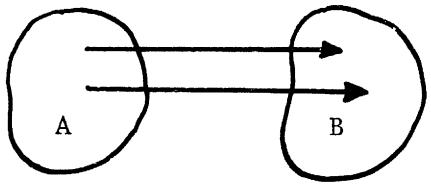
Hence: The maximal trap in a minimal deadlock of a live and safe Free-Choice Net defines a State Machine as consistent subnet.

e) Suppose the maximal trap is not a deadlock itself. There must be a transition which puts a token on the trap without taking one away, hence liveness implies unsafeness, as above.



Hence: LSFC ⇒ every minimal deadlock is a trap and defines a State Machine.

f) Suppose a minimal deadlock that is a non-strongly connected State Machine:



But then, if AB is a deadlock, so is A, hence AB cannot be minimal.

LSFC ⇒ every minimal deadlock defines a SCSM.

g) From b) and f) it follows that every token is the unique token in a SCSM. But the net is assumed to be live: any place can hold a token at some time. (We exclude nets with isolated places.)

Hence: LSFC ⇒ covered by one-token SCSM's.

q.e.d.

4.2.3 Sufficiency Condition for Safeness in a Live Free Choice Net.

Now we wish to prove that a one-token SCSM covering is sufficient for safety, and derive a necessary and sufficient condition for live-and-safeness of a Free Choice net.

Lemma 5: In a Free Choice net that does not have a live and safe marking, every live marking is unbounded (some place collects an unbounded number of tokens).

Proof: By hypothesis, every live marking is unsafe. From the liveness theorem we know that if a marking M is live, so is the marking $M' = M \cap 1$ obtained by removing, from every place, every token except one: Every trap remains marked.

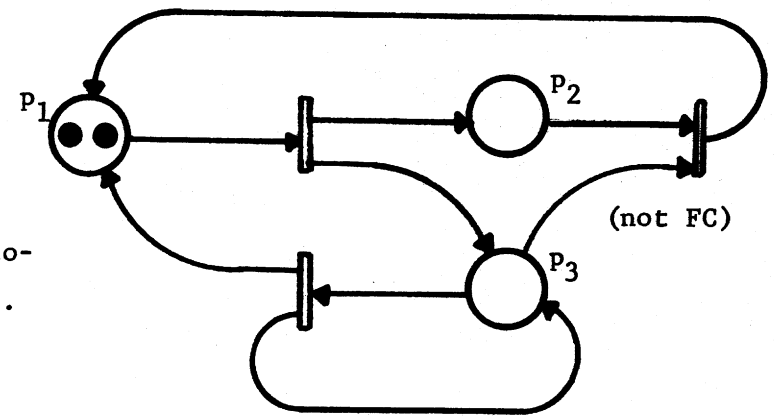
Let M_o be a live marking, hence unsafe. We shall fire until we reach a marking $M_o' \in \vec{M}_o$ where some place has more than one token. We now paint, in every place, every token red except one, and pledge not to move the red tokens anymore. We continue firing with the non-painted tokens, effectively we fire now in $\vec{M}_1$ where $M_1 = M_o' \cap 1$.

Since M_1 is live, it is unsafe; fire until M_1' where some place contains more than one token, paint some tokens red, continue firing in $\vec{M}_2$ where $M_2 = M_1' \cap 1$, etc. At each step, the number of red tokens <u>strictly increases</u>. But our pledge not to move them is perfectly consistent with the firing rule in $\vec{M}_o$, any marking in $\vec{M}_i$ together with all red tokens accumulated so far is a marking in $\vec{M}_o$: $\vec{M}_o$ is <u>unbounded</u>: there is no bound on the number of tokens in the markings of $\vec{M}_o$.

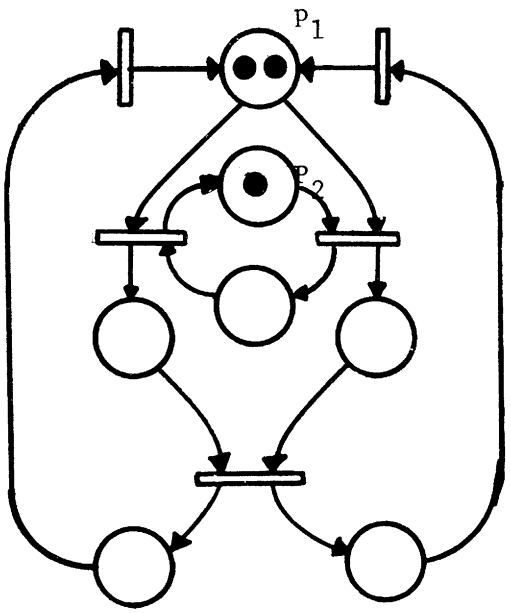
q.e.d.

The above lemma only depends on the fact that liveness is determined by places having tokens or not, in contrast to having a specific number of tokens. This property holds for FC nets but not for more general nets:

This net is live for one or more tokens at place p_1 .



But it is false for the following net:



No live marking is safe, but the marking $\begin{vmatrix} M(p_1) = 2 \\ M(p_2) = 1 \end{vmatrix}$ is live, unsafe, bounded.

But removing one token from p_1 kills the net. Surprisingly, adding one token to p_2 also kills the net!

4.3 The Live-and-Safeness Theorem

Theorem 5: If a Free Choice net is covered by Strongly Connected State Machines and has a live marking, it has a live and safe marking.

Proof: The number of tokens on any of the covering SCSM's is constant for all firing sequences. Hence an upper bound for the number of tokens is the sum of the number of tokens over all covering SCSM's. (If a token is shared among several covering SCSM's, it is counted several times.) But then, by lemma 5, if there is a live marking there must be a live and safe marking.

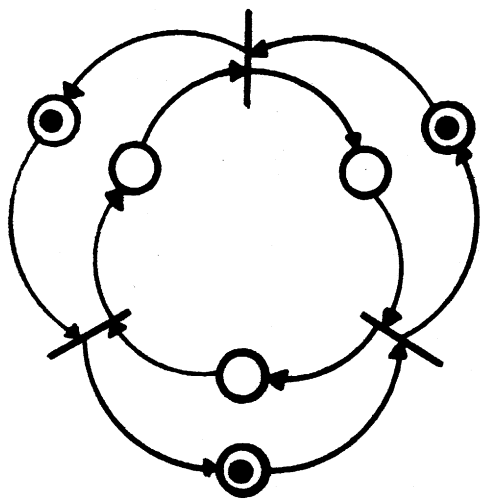
q.e.d.

From the proof of Theorem 4 (necessary condition for safeness) it follows that in a live and safe Free Choice net every minimal deadlock is a SCSM. Conversely, a SCSM is always a minimal deadlock and contains a trap, namely itself.

Hence:

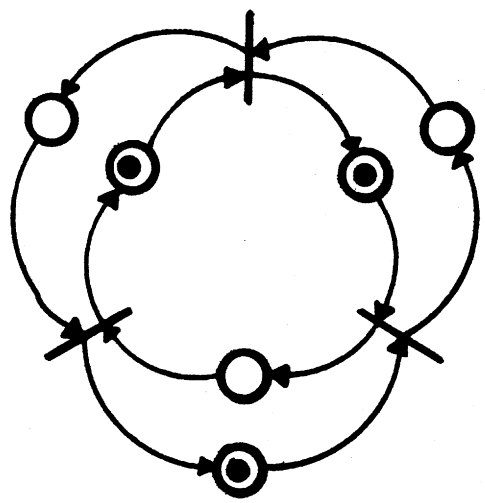
Live-and-Safeness Theorem: A Free Choice net is live and safe if and only if it is covered by one-token SCSM's and every minimal deadlock is a marked SCSM.
(Theorem 6)

The following example shows the importance of the word marked SCSM:



- covered by one-token SCSM's
- every minimal deadlock is a SCSM
- some minimal deadlock is blank

not Live and Safe



- covered by one-token SCSM's
- every minimal deadlock is SCSM, and marked
- some minimal deadlock has 2 tokens

Live and Safe

Corollary: A Free Choice net has a live and safe marking if and only if it is covered by SCSM's and every minimal deadlock is a SCSM.

Proof: The only-if part follows immediately from Theorem 6. Now suppose every minimal deadlock is a SCSM, hence contains a trap: The marking that has at least one token on each SCSM is live. Then, by Theorem 5, it has a live-and-safe marking.

q.e.d.

This empty page was substituted for a blank page in the original document.

CHAPTER 5
Decomposition of Free Choice Petri Nets

5.1 Well-Formedness in Free Choice Petri Nets

In the Live-and-Safeness Theorem (Theorem 6) we used the concept of a covering by Strongly Connected State Machines. In this chapter we shall consider an algorithm for obtaining such a decomposition. There may be several possible coverings of SCSM's that satisfy the corollary of Theorem 6 (Existence of a Live-and-Safe Marking). Our algorithm will produce all such coverings. If the net has no SCSM coverings that satisfy Theorem 6, the algorithm will produce subnets that are not strongly connected, or not State Machines. This gives us yet another test for the existence of a Live-and-Safe Marking in a Free Choice net.

For convenience, we shall call a Free Choice net that satisfies the corollary of Theorem 6 a Well-Formed (WF) Free Choice Net. This chapter then discusses various Well-Formedness criteria and tests.

Definition: A Free Choice Petri Net is Well-Formed if it is covered by Strongly Connected State Machines and every minimal deadlock is a Strongly Connected State Machine.

Corollary: A Free Choice Petri Net has a Live-and-Safe Marking if and only if it is Well-Formed.

FC: $\exists$LS $\Leftrightarrow$ WF

5.2 Duality, Reverse-Duality; Open and Closed Consistent Subnets

The decomposition algorithms and proofs in this chapter require the definition of some new concepts.

If we compare the definitions of Deadlocks and Traps, or State Machines and Marked Graphs, we note a striking similarity: A Trap has the same definition as a Deadlock if we reverse all arrows, i.e. if we transpose, throughout the definition, the words input and output. A

Marked Graph has the same definition as a State Machine if we transpose, throughout the definition, the words place and transition. In the first case, we say that a Deadlock is the reverse of a Trap (and vice versa); in the second case, we say that a Marked Graph is the dual of a State Machine (and vice versa).

If we now look at the definition of a Free Choice Net, we observe that by transposing the words input and output (and also transpose to and from), and then transposing the words place and transition, we get the same definition:

before: Every arc from a place to a transition is either the unique input arc to a transition, or the unique output arc from a place.

after: Every arc to a transition from a place is either the unique output arc from a place, or the unique input arc to a transition.

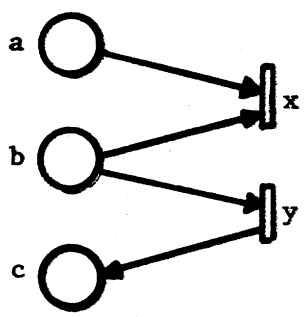
We express this by saying that the reverse-dual of a Free Choice Net is a Free Choice net.

Formally, we have:

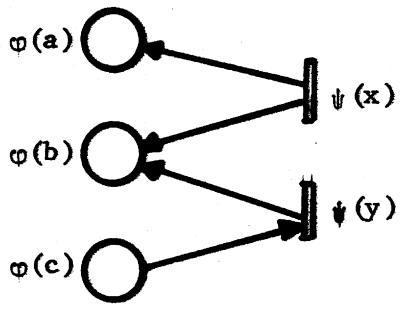
Definition: • The reverse of a Petri Net $\langle \Pi, \Sigma, \cdot \rangle$ is a Petri Net $\langle \Pi', \Sigma', o \rangle$ such that there are two bijections φ and ψ:

$$\left.\begin{array}{l}\varphi : \Pi \to \Pi' \\ \psi : \Sigma \to \Sigma'\end{array}\right) \quad \text{and} \quad \left.\begin{array}{l}\forall p \in \Pi \\ \forall t \in \Sigma\end{array}\right) \; p \cdot t \Leftrightarrow \psi(t) \; o \; \varphi(p)$$

(arrow-reversal)



primal

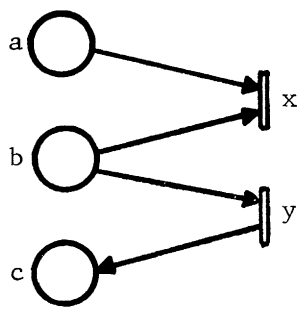


reverse

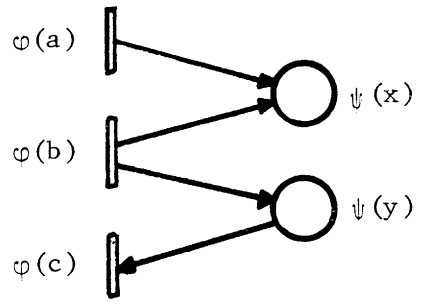
Definition: • The dual of a Petri Net $\langle \Pi, \Sigma, \cdot \rangle$ is a Petri Net $\langle \Pi', \Sigma', o \rangle$ such that there are two bijections φ and ψ:

$$\left.\begin{array}{l} \varphi : \Pi \to \Sigma' \\ \psi : \Sigma \to \Pi' \end{array}\right\} \quad \text{and} \quad \left.\begin{array}{l} \forall p \in \Pi \\ \forall t \in \Sigma \end{array}\right\} \; p \cdot t \Leftrightarrow \varphi(p) \circ \psi(t)$$

(place-transition interchange)



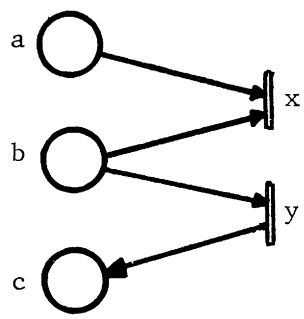
primal



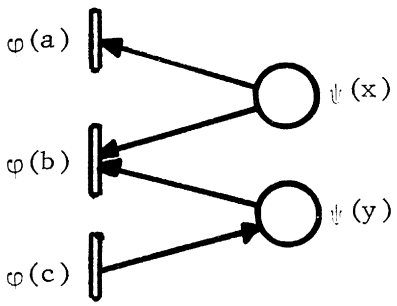
dual

Definition: • The reverse-dual of a Petri Net $\langle \Pi, \Sigma, \cdot \rangle$ is the net $\langle \Pi', \Sigma', o \rangle$ such that there are two bijections φ and ψ:

$$\left.\begin{array}{l} \varphi : \Pi \to \Sigma' \\ \psi : \Sigma \to \Pi' \end{array}\right\} \quad \text{and} \quad \left.\begin{array}{l} \forall p \in \Pi \\ \forall t \in \Sigma \end{array}\right\} \; p \cdot t \Leftrightarrow \psi(t) \circ \varphi(p)$$



primal



reverse-dual

It is clear that:
- reverse of dual = dual of reverse = reverse-dual
- dual of dual = primal
- reverse of reverse = primal
- reverse-dual of reverse-dual = primal

(primal = the original net)

Lemma 6: The reverse-dual of | Free-Choice is Free Choice.
| State Machine is Marked Graph
| Marked Graph is State Machine
| Strongly connected is Strongly Connected.

Proof: Let the primal be $\langle \Pi, \Sigma, \cdot \rangle$

Let the reverse-dual be $\langle \psi(\Sigma), \varphi(\Pi), o \rangle$

where φ and ψ are bijections.

Then: (FC in primal) $\equiv (\forall p \in \Pi \ \forall t \in \Sigma: p \cdot t \Leftrightarrow p^{\cdot} = \{t\}$ or $ {}^{\cdot}t = \{p\})$

But, in the reverse-dual, we get:

$$p \cdot t \Leftrightarrow \psi(t) \,.\, \varphi(p)$$

$$p^{\cdot} = \{t\} \Leftrightarrow {}^{\cdot}\varphi(p) = \{\psi(t)\}$$

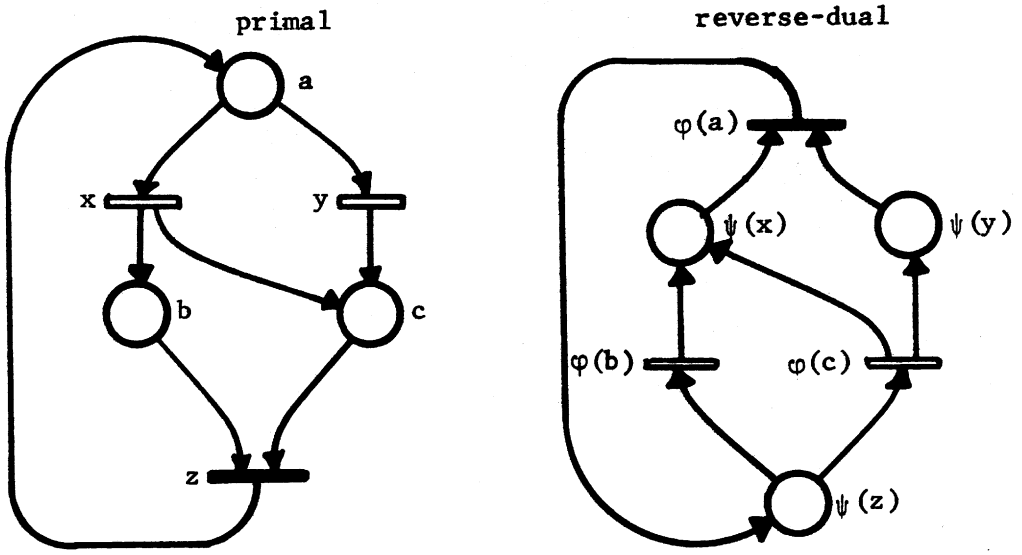
$$ {}^{\cdot}t = \{p\} \Leftrightarrow \psi(t)^{o} = \{\varphi(p)\}$$

hence: $\psi(t) \circ \varphi(p) \Leftrightarrow {}^{o}\varphi(p) = \{\psi(t)\}$ or $\psi(t)^{o} = \{\varphi(p)\}$

$$p' \cdot t' \Leftrightarrow {}^{\cdot}t' = \{p'\} \quad \text{or} \quad p'^{\cdot} = \{t'\}$$

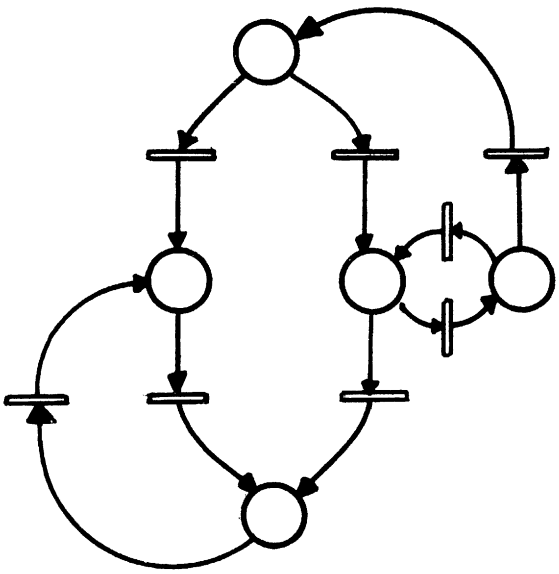
The three remaining points of the Lemma are trivial.

Example 1: Strongly connected Free Choice net. This example happens to be self - reverse-dual.

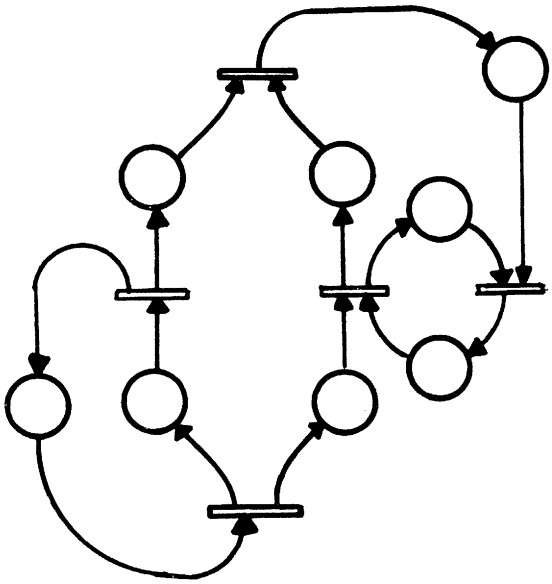


Example 2:

primal



reverse-dual



(both not strongly connected)

State Machine

Marked Graph

We defined the notion of a consistent subnet defined by a set of places. The dual (and reverse-dual) notion of this is a consistent subnet defined by a set of transitions, and consisting of these transitions and all places connected to them:

Definition: A consistent subnet defined by a set of transitions $T \subseteq \Sigma$ of a Petri Net $\langle \Pi, \Sigma \rangle$ is the Petri net $\langle {}^{\cdot}T \cup T^{\cdot}, T \rangle$.

We shall emphasize the distinction of the two kinds of consistent subnets by calling them closed and open respectively:

Definition:

- A closed consistent subnet is a subnet $\langle \Pi, \Sigma \rangle$ such that $\Sigma = {}^{\cdot}\Pi \cup \Pi^{\cdot}$ (defined by its places)
- An open consistent subnet is a subnet $\langle \Pi, \Sigma \rangle$ such that $\Pi = {}^{\cdot}\Sigma \cup \Sigma^{\cdot}$ (defined by its transitions).

The distinction takes its name from the fact that the former is separated from the rest of the net by a boundary* of transitions, the latter by a boundary of places (more "open" than transitions).

To every statement about a Free-Choice Petri net corresponds a statement about the reverse-dual net:

primal	reverse-dual
place	transition
input (to)	output (from)
input arc to a transition	output arc of a place
covering by SCSM's	covering by SCMG's (Strongly Connected Marked Graphs)
Closed Consistent Subnet	Open Consistent Subnet
SM-allocation	MG-allocation

etc.

Note also that the reverse of a trap is a deadlock, but we have no interpretation yet for the dual or the reverse-dual notion of a trap.

5.3 Decomposition of a Free-Choice Net into a Covering of SCSM's

We shall describe a reduction method which, given an FC net, constructs all possible SCSM's that form a covering. The method is such that if the net is well-formed, every reduction yields a SCSM and they cover the net; if the net is not well-formed, some reduction will not yield a SCSM, or the reductions will not cover the net.

We recall, from the proof of Theorem 4, that in a Free Choice net we can construct a minimal deadlock by choosing any one of the input places to a transition that has one output place committed to the deadlock. So, to reduce the net to one of its component SCSM's we make such a choice ahead of time for all transitions.

---

*The boundary of a subnet $\langle \Pi', \Sigma' \rangle$ in a net $\langle \Pi, \Sigma \rangle$ is the set $\{x \mid x \in \Pi' \cup \Sigma' \text{ and } (x^{\cdot} \cup {}^{\cdot}x) \cap ((\Pi - \Pi') \cup (\Sigma - \Sigma')) \neq \emptyset\}$

We shall therefore define an allocation of input places to transitions much like we defined an allocation of output transitions to places in the proof of Lemmas 3 and 4. Since we wish to construct state machines, we distinguish this allocation by calling it a state-machine allocation, or SM-allocation.

IMPORTANT NOTE: We shall from now on interpret "strongly connected" and "SC" as "consisting of strongly connected components."

Hence, a reduced net consisting of several disjoint but individually strongly connected State Machines (or Marked Graphs) will also be called SCSM (or SCMG).

Definition: An SM-allocation over a Free Choice net $\langle \Pi, \Sigma \rangle$ is a function $B : \Sigma \rightarrow \Pi$ such that:

$$\forall t \in \Sigma \quad B(t) \in {}^{\bullet}t$$

Given such an SM-allocation B we will now reduce the net to a SCSM (if possible) that does not contain unallocated places:

Step 1: Delete all unallocated places. $(\Pi - B(\Sigma))$

Step 2: Delete all transitions that have all output places already deleted.

Step 3: Delete all places that have at least one output transition already deleted.

Repeat Steps 2 and 3 until neither is applicable anymore.

What is left over is the reduced net. Each step eliminates some elements that would not be part of a SCSM consistent with the SM-allocation.

Formally, we construct the sets of eliminated places (E_p) and transitions (E_t) as follows, given an SM-allocation B on a FC net $\langle \Pi, \Sigma \rangle$:

$$\left.\begin{array}{l}\forall p \in \Pi \\ \forall t \in \Sigma\end{array}\right| \begin{array}{ll} {}^{\bullet}t - \{B(t)\} \subseteq E_p & \text{(step 1)} \\ t^{\bullet} \subseteq E_p \Leftrightarrow t \in E_t & \text{(step 2)} \\ p^{\bullet} \cap E_t \neq \emptyset \Leftrightarrow p \in E_p & \text{(step 3)} \end{array}$$

Then the SM-reduced net is defined as the Petri Net $\langle \Pi - E_p, \Sigma - E_t \rangle$, say $\langle Q_p, Q_t \rangle$. Hence:

$$Q_p = \Pi - E_p$$
$$Q_t = \Sigma - E_t$$

From the definition follows immediately:

$$Q_p \subseteq {}^{\bullet}Q_t \qquad Q_p \subseteq Q_t^{\bullet}$$
$${}^{\bullet}Q_p \subseteq Q_t \qquad Q_p^{\bullet} \subseteq Q_t$$

and hence:

$$Q_p \subseteq {}^{\bullet}Q_t \cap Q_t^{\bullet}$$
$${}^{\bullet}Q_p \cup Q_p^{\bullet} \subseteq Q_t$$

Now assume $t^{\bullet} \cap Q_p = \emptyset$

It follows that: $t^{\bullet} \subseteq E_p$

$t \in E_t$

$t \notin Q_t$

Hence: $(t \in Q_t) \Rightarrow (t^{\bullet} \cap Q_p \neq \emptyset) \Rightarrow (\exists p \in Q_p : t \in {}^{\bullet}p) \Rightarrow (t \in {}^{\bullet}Q_p)$

i.e. $Q_t \subseteq {}^{\bullet}Q_p$

Hence: ${}^{\bullet}Q_p \cup Q_p^{\bullet} \subseteq Q_t \subseteq {}^{\bullet}Q_p$

i.e.: $Q_t = {}^{\bullet}Q_p \cup Q_p^{\bullet}$ (closed consistent subnet)

$Q_p^{\bullet} \subseteq {}^{\bullet}Q_p$ (trap)

Also, by construction, $\forall t \; |{}^{\bullet}t \cap Q_p| \leq 1$: (non-decreasing)

Lemma 7: An SM-reduction of a FC net is a closed consistent subnet defined by a non-decreasing trap.

We shall now prove a sufficient condition for Well-Formedness in terms of SM-reductions of a FC net:

Theorem 7: If every SM-reduction of a FC net is a SCSM, and they cover the net, then the net is WF.

Proof: All that is required to prove is that every minimal deadlock $\langle D, D^{\cdot} \rangle$ is a SCSM.

We know that because D is minimal in a FC net,

$$\forall t \in D^{\cdot} \quad |{}^{\cdot}t \cap D| = 1$$

We say that an SM-allocation B and the corresponding SM-reduction are consistent with the minimal deadlock D

if: $\forall t \in D^{\cdot} \; : \; {}^{\cdot}t \cap D = \{B(t)\}$

Such allocations exist because of the fact that $|{}^{\cdot}t \cap D| = 1$. (Note that, since the deadlock is minimal, this implies $B(D^{\cdot}) = D$.)

First, we show that the minimal deadlock D must intersect each SM-reduction $\langle Q_p, Q_t \rangle$ consistent with D, i.e. that $D \cap Q_p \neq \emptyset$.

Assume the contrary: $D \cap Q_p = \emptyset$ for every SM-allocation B consistent with D, whose associated SM-reduction is $\langle Q_p, Q_t \rangle$.

case 1: $\forall t \in D^{\cdot} \quad |{}^{\cdot}t| = 1$

In this case, every SM-allocation is consistent with D, hence deletes all of D (since, by assumption, $D \cap Q_p = \emptyset$). This contradicts the fact that the reductions cover the net.

case 2: $\exists t_o \in D^{\cdot} \quad |{}^{\cdot}t| \geq 2$

For any SM-allocation B' not consistent with D, let:

$$\left|\begin{array}{l} {}^\bullet t_o \cap D = \{p_o\} \\ B'(t_o) = p_1 \\ p_1 \neq p_o \qquad (B' \text{ not consistent with } D) \end{array}\right.$$

Then, every SM-allocation not consistent with D (such as B') deletes p_o (Step 1: p_o is unallocated). But, by assumption, every SM-allocation consistent with D also deletes p_o:
The reductions do not cover: <u>contradiction</u>.

<u>Hence</u>: Every minimal deadlock D intersects some SM-reduction $\langle Q_p, Q_t \rangle$ consistent with D:

$$D \cap Q_p \neq \emptyset$$

Now, let $p \in D \cap Q_p$

then: ${}^\bullet p \subseteq D^\bullet$ because D is a deadlock.

${}^\bullet p \subseteq Q_t$ because the reduction is a Closed Consistent subnet.

also, $\forall t \in {}^\bullet p : {}^\bullet t \cap D = B(t) \in Q_p$,

because the reduction $\langle Q_p, Q_t \rangle$, defined by SM-allocation B, is consistent with $\langle D, D^\bullet \rangle$.

Hence: ${}^\bullet({}^\bullet p) \cap D \subseteq Q_p \cap D$

By repeating this process for each place in ${}^\bullet({}^\bullet p)$ along backwards paths until D <u>or</u> Q_p is exhausted (which must happen since D and Q_p are minimal deadlocks -- the latter because it is SCSM -- and hence every place can be reached by a backwards path) we get $D \subseteq Q_p$ or $Q_p \subseteq D$. But since both are minimal deadlocks, we must have:

$$D = Q_p$$

Hence D is SCSM.

q.e.d.

5.4 Decomposition of a FC Net Into a Covering of Strongly Connected Marked Graphs

A Free-Choice Net can be considered as an extension of State Machines by allowing Marked-Graph-type concurrency, or as an extension of Marked Graphs by allowing State-Machine-type conflict. Historically, this view is at the origin of the concept of Free-Choice nets.

So far, we were concerned with the State-machine-like behavior of FC nets. But, noting that the reverse-dual of a FC net is also FC, we shall now use this as a tool for analyzing Marked-Graph-related properties.

We used SM-allocation reduction to get a decomposition into Closed Consistent Subnets. Now, we define Marked-Graph allocation as the reverse-dual concept and use it to get Open Consistent Subnets.

Definition: A Marked-Graph Allocation (MG-allocation) over a Free Choice net $\langle \Pi, \Sigma \rangle$ is a function

$$A : \Pi \rightarrow \Sigma$$

such that $\forall p \in \Pi : A(p) \in p^{\cdot}$

This is exactly the type of allocation we used over a subset of places in the proof of Theorem 2.

Now we define MG-reduction, given an MG-allocation A, by translating the definition of SM-reduction into the corresponding reverse-dual concepts:

Step 1: Delete all unallocated transitions.

Step 2: Delete all places that have all input transitions already deleted.

Step 3: Delete all transitions that have at least one input place already deleted.

Repeat Steps 2 and 3 until neither is applicable anymore.

What is left over is the reduced net. Each step eliminates some elements that would not be part of a SCMG consistent with the MG-allocation.

However, we can also interpret this reduction as the elimination of all those parts in the net that would not be active if we were to use the allocation as a choice for multiple-output places: We deliberately choose not to fire unallocated transitions (Step 1); if all token flow is interrupted to a place, that place becomes inactive (Step 2); and if some input place to a transition is inactive, that transition will be inactive (Step 3). This description is informal at best, but if we interpret "inactive" as "receiving only a finite number of tokens," or "firable only a finite number of times," it will be useful for proofs about liveness.

Formally, we define the reduced net as follows:

- Sets of deleted places E_p, deleted transitions E_t:

$$\left.\begin{array}{l}\forall p \in \Pi \\ \forall t \in \Sigma\end{array}\right) \quad \left|\begin{array}{ll} p^{\bullet} - \{A(p)\} \subseteq E_t & \text{(Step 1)} \\ {}^{\bullet}p \subseteq E_t \Leftrightarrow p \in E_p & \text{(Step 2)} \\ {}^{\bullet}t \cap E_p \neq \emptyset \Leftrightarrow t \in E_t & \text{(Step 3)}\end{array}\right.$$

The MG-reduced net, via MG-allocation A, is the net $\langle Q_p, Q_t \rangle$ where $Q_p = \Pi - E_p$

$$Q_t = \Sigma - E_t$$

As in the case of SM-reduction, we get by reverse-duality:

$$Q_p = Q_t^{\bullet} \cup {}^{\bullet}Q_t = Q_t^{\bullet} \; ; \quad \underline{\text{Open Consistent Subnet}}$$

$${}^{\bullet}Q_t \subseteq Q_t^{\bullet}$$

$$\forall p \; |p^{\bullet} \cap Q_t| \leq 1 \qquad \underline{\text{Conflict-free}}$$

We have no significant interpretation yet for ${}^{\bullet}Q_t \subseteq Q_t^{\bullet}$. We summarize these facts by:

Lemma 8: An MG-reduction of a FC net is a conflict-free open consistent subnet. MG-reductions provide us with a necessary condition for well-formedness.

Lemma 9: If some MG-reduction of a FC net is empty the net is not live.

Proof: If some MG-reduction is empty, the set of eliminated transitions E_t and eliminated places E_p form the whole net, for some MG-allocation A. Let us do the reductions step by step and check for possible firings of the eliminated transitions by A-sequences (see proof of Theorem 2).

Step 1: No A-sequence fires any unallocated transition, by definition. We start building E_t with transitions firable at most a bounded number of times.

Step 2: Eliminate those places that have only deleted input transitions. By inductive hypothesis, these transitions can only fire a bounded number of times. Hence, these eliminated places can fire their output transitions only a bounded number of times.

Step 3: Eliminate those transitions that have at least one input place deleted. By the explanation of step 2, they can fire only a bounded number of times: This supports the inductive hypothesis of bounded firability for a repetition of steps 2 and 3.

Since all transitions will be eliminated by hypothesis, every A-sequence can fire each transition only a bounded number of times.

Now let M be any marking, and let σ be an A-sequence such that no transition is firable by an A-sequence starting at $M' = M[\sigma\rangle$. We just proved the existence of such an A-sequence.

By the same reasoning as used in the proof of Lemma 4, we show that every firing sequence starting at M' must be an A-sequence, i.e. no transition can be fired by any firing sequence starting at M'. For suppose some transition is firable at M'. It must be an unallocated transition $t_o \in p_o^{\cdot} - \{A(p_o)\}$ for some p_o, since it must be part of a non-A-sequence. But, by Free Choice hypothesis: t_o firable $\Leftrightarrow$ $A(p_o)$ firable, which contradicts the assumption that no A-sequence can fire at M'.

q.e.d.

Lemma 10: If some MG-reduction of a live FC net is not a SCMG, the net is unsafe.

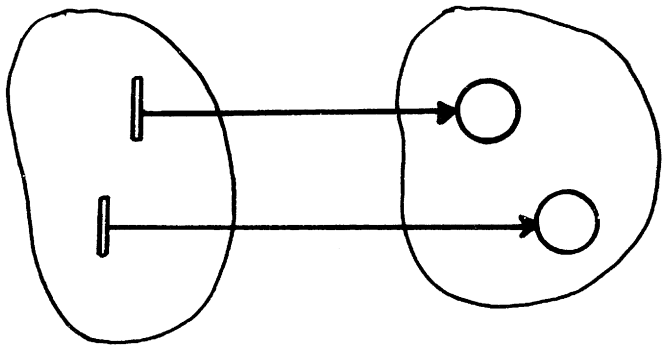
Proof: a) Let us consider the MG-reduction within the original net. Since each transition in the subnet has all the places connected to it both in the original net and in the subnet (open consistent subnet) a transition is firable in the subnet if and only if it is firable in the original net, and the effect of that firing on the marking is the same. Hence, if a firing in the subnet leads to an unsafe marking, the net is unsafe; if it leads to a marking where no transition in the subnet can be fired (A-sequence), then no firing sequence in the original net can fire any transition in the subnet; in this latter case, the same argument used in Lemma 4 and Lemma 9 applies again.

Hence: Net live $\Rightarrow$ MG-reduction live

MG-reduction unsafe $\Rightarrow$ Net unsafe

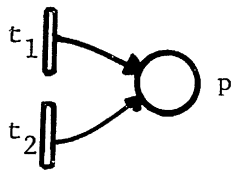
b) Now consider the MG-reduced net $\langle Q_p, Q_t \rangle$ alone. Assume it has a live marking. We shall show it is unsafe if it is not a SCMG.

- if it is not strongly connected, it must be unsafe:

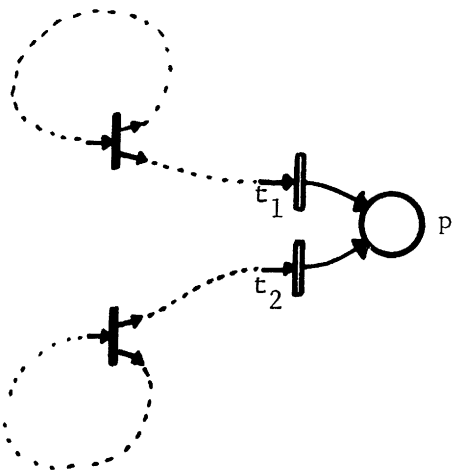


(We must assume here that no transition t is such that $t^{\bullet} = \emptyset$; but this is guaranteed if the original net does not contain such a transition.)

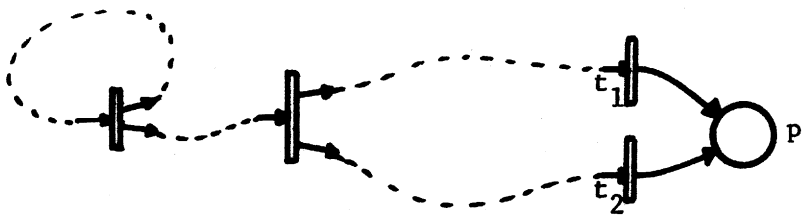
- if it is not a Marked Graph, it must contain a place p with more than one input transition, since more than one output transition is excluded by construction. Since ${}^{\bullet}Q_t \subseteq Q_t^{\bullet}$ there exists an infinite backwards path from each input transition to p, i.e. the backwards path ends in a loop. There are two cases:



- the paths do not intersect:
  Then liveness implies that t_1 and t_2 be concurrently firable, hence p is unsafe.



- the paths intersect. Then, since no place has several outputs, the paths must recombine at a transition:



Again, liveness implies unsafeness.

Hence: not SCMG / live $\Rightarrow$ unsafe

q.e.d.

Lemma 11: If, in a Strongly Connected Free Choice net, every MG-reduction is strongly connected and non-empty, the reductions cover the net.

Proof: If the transitions are covered, the places are covered because the reductions are open consistent subnets. Assume some transition t is not covered, i.e. t is not in any MG-reduction. Since the net is strongly connected, we have: $\forall t$, $|{}^\bullet t| \geq 1$.

Case 1: $|{}^\bullet t| = 1$. Then, if every reduction eliminates t, every reduction must eliminate ${}^\bullet t$, hence all of ${}^\bullet({}^\bullet t)$ (Step 2 of reduction). If all $t' \in {}^\bullet({}^\bullet t)$ are such that $|{}^\bullet t'| = 1$, repeat case 1 for some t'. If not, apply case 2.

Case 2: $|{}^\bullet t| \geq 2$. This case must arise at some time because if not the search assumed in case 1 would exhaust the net, which contradicts the assumption that no reduction is empty.

But now, by Free Choice hypothesis, each place in $\cdot t$ is a single-output place. If each reduction eliminates all of $\cdot t$, repeat the argument for $t' \in \cdot(\cdot t)$ as in case 1.

If some reduction eliminates only part of $\cdot t$, since it eliminates t there would be places without output transitions in the reduced net: not strongly connected.

In any case, the existence of an uncovered transition implies the existence of either an empty or a non-strongly-connected MG-reduction.

q.e.d.

From Lemmas 9, 10, and 11 and Theorem 6 with the well-formedness cololllary we get:

Theorem 8: If a Free Choice net is Well-Formed, every MG-reduction is a non-empty SCMG and the reductions cover the net.

5.5 The Well-Formedness Theorem

We are now ready for the Well-Formedness Theorem, which includes all criteria for the existence of a Live and Safe Marking, including Theorems 7 and 8 and their converses.

Well-Formedness Theorem: (Theorem 9) In a Free Choice Petri net, the following are equivalent:

a) The net is Well-Formed:
   - every minimal deadlock is SCSM
   - there is a covering of SCSM's

b) The net has a Live and Safe marking.

c) The reverse-dual is Well-Formed.

d) Every SM-reduction is a SCSM, the reductions cover the net, no reduction is empty.

e) Every MG-reduction is a SCMG, the reductions cover the net, no reduction is empty.

Proof:

Note: If a is a statement about a FC-net, let a' be the same statement for the reverse-dual of the net.
Example: c = a'

a ⇔ b: Corollary of Theorem 6

a ⇒ e: Theorem 8

e ⇔ d', reverse-dual of e for the reverse-dual net, i.e.

(e for primal) ⇔ (d for reverse-dual)

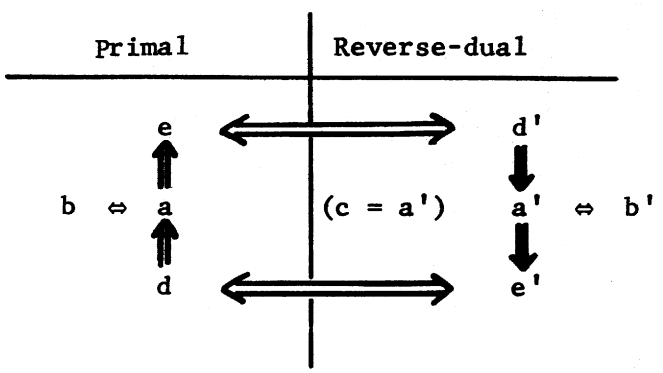
If the primal is such that every MG-reduction is a SCMG etc., the reverse-dual is such that every SM-reduction is a SCSM.

d' ⇒ a' : Theorem 7
d' ⇒ c
a' ⇒ e' : Theorem 8
} for the reverse-dual net

e' ⇔ d : reverse-duality

d ⇒ a : Theorem 7

We have the following diagram:



The implication path is closed and hence gives us the equivalence of statements a, b, c, d, e, a', b', d' and e'.

Remark: Statement d is more complete than the one used in Theorem 7; the part "no SM-reduction is empty" follows by reverse-duality of the full statement of Theorem 8. It is not essential in the proof of this theorem.

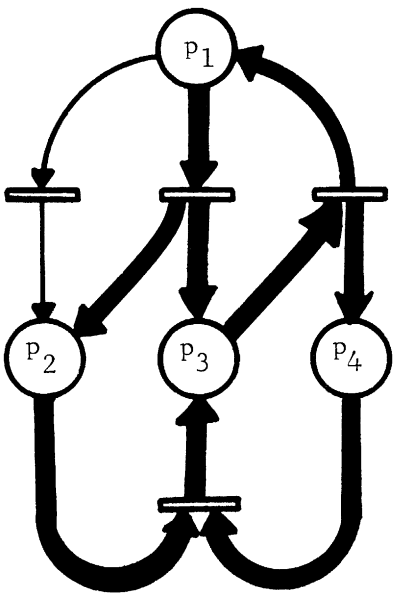
q.e.d.

5.6 Examples of Decompositions

We give below four examples of non-Well-Formed Free Choice Petri Nets. All four are strongly connected, but show different possibilities of structural unsoundness.

Example 1:

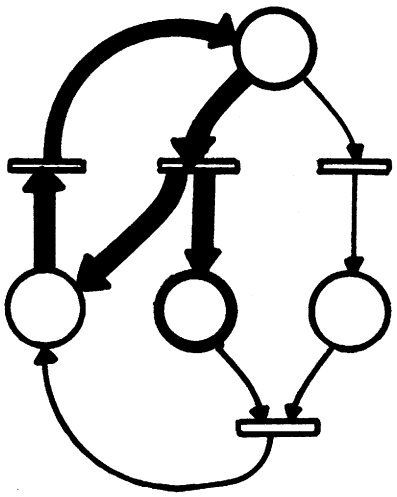
(Reductions shown are superimposed in bold on the original net.)



- one MG-reduction is not a MG (shown).
- one SM-reduction is not a SM.
- the other SM-reduction is empty.
- MG-reductions cover, the SM-reductions do not cover.

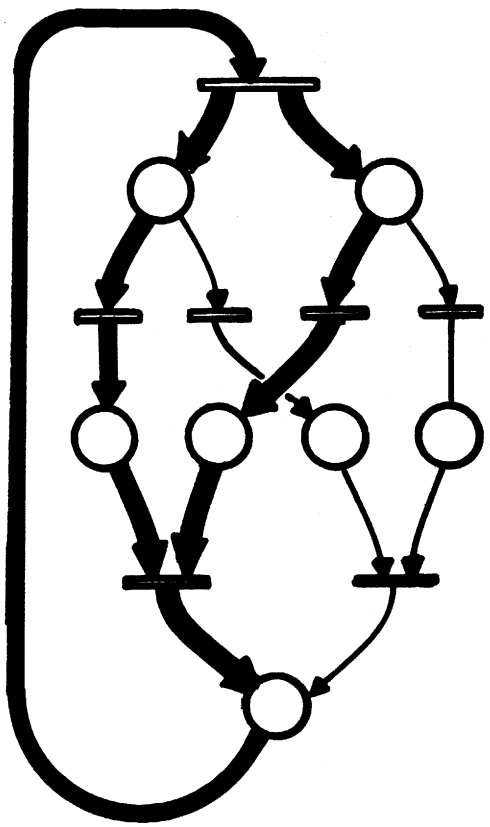
This example has live markings: The minimal deadlocks are $\{p_1, p_2, p_3\}$, which is a trap, and $\{p_1, p_3, p_4\}$, which contains the trap $\{p_3, p_4\}$. But no live marking is safe.

Example 2:



- one MG-reduction is not SC (shown).
- the other MG-reduction is empty.
- one SM-reduction is empty.
- neither SM-reductions nor MG-reductions cover the net.

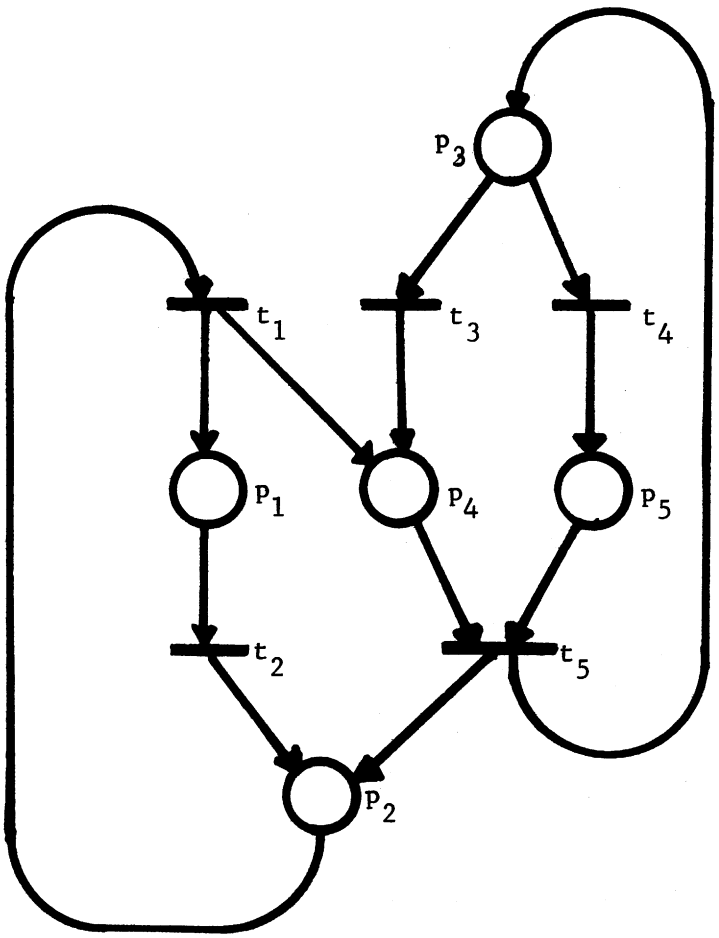
Example 3:



- two MG-reductions (one is shown) are SCMG's and cover the net.
- the two other MG-reductions are empty.
- same for SM-reductions (the net is self-reverse-dual)

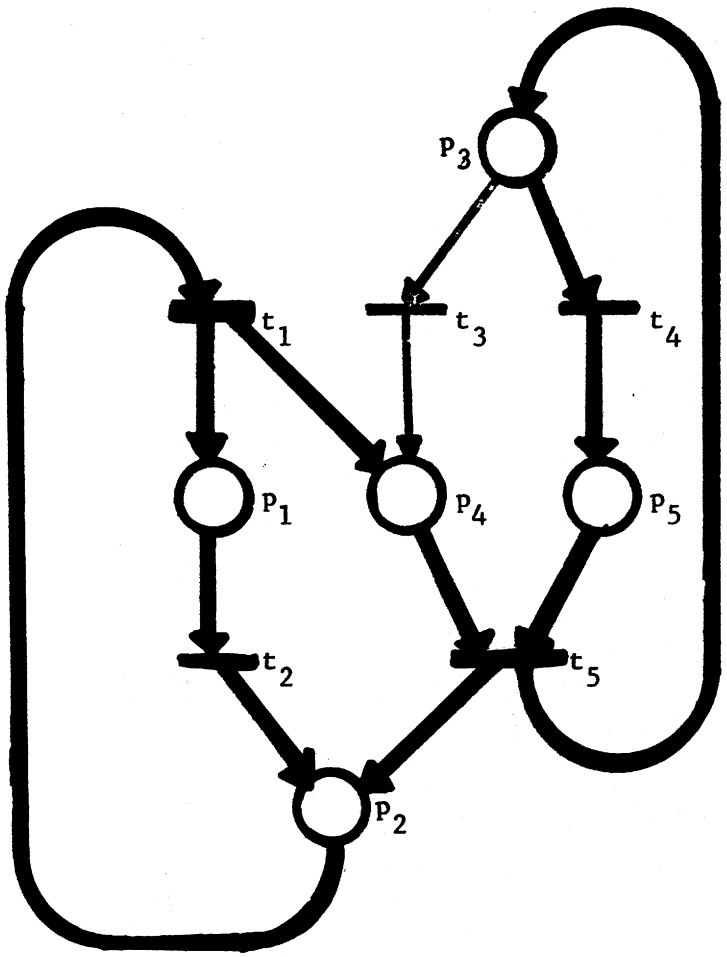
Examples 2 and 3 have no live markings: The empty MG-reduction guarantees the existence of a killing sequence.

Example 4:



This Petri Net has no live marking, but it contains a live subnet $\{\langle p_1, p_2\rangle, \langle t_1, t_2\rangle\}$. This subnet will be covered by every MG-reduction. We call such a Petri Net pseudo-live: a pseudo-live marking is a marking such that some, but not all, transitions are live.

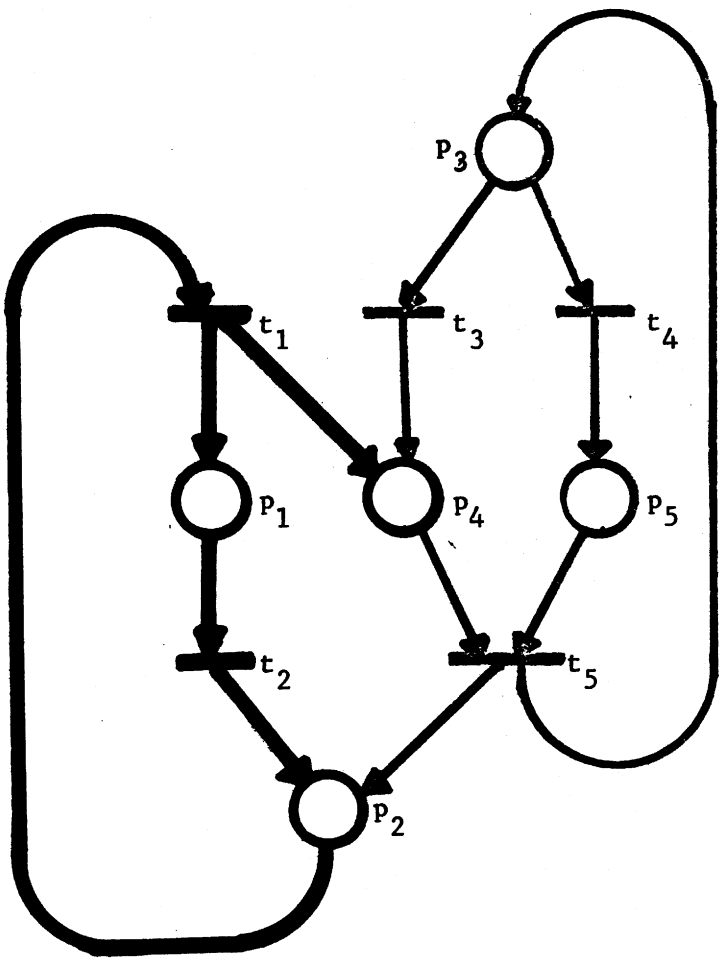
MG-reduction no. 1 of Example 4:



unallocated arc: $p_3 \cdot t_3$

The MG-reduction is not a Marked Graph.

MG-reduction no. 2 of Example 4:

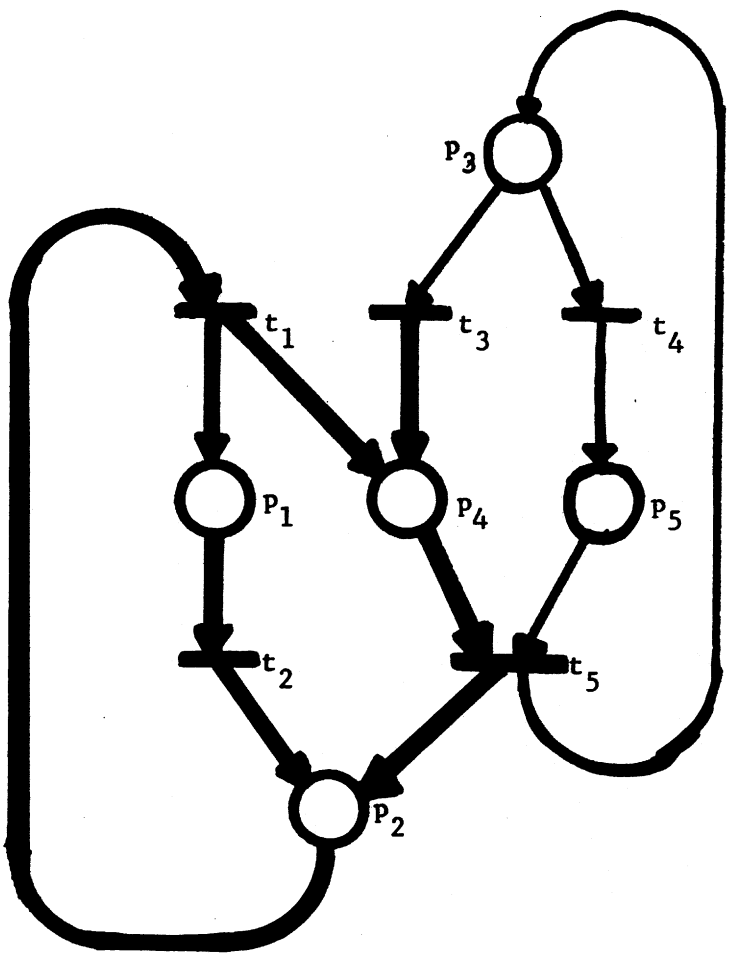


unallocated arc: $p_3 \cdot t_4$

The MG-reduction is not strongly connected.

The two MG-reductions cover everything except t_3.

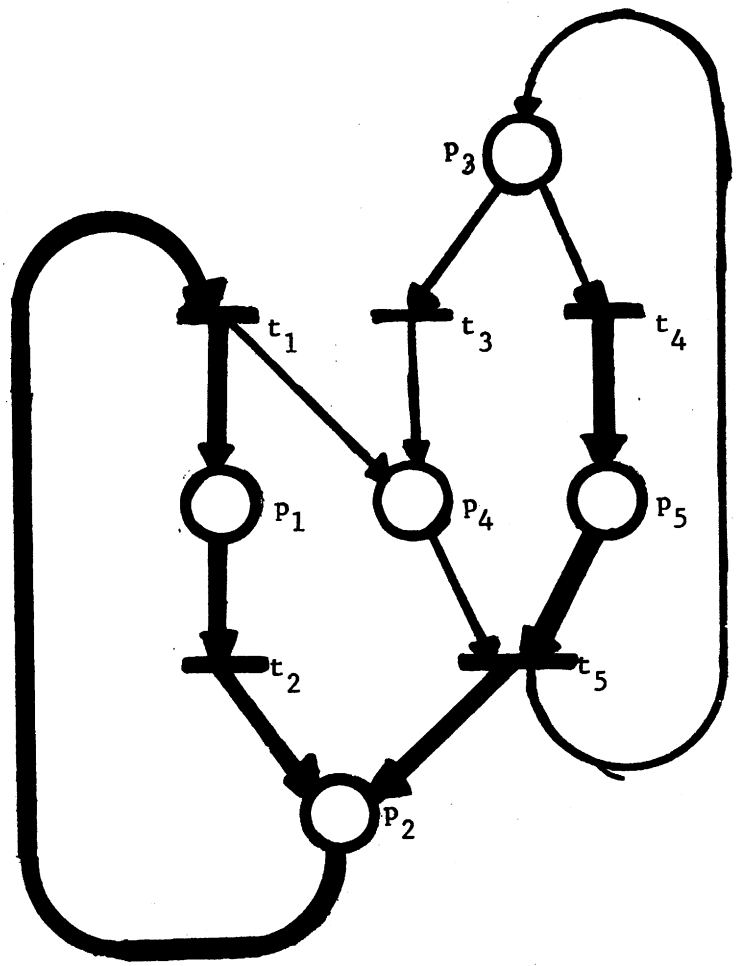
SM-reduction no. 1 of Example 4:



unallocated arc: $p_5 \cdot t_5$

The reduction is not a State Machine and not strongly connected.

SM-reduction no. 2 of Example 4:



unallocated arc: $p_4 \cdot t_5$

The reduction is not strongly connected.

The two SM-reductions cover everything except p_3.

This empty page was substituted for a blank page in the original document.

CHAPTER 6

Application of the Mathematical Results

In this chapter, we present the full decomposition of the example of a Well-Formed Production Schema shown in 2.3.

The next pages show first a reproduction of the example and the corresponding Petri Net. The labels on the Production Schema indicate the corresponding Petri Net elements. Some contractions have been performed in the translation process, as suggested in 2.4. We also have used only one transition to represent the two operations labeled j and j' in the Production Schema; this of course does not change the structure of dependencies.

We then present all SM-reductions superimposed in bold on the original net. For each reduction, we indicate the SM-allocation by crossing out the unallocated arcs.

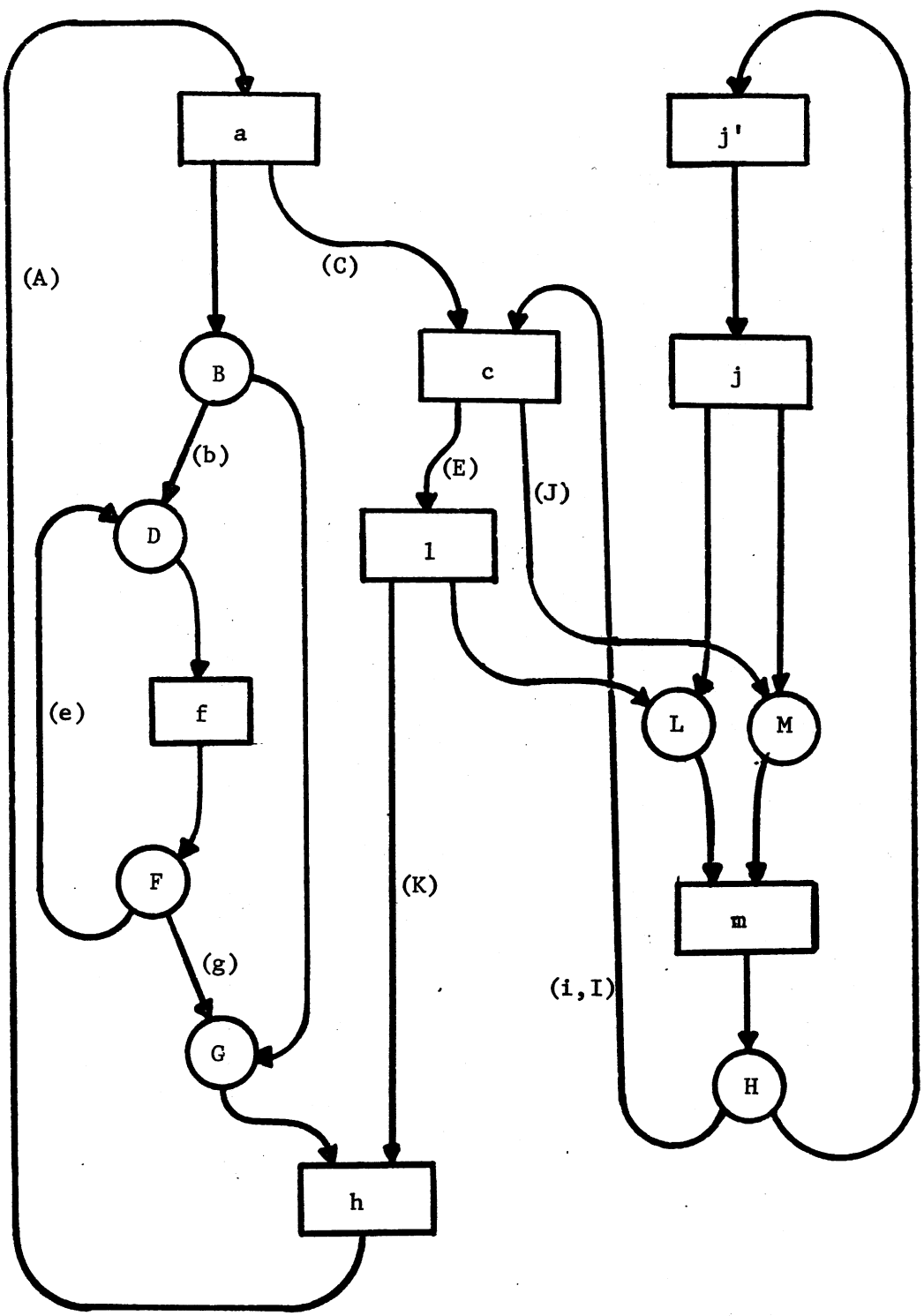
We record the progress of the reduction algorithm by numbering the elements as they are eliminated. The unallocated places, disappearing at step 1 (cf 5.3), are labelled (1). The transitions eliminated by the first application of step 2 are labelled (2); those eliminated by the n^{th} application of step 2 are labelled (2n). The places eliminated by the n^{th} application of step 3 are labelled (2n + 1).

Since there are three two-input transitions, and all other transitions have a single input place, the unallocated arcs will be chosen from three pairs of arcs. We therefore expect eight (2^3) possible SM-reductions.

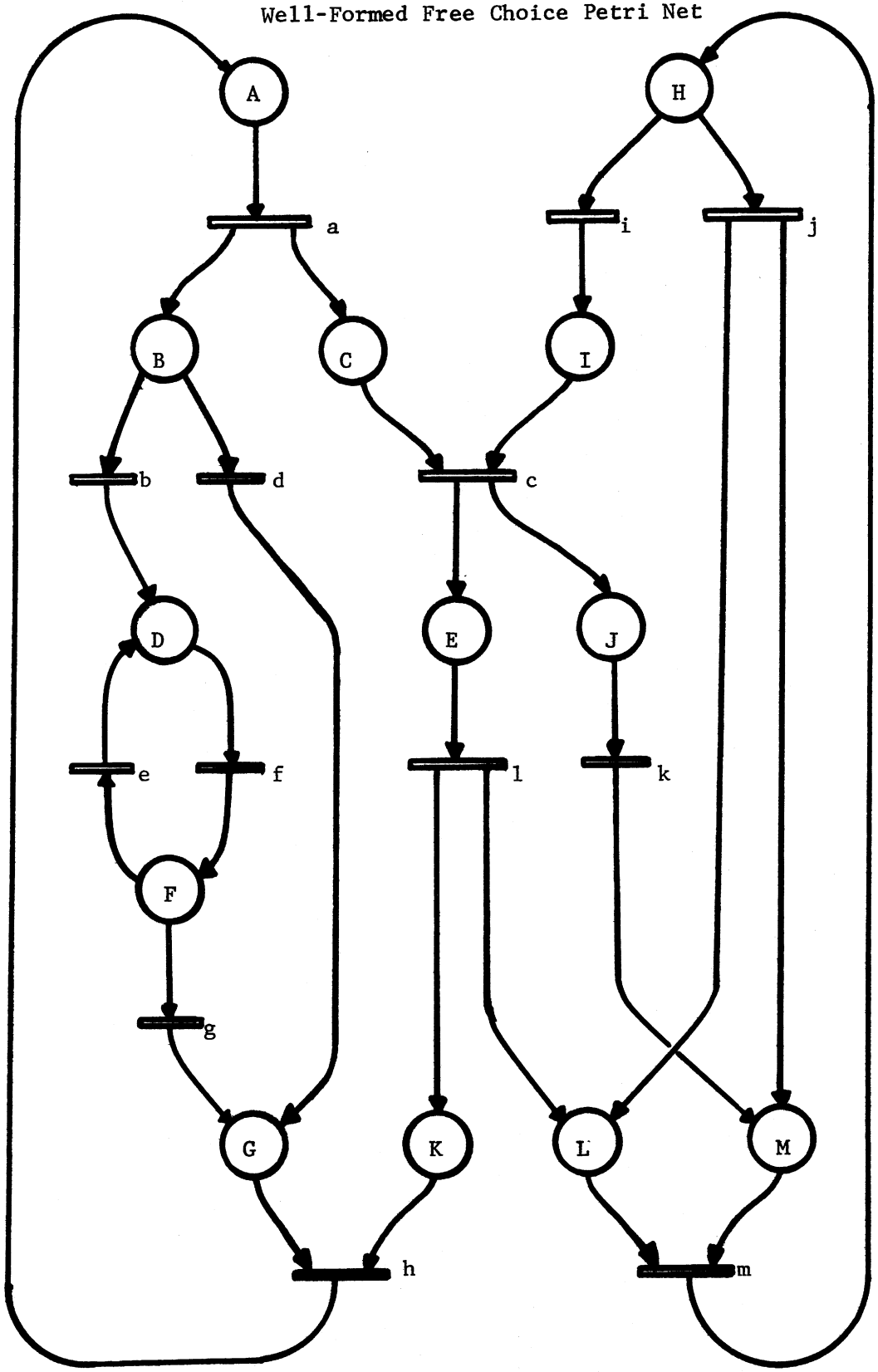
However, two different SM-allocations may yield the same reduced net. This is illustrated in the first example (SM-reduction No. 1): We notice that the choice at transition c eliminates transition m on move (4), and this independently from the choice made at m. Hence, the choice between L and M for the allocation at m is irrelevant: The two allocations yield the same reduced net. The same applies to SM-reduction No. 4.

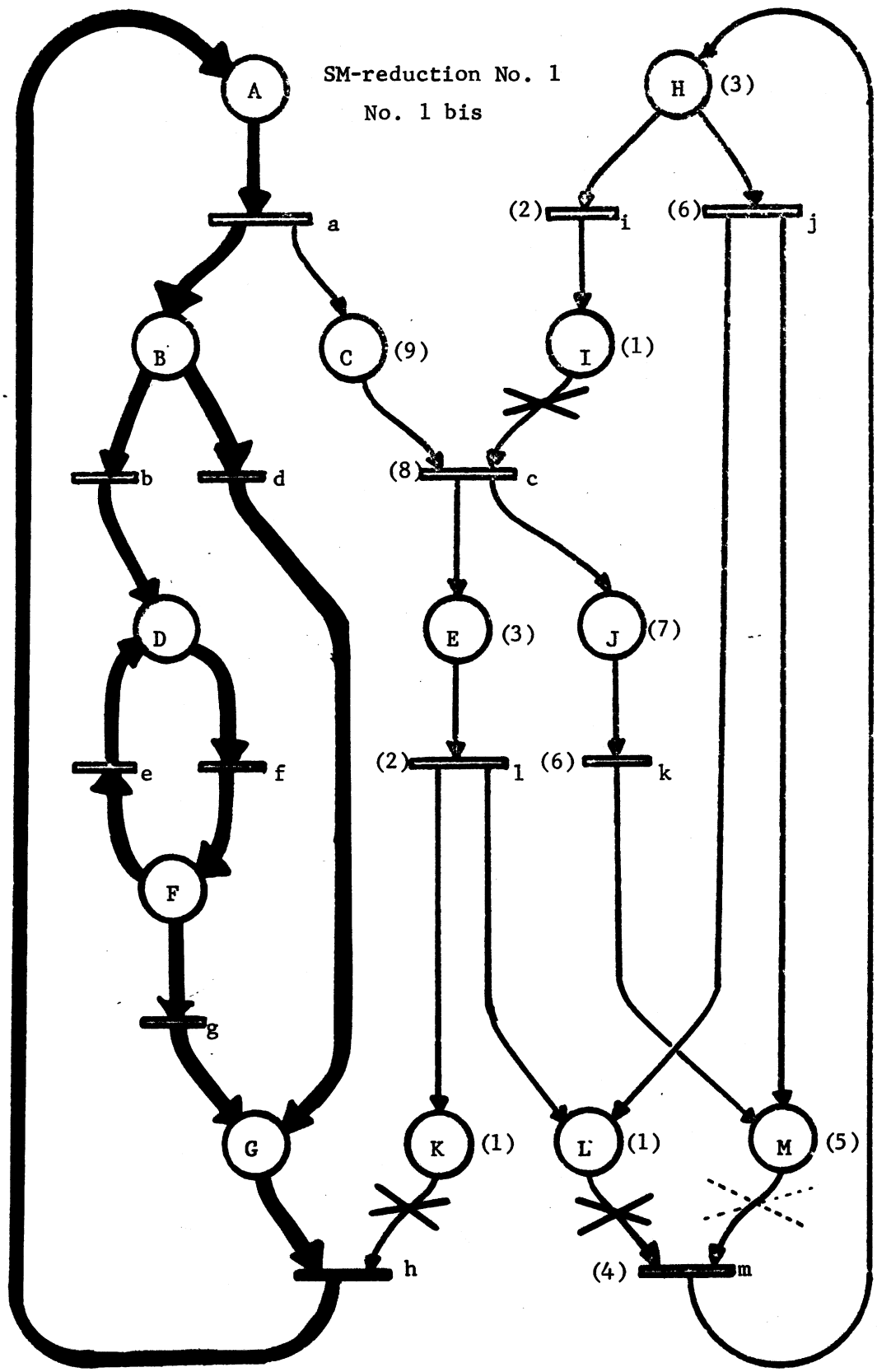
In SM-reductions Nos. 5 and 6, we also notice a multiple-input transition, namely h, that has been deleted. However, this is due to the combined choice at c and h; if we allocated G to h instead of K, we do not delete h (SM-reductions Nos. 2 and 3).

Well-Formed Production Schema



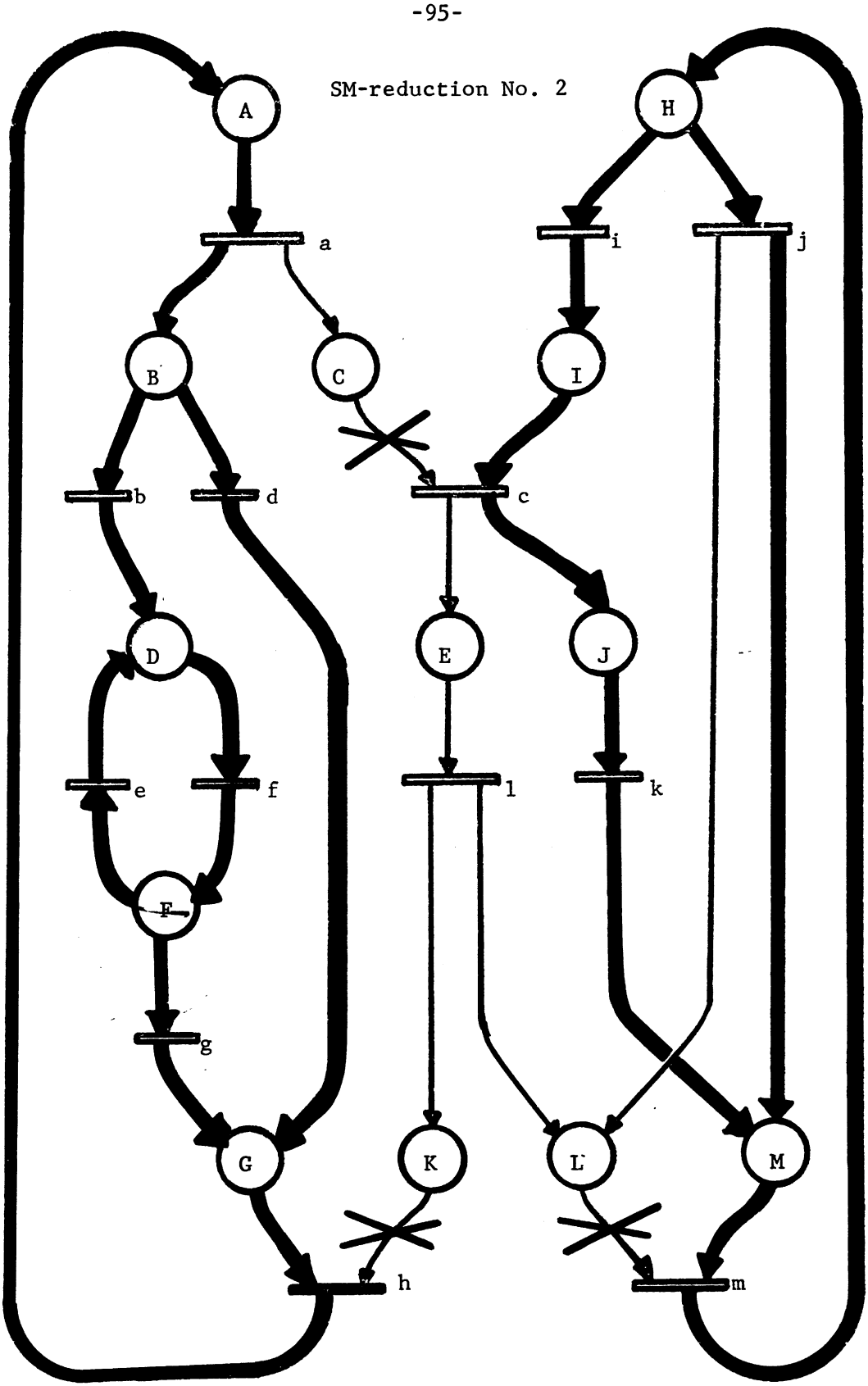
Well-Formed Free Choice Petri Net



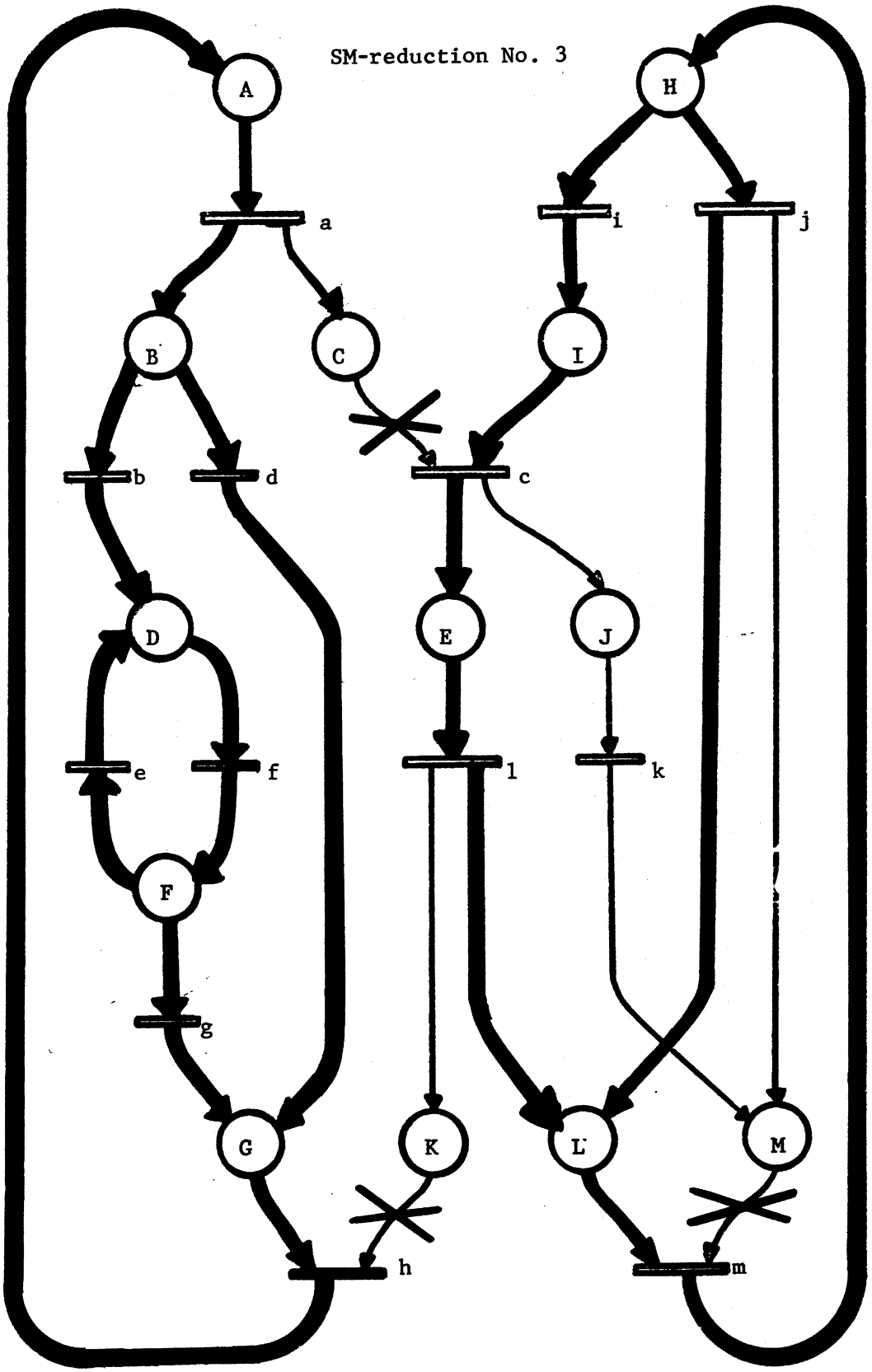


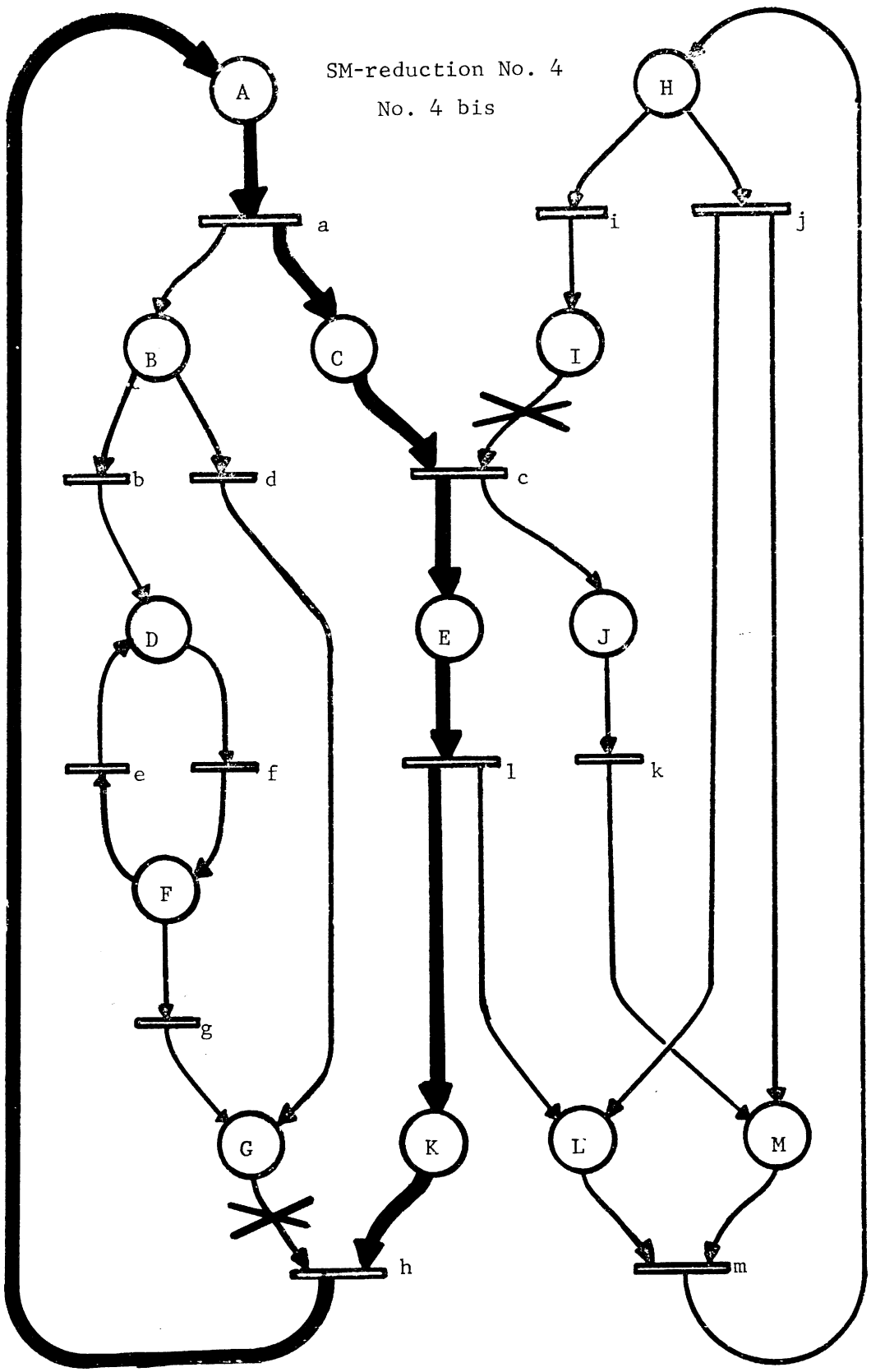
unallocated arcs: K · h, I · c, L · m (for 1 bis: M · m)

SM-reduction No. 2

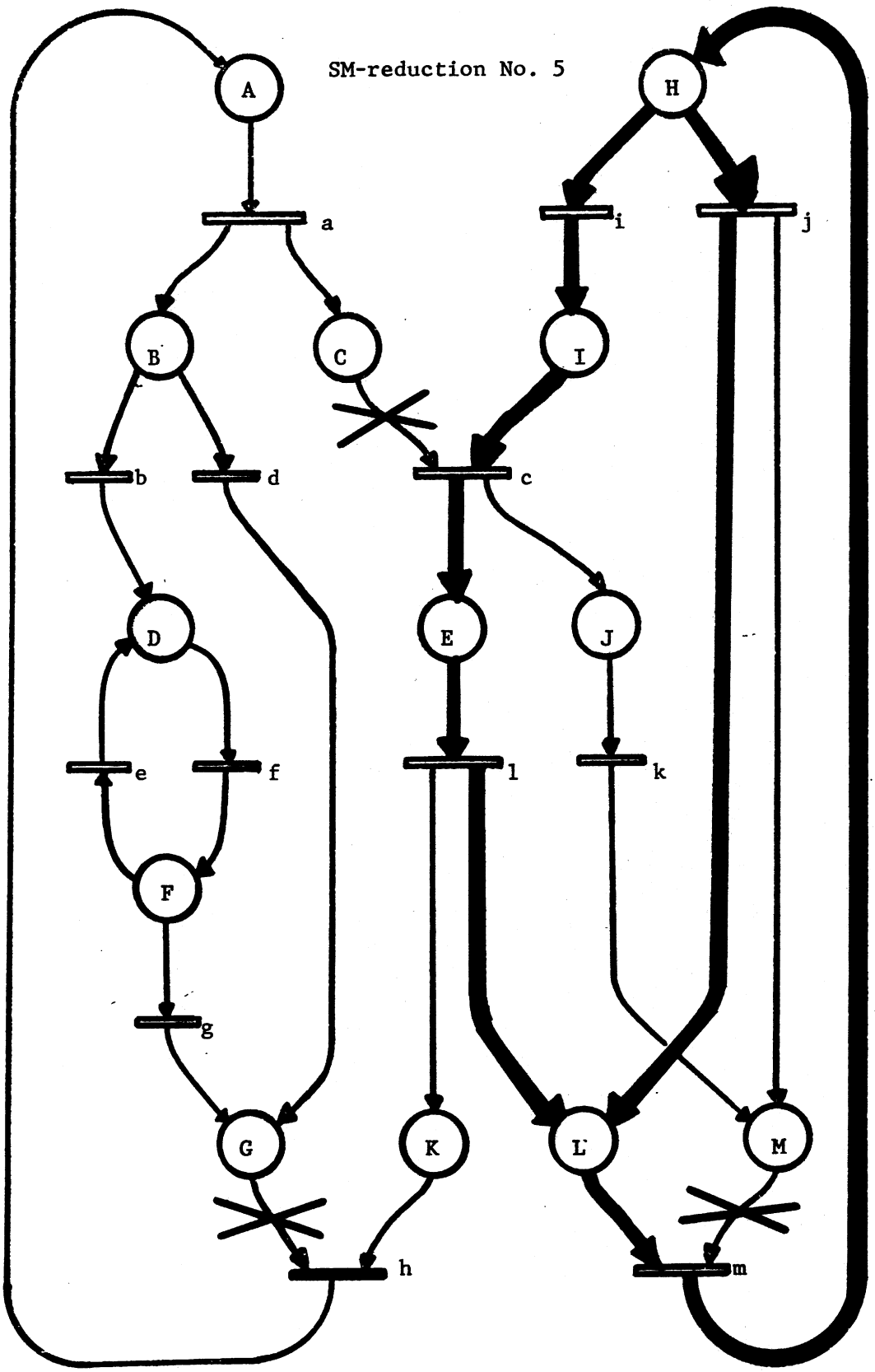


SM-reduction No. 3

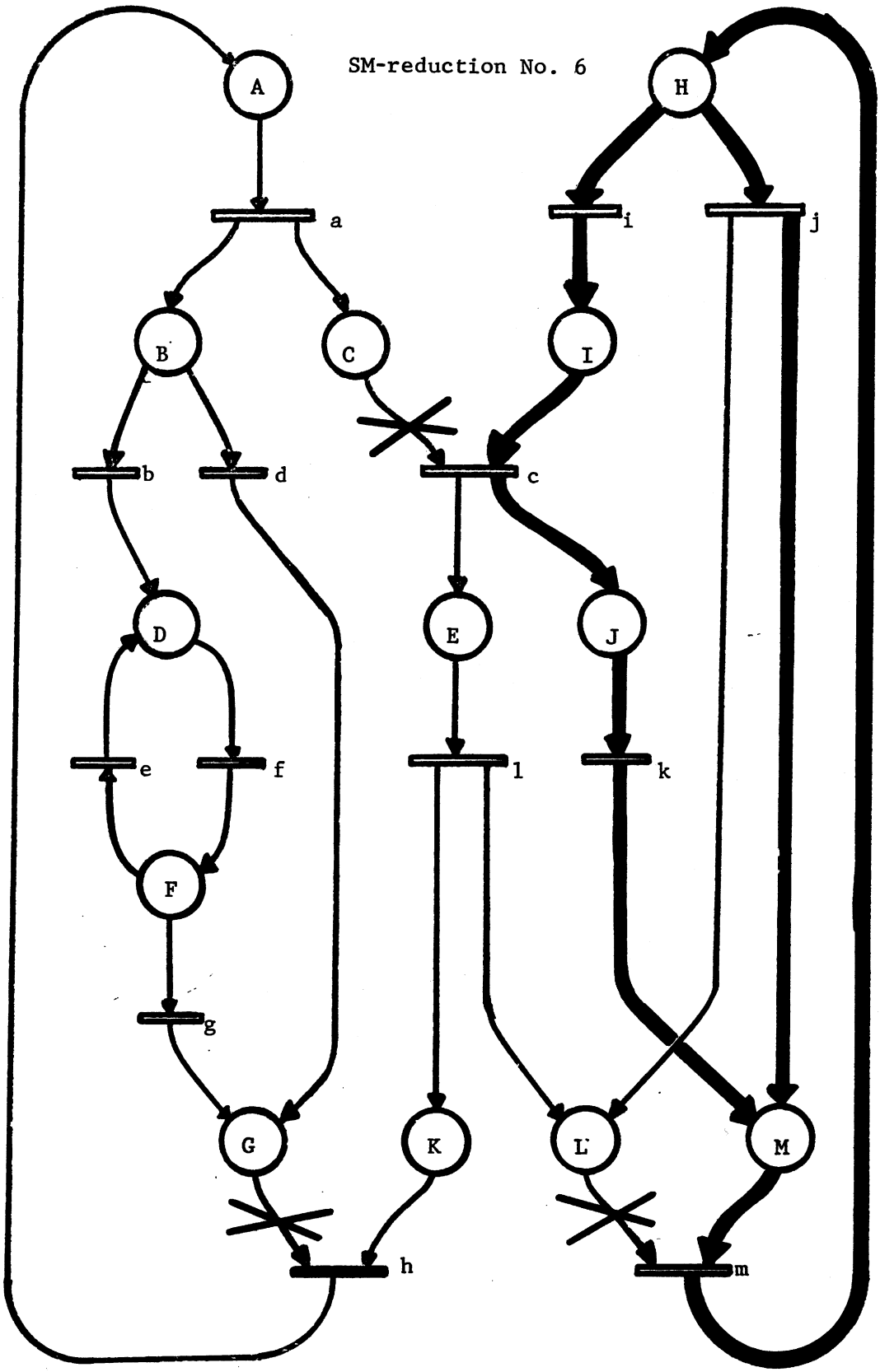



SM-reduction No. 4
No. 4 bis
A
H
a
i
j
B
C
I
b
d
c
D
E
J
e
f
l
k
F
g
G
K
L
M
h
m

SM-reduction No. 5



SM-reduction No. 6



The SM-allocation of SM-reduction 1 is formally the function B, consisting of the set of pairs $\langle x, B(x)\rangle$:

argument: $x \in \Sigma$	a	b	c	d	e	f	g	h	i	j	k	l	m
value: $B(x) \in {}^{\bullet}x$	A	B	C	B	F	D	F	G	H	H	J	E	L

We get the same reduced net by replacing the argument-value pair $\langle m, L\rangle$ by $\langle m, M\rangle$. We distinguish the allocations yielding SM-reduction No. 1 by calling them SM-allocation No. 1 and No. 1 bis respectively.

We also note that a reduction may consist of several disjoint parts. This should not be surprising, and the warning on page was given with this in mind. It is simply convenient not to distinguish between the two interpretations of "strongly connected;" context usually makes the difference clear when it is relevant (when talking about minimal deadlocks for example). We shall say individual SCSM if we want to emphasize one component.

The individual SCSM's (the minimal deadlocks) are the SM-reductions Nos. 1, 4, 5, 6. SM-reduction Nos. 2 and 3 are combinations of 1 and 6 respectively 5. In this net, all minimal deadlocks are required to cover the net. In terms of reductions, only three are required: 2, 4, and 5 for example.

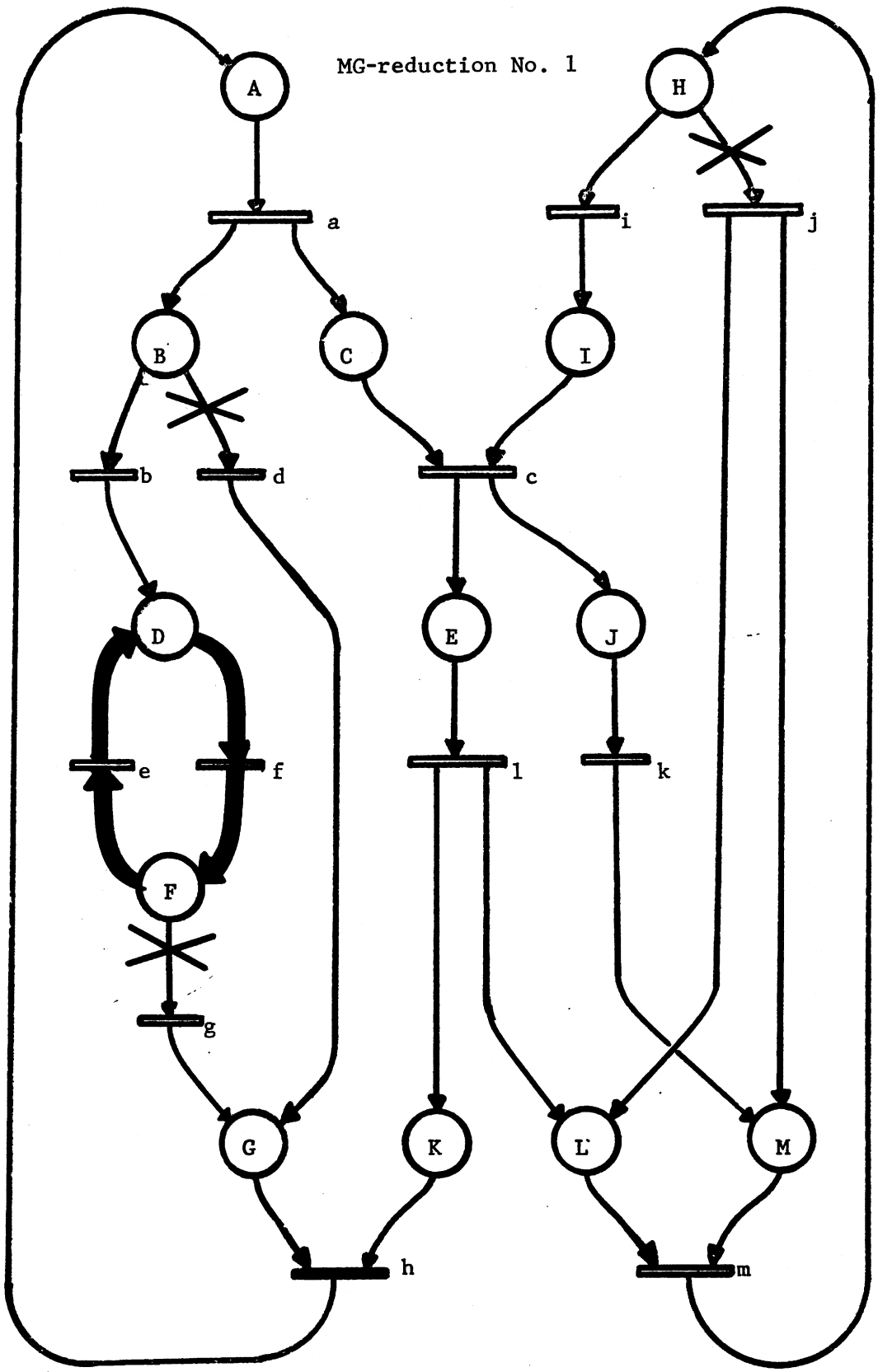
There are 8 SM-allocations (the product of the number of input arcs over all transitions) yielding 6 different SM-reductions and 4 individual SCSM's. Note also that the union of SM-reductions No. 3 and No. 6 covers all transitions, but leaves out places C and K.

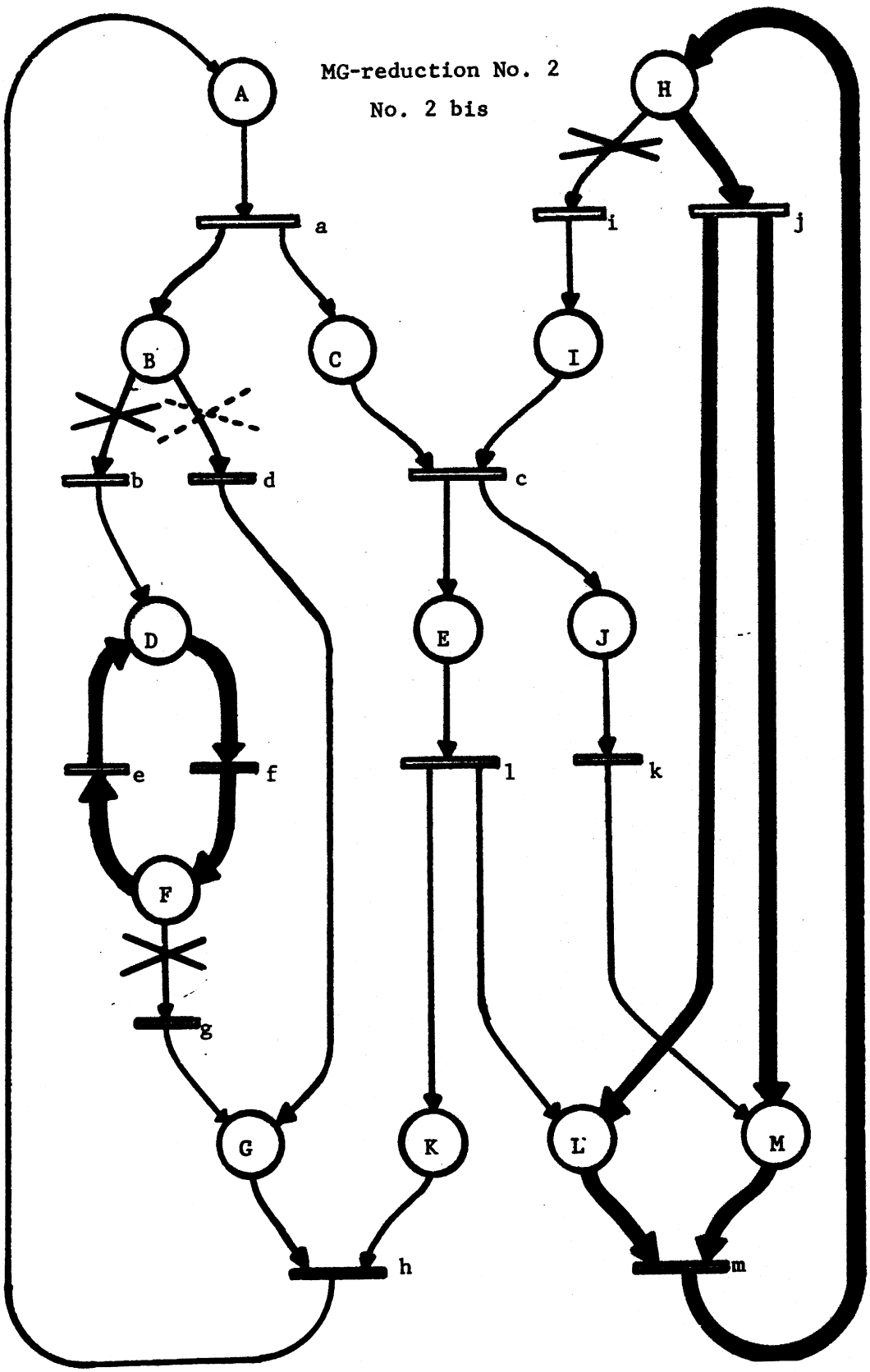
From the SM-decomposition we can infer a few facts about a possible live-and-safe marking.

- Since there are four minimal deadlocks, and each has at least one place that appears in no other individual SCSM (four such places are G, K, L, M for example), the maximum number of tokens in the net is four.

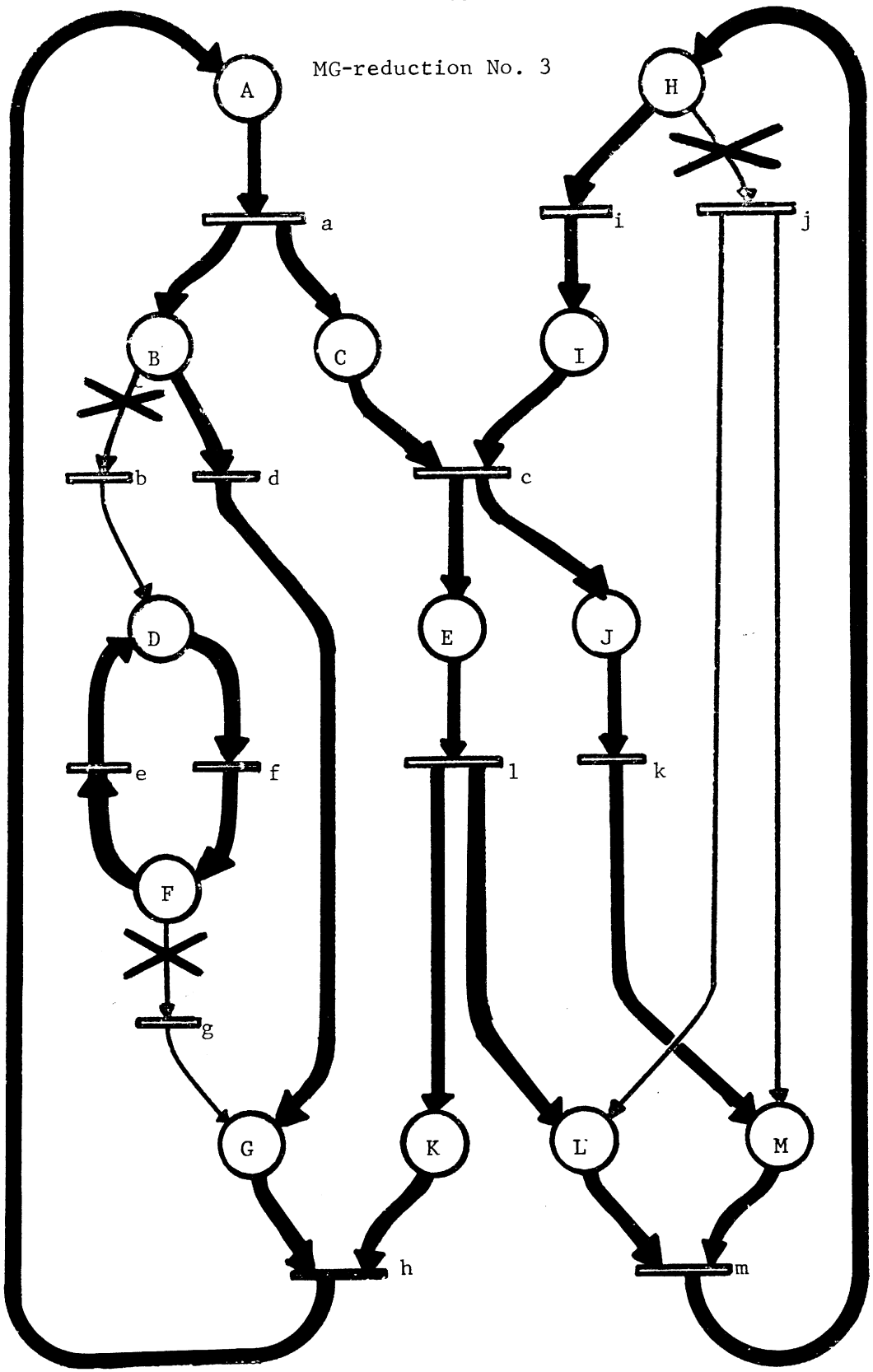
- Since no place is shared by more than two individual SCSM's, but A is shared by two SCSM's and H by the other two, the minimum number of tokens in a live and safe marking is two. It is also easy to see that there is only one live-and-safe marking class, determined by the initial marking {A, H} for example.

MG-reduction No. 1

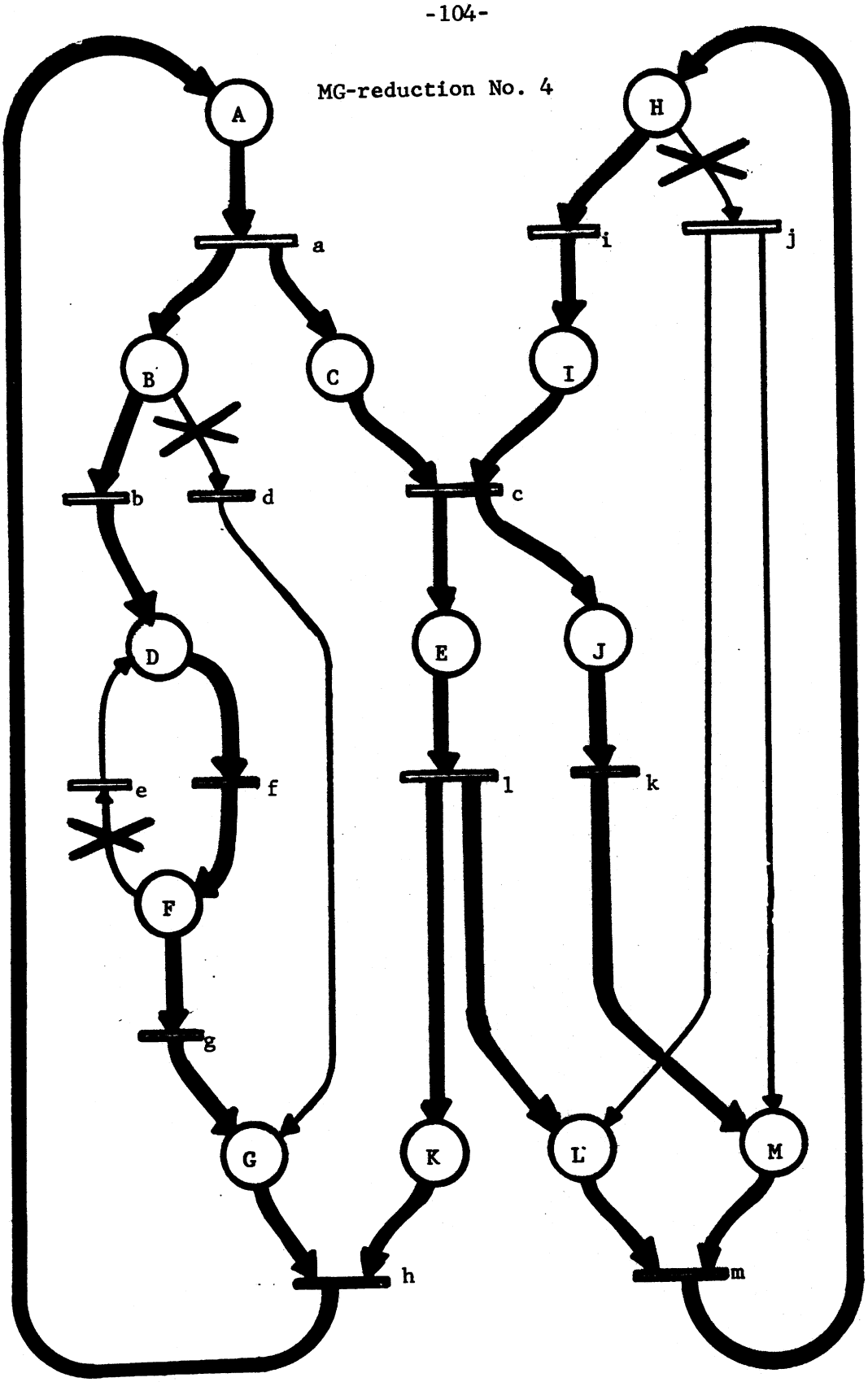



MG-reduction No. 2
No. 2 bis
A
H
a
i
j
B
C
I
b
d
c
D
E
J
e
f
l
k
F
g
G
K
L
M
h
m

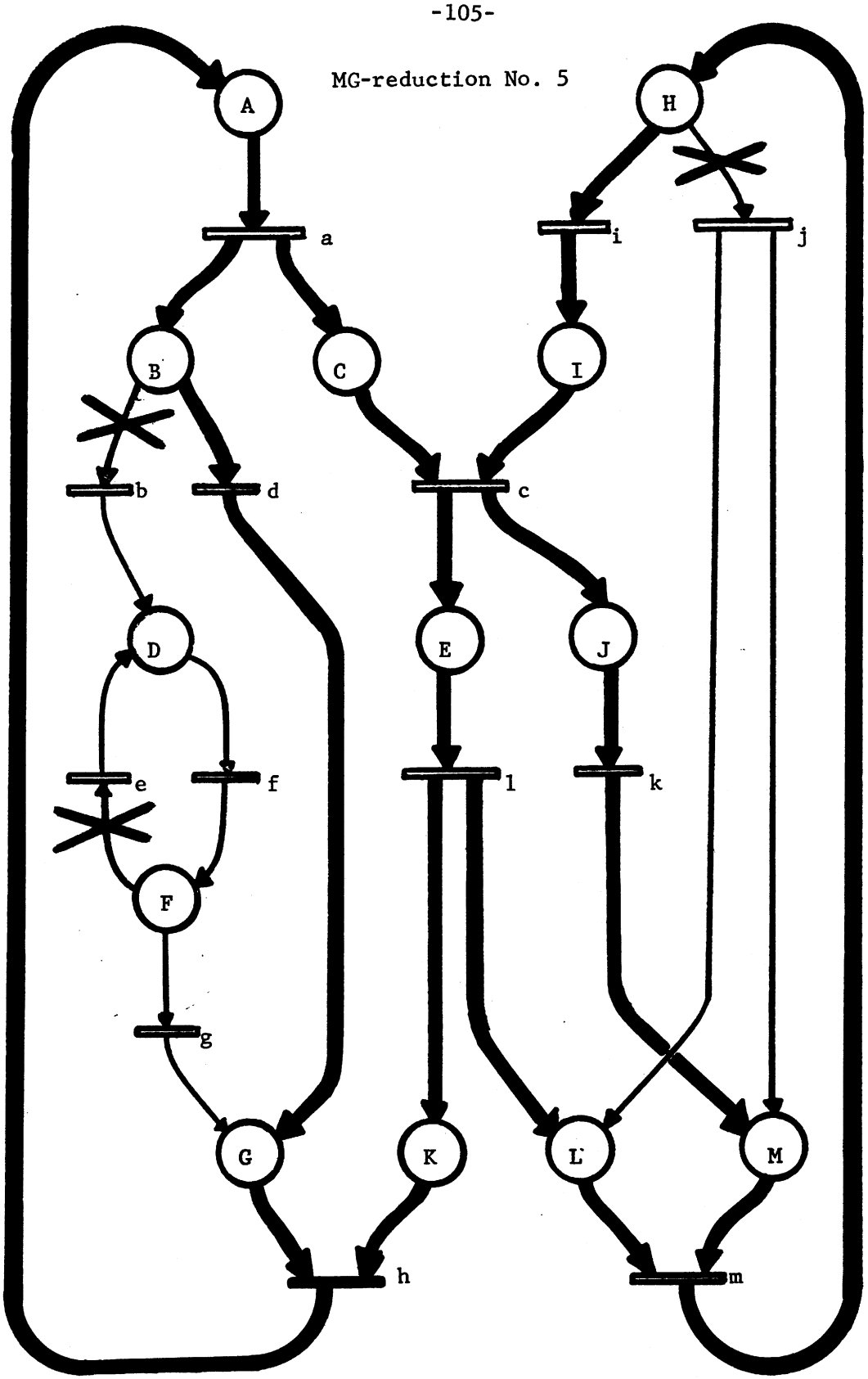
MG-reduction No. 3



MG-reduction No. 4

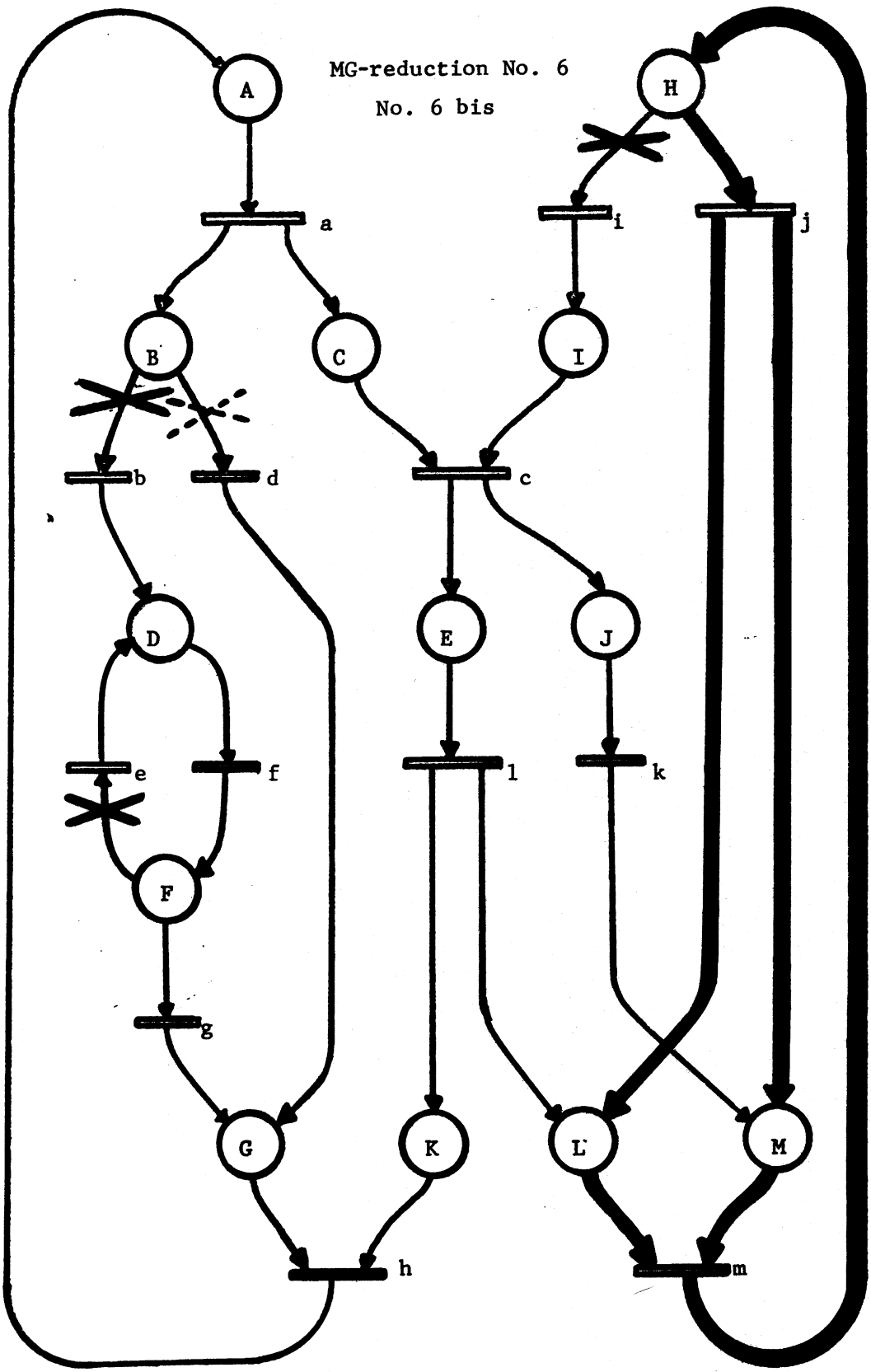


MG-reduction No. 5



MG-reduction No. 6

No. 6 bis



The MG-reductions have been constructed in an analogous way. Again, unallocated arcs have been crossed out. The MG-allocation for MG-reduction No. 1, for example, is the function A, consisting of the set of pairs $\langle x, A(x)\rangle$:

argument: $x \in \Pi$	A	B	C	D	E	F	G	H	I	J	K	L	M
value: $A(x) \in x^{\bullet}$	a	b	c	f	l	e	h	i	c	k	h	m	m

The unallocated transitions are d, g, j.

Much of what has been said about SM-reductions can be said about MG-reductions. We again have 8 MG-allocations (product of the number of output arcs over all places) yielding 6 distinct MG-reductions and 4 individual SCMG's: reductions Nos. 1, 4, 5 and 6.The coincidence with SM-reductions is totally fortuitous (even the fact that MG-reductions Nos. 2 and 3 are composed of reductions No. 1 plus 6 and 6, respectively); to show this, it is enough to imagine an additional choice for B, going to F via a new transition n, for example. Now we would have 12 MG-allocations, and we would get more SCMG's , but the only change to SM-reductions would be that the individual SCSM No. 1 would look different in SM-reductions Nos. 1, 2 and 3.

Note that MG-reduction No. 4 covers all places by itself, but transitions e, d and j are not covered. A complete MG-covering would be 2, 4, 5 for example, consisting of all four individual SCMG's.

We can consider a covering by SCSM's as a set of State Machines communicating by exchanging synchronization signals via shared transitions h, m and c. Since we interpret the net as a representation of some production facility, these transitions correspond to points where one process must wait for another. If two transitions, say c and j, belong to the same individual SCSM, they may represent facilities using the same resources, since they will never compete for common resources.

The decomposition into Marked Graphs shows concurrency among the composing State Machines. But it also shows possible complete independencies. For instance, MG-reductions Nos. 2 and 3 consist of two disjoint SCMG's. The two SCMG's of MG-reduction No. 2, however, cannot operate concurrently, because the individual SCMG No. 5 intersects the individual SCSM No. 1 containing SCMG No. 1: SCSM No. 1 would contain two tokens. But all four individual SCSM's are needed for the covering, and hence all must be one-token SCSM's.

On the other hand, this restriction does not apply to MG-reduction No. 2, where the two components are indeed totally independent of each other.

An interesting result for production facilities obtained from the Well-Formedness Theorem in connection with MG-reductions is the following:

> If a production facility "works properly" for every constant set of decisions (constant predicates for multiple choice places) (i.e. every MG-reduction is LS, hence SCMG) then it "works properly" for any dynamic choice (i.e. the net is LS).

CONCLUSION

This thesis has extended the structural analysis methods to concurrent systems with decisions and conflicts. Before, most work in this area was concerned with marked-graph type schemata [3 , 12]. Baer, Bovet and Estrin restricted themselves to directed acyclic bilogical (i.e. conjunctive and disjunctive nodes) graphs [1]. The legality they refer to corresponds to our Well-Formedness; in that sense this thesis extends their work to directed cyclic bilogical graphs.

The concept of decomposition of Petri Nets seems very promising. It permits the identification of meaningful subsystems and their interconnections in a complex system. It may be used to enhance structural transparency in the synthesis of complex concurrent systems. It also provides criteria for the hang-up free interconnection of State Machines, and sheds a new light on the results about the interconnections of determinate systems obtained by Patil [17].

An interesting field of future research is the semantic interpretation of the decomposition results, notably the significance of the dual coverings -- by Marked Graphs and by State Machines -- of Petri Nets. We expect a strong influence in this field from recent research on the semantics of Petri Nets, by Holt [11].

A different approach to decomposition has been made by Furtek [8]. It is based on an analysis of the information flow along arcs that governs the token flow at firings. Combining the two approaches should prove very fruitful.

The next step will be to extend our results and methods to wider classes of Petri Nets. Simple Nets seem to be the next target, and a few results similar to those for Free Choice Nets have already been obtained for Simple Nets. Ultimately, we hope to gain a full understanding of the structural properties of General Petri Nets, and we expect that some of the tools provided in this thesis will be useful to that effect. If we get theorems and Live-and-Safeness criteria similar to those expressed here for a larger class of Petri Nets, we will be able to extend the definition of Production Schemata to represent and analyze an even larger class of Systems.

This empty page was substituted for a blank page in the original document.

INDEX

	Section	Page
acknowledgement	2.2	25
allocated transitions, A( )	4.1.2	49
allocation, A	4.1.2	49
A-sequence	4.1.2	51
assembly	2.2	21
asynchronous	2.2	23
backflow	2.2	25
backward path	5.3	74
blank	1.1	11
blank (set of places)	3.3	42
boundary	5.2	70
buffer	2.2	24
circuit-free	4.1.2	49
Closed Consistent Subnet	1.4.2	18
Closed Consistent Subnet (formal)	5.2	69
collector	2.3	26
concurrent	Introd.	7
condition	1.2	13
conflict	Introd.	7
conflict-free	5.4	76
conjunctive	2.1	21
consistent	5.3	73
Consistent Subnet	1.4.2	18
Consistent Subnet (defined by a set of places)	4.2.1	56
Consistent Subnet (defined by a set of transitions)	5.2	69
contraction	2.4	33
Commoner's Liveness Theorem	4.1	55
corollary of Theorem 6	4.3	63
covering	1.4.2	19
covering (formal)	4.2.1	57

	Section	Page
deadlock	1.4.2	17
deadlock (formal)	3.3	42
decision	Introd.	7
delete	5.3	71
disassembly	2.2	22
disjunctive	2.3	25
dual	5.2	67
dynamic choice	6	108
empty (set of places)	3.3	42
event	1.2	13
excluded transition, $\bar{A}(\ )$	4.1.2	50
firable	1.1	11
firing	1.1	11
firing (formal)	3.1	39
firing sequence, σ	3.1	40
Flow of Control	2.1	21
Flow of Objects	2.1	21
Free Choice Petri Net, FC	1.3	14
Free Choice Petri Net, FC (formal)	3.2	42
free interpretation	2.3	28
head (of a circuit-free allocation)	4.1.2	52
hold (condition)	1.2.3	13
individual SCSM	6	100
individual SCMG	6	107
initial marking	1.1	11
input place	1.1	11
killing sequence	4.1.2	49

	Section	Page
Lemma 1	4.1.1	43
Lemma 2	4.1.1	44
Lemma 3	4.1.2	50
Lemma 4	4.1.2	51
Lemma 5	4.2.3	59
Lemma 6	5.2	68
Lemma 7	5.3	73
Lemma 8	5.4	77
Lemma 9	5.4	77
Lemma 10	5.4	78
Lemma 11	5.4	80
link	2.4	31
live	1.2	12
live (formal)	3.1	40
Live-and-Safe, LS	4.2	56
Live-and-Safeness Theorem	4.3	62
LSFC (Live-and-Safe Free Choice net)	4.2.2	58
liveness	1.2	12
malfunction	2.2	24
marked	1.1	11
marked (set of places)	3.3	42
Marked Graph, MG	1.3	14
Marked Graph, MG (formal)	3.2	42
marked trap	4.1	43
marking	1.1	11
marking, M (formal)	3.1	39
marking class	1.1	11
marking class, $\vec{M}$ (formal)	3.1	40
maximal trap	4.1.2	50
MG-allocation	5.4	75
minimal deadlock	1.4.2	19

	Section	Page
non-determinacy	2.2	24
occur	1.2	13
Open Consistent Subnet	1.4.2	18
Open Consistent Subnet (formal)	5.2	69
operation	2.1	21
output place	1.1	11
Petri Net	1.1	11
Petri Net (formal)	3.1	39
pipeline	2.2	24
place	1.1	11
place (formal)	3.1	39
primal	5.2	67
Production Schema	2.1	21
pseudo-live	5.6	85
reachable	1.1	11
reduced net	5.3	71
reduction method	5.3	70
reverse	5.2	66
reverse-dual	5.2	67
safe	1.2	13
safe (formal)	3.1	40
safeness	1.2	12
SC, strongly connected	5.3	71
SCMG, Strongly Connected Marked Graph	5.4	76
SCSM, Strongly Connected State Machine	4.2.2	57
self-reverse-dual	5.2	68
Simple Petri Net	1.3	14
SM-allocation	5.3	71

	Section	Page
SM-reduced net	5.3	72
SM-reduction	5.3	73
State Machine, SM	1.3	14
State Machine, SM (formal)	3.2	42
structural	Introd.	7
subclasses	1.3	14
subnet	1.4.2	18
subnet (formal)	3.1	41
switch	2.3	25
synchronization signal	6	107
syntactical	1.3	14
Theorem 1	4.1.1	49
Theorem 2	4.1.2	55
Theorem 3	4.1.2	55
Theorem 4	4.2.2	57
Theorem 5	4.3	61
Theorem 6	4.3	62
Theorem 7	5.3	73
Theorem 8	5.4	81
Theorem 9	5.5	81
timing	2.2	23
token	1.1	11
transition	1.1	11
transition (formal)	3.1	39
trap	1.4.2	17
trap (formal)	3.3	39
uninterpreted	2.2	24
union (of Consistent Subnets)	4.2.1	56
unpredictability	2.2	24

	Section	Page
vertices	3.1	39
Well-Formed, WF	5.1	65
Well-Formedness corollary	5.1	65
Well-Formedness Theorem	5.5	81
Well-Formed Production Schema	2.4	35
$\langle \Pi, \Sigma \rangle$	3.1	41
$\langle \Pi, \Sigma, \cdot \rangle$	3.1	39

REFERENCES

1. Baer, J. L., D. P. Bovet, and G. Estrin. Legality and other properties of graph models of computations. J. of the ACM, Vol. 17, No. 3 (July 1970), pp 543-554.

2. Berge, C. The Theory of Graphs and its Applications. John Wiley and Sons., Inc., New York 1962.

3. Bruno, J. and S. M. Altman. Asynchronous control networks. IEEE Conference Record. Tenth Annual Symposium on Switching and Automata Theory, 1969, pp 61-73.

4. Commoner, F. Private communication.

5. Commoner, F., A. W. Holt, S. Even, and A. Pnueli. Marked directed graphs. J. of Computer and System Sciences, Vol. 5 (October 1971), pp 511-523.

6. Dennis, J. B. Modular, asynchronous control structures for a high performance processor. Record of the Project MAC Conference on Concurrent Systems and Parallel Computation, ACM, New York 1970, pp 55-80.

7. Dijkstra, E. W. Co-operating sequential processes. Programming Languages, F. Genuys, Ed., Academic Press, New York 1968. [First published as Report EWD 123, Department of Mathematics, Technological University, Eindhoven, The Netherlands, 1965.]

8. Furtek, F. Modular Implementation of Petri Nets. S.M. Thesis, Department of Electrical Engineering, M.I.T., September 1971.

9. Genrich, H. J. Einfache Nicht-sequentielle Prozesse (Simple Nonsequential Processes). Bericht Nr. 37, Gesellschaft für Mathematik und Datenverarbeitung, Bonn 1971.

10. Holt, A. W. Final Report of the Information System Theory Project. Technical Report RADC-TR-68-305, Rome Air Development Center, Griffiss Air Force Base, New York, 1968.

11. Holt, A. W. Course Notes for 6.541. Department of Electrical Engineering, M.I.T., Fall 1971.

12. Holt, A. W. and F. Commoner. Events and Conditions. (In three parts), Applied Data Research, New York 1970.

13. Karp, R. M. and R. E. Miller. Parallel program schemata: A mathematical model for parallel computation. I.E.E.E. Conference Record, Eighth Annual Symposium on Switching and Automata Theory, October 1967, pp 55-61. Also: IBM Research Paper RC-2053, No. 10496, April 1968.

14. Muller, D. E. and W. S. Bartky. A theory of asynchronous circuits. Proceedings of an International Symposium on the Theory of Switching, April 1957. Annals of the Computation Laboratory of Harvard University, Vol. 29, Part 1, Harvard University Press, Cambridge, Massachusetts, 1959, pp 204-243.

15. Paterson, M. S. and C. W. Hewitt. Comparative schematology. Record of the Project MAC Conference on Concurrent Systems and Parallel Computation, ACM, New York 1970, pp 119-127.

16. Patil, S. S. Coordination of Asynchronous Events. Report MAC-TR-72, Project MAC, M.I.T., Cambridge, Massachusetts, June 1970.

17. Patil, S. S. Closure properties of interconnections of determinate systems. Record of the Project MAC Conference on Concurrent Systems and Parallel Computation, ACM, New York 1970, pp 107-116.

18. Petri, C. A. Communication With Automata. Supplement 1 to Technical Report RADC-TR-65-377, Vol. 1, Griffiss Air Force Base, New York 1966. [Originally published in German: Kommunikation mit Automaten, University of Bonn, 1962.]

19. Rodriguez, J. E. A Graph Model for Parallel Computation. Report MAC-TR-64, Project MAC, M.I.T., Cambridge, Massachusetts, 1969.

20. Shapiro, R. M. and H. Saint, Representation of Algorithms. RADC TR-69-313, Vol. II, September 1969.

21. Slutz, D. R. The Flow Graph Schemata Model of Parallel Computation. Report MAC-TR-53, Project MAC, M.I.T., Cambridge, Massachusetts, 1968.

More references about these and other related subjects can be found in the Bibliography of the Record of the Project MAC Conference on Concurrent Systems and Parallel Computation, ACM, New York 1970, p. 187.

This empty page was substituted for a blank page in the original document.

CS-TR Scanning Project
Document Control Form

Date : 1/23/96

Report # LCS-TR-94

Each of the following should be identified by a checkmark:
Originating Department:

☐ Artificial Intellegence Laboratory (AI)
☒ Laboratory for Computer Science (LCS)

Document Type:

☒ Technical Report (TR) ☐ Technical Memo (TM)
☐ Other:_____

Document Information

Number of pages: 120 (126-IMAGES)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:
☐ Single-sided or
☒ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☐ Laser Print
☐ InkJet Printer ☒ Unknown ☐ Other:_____

Check each if included with document:
☒ DOD Form ☐ Funding Agent Form ☒ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages (by page number): _____

Photographs/Tonal Material (by page number): _____

Other (note description/page number):

Description :	Page Number:
IMAGE MAP: (1-120) UN#'ED TITLE PAGE, 2-19, UN#'ED BLANK, 21-35, UN#'ED BLANK, 37-UN#'ED BLANK, 39-63, UN#BLK, 65-89, UN#BLK, 91-109, UN#BLANK, 111-119, UN#BLANK	(121-126) SCAN CONTROL, COVER, DOD, TRGT'S (3)

Scanning Agent Signoff:
Date Received: 1/23/96 Date Scanned: 1/24/96 Date Returned: 1/25/96

Scanning Agent Signature: Michael W. Cook

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Massachusetts Institute of Technology
Project MAC

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP
NONE

3. REPORT TITLE
Analysis of Production Schemata by Petri Nets

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Master of Science, Department of Electrical Engineering, February 1972

5. AUTHOR(S) *(Last name, first name, initial)*
Hack, Michel Henri Théodore

6. REPORT DATE
February 1972

7a. TOTAL NO. OF PAGES
120

7b. NO. OF REFS
21

8a. CONTRACT OR GRANT NO.
N00014-70-A-0362-0001

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
MAC TR-94 (THESIS)

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. AVAILABILITY/LIMITATION NOTICES
Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES
None

12. SPONSORING MILITARY ACTIVITY
Advanced Research Projects Agency
3D-200 Pentagon
Washington, D. C. 20301

13. ABSTRACT Petri nets provide a powerful graphical tool for representing and analyzing complex concurrent systems. Properties such as hang-up freeness, determinacy, conflict, concurrency and dependency, can be represented and studied. The precise relationship between structural and behavioral properties, and between local and global properties is not well understood for the most general class of Petri Nets. This thesis presents such results for a restricted class of Petri Nets called Free Choice Petri Nets, and for a corresponding class of Systems called Production Schemata. Results on structural constraints guaranteeing global operation, and decompositions of complex systems into meaningful parts, are also presented.

14. KEY WORDS
Information system
Parallel schemata
Petri nets
Concurrent systems
Systemics
Structural analysis of systems

DD FORM 1 NOV 65 1473 (M.I.T.)

UNCLASSIFIED
Security Classification

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**

Scanned
Date: 1/24/1996
M.I.T. Libraries
Document Services