PERFORMANCE ANALYSIS OF TIMED PETRI NETS BY DECOMPOSITION OF THE STATE SPACE

FLOAREA SUCIU

0-612-34234-4

Abstract

Performance evaluation of systems is a very important part of system design. Modeling tools which allow the analysis of systems and their behaviors should also provide performance analysis of the modeled system as it is less costly to perform changes at the model level. Petri nets are becoming popular modeling tools that can represent and analyze concurrency, parallelism, synchronization, mutual exclusion and conflicts. However, time and space requirements of the classical approach of exhaustive generation of all possible behaviors of the system (or its state space) grow quickly with the size of the model. An alternative approach based on structural properties can be applied only to particular classes of nets.

A new way to derive performance measures for timed Petri nets is based on decomposition of the state space. This is a hybrid method that uses both reachability and structural analysis. The state space of the original net is decomposed into state spaces of smaller nets, and these smaller nets are then analyzed by the reachability analysis method. Since the nets are quite simple, reachability analysis is straightforward and cannot be affected by the "state explosion" problem. The performance indices for smaller nets are then used for performance analysis of the original net.

Acknowledgements

I would like to express my sincere thanks to my supervisor, Dr. Wlodek Zuberek for all the things he has taught me. Without his guidance, challenges and support this thesis would not have taken this final form.

I am grateful to the Petri net newsgroup and especially to Bill Henderson, Matteo Sereno, Peter Kemper, Falko Bause, Mathias Becker and Peter Ziegler who provided very useful references.

Many thanks to the faculty and staff of Computer Science department for their assistance and support throughout my program at MUN.

Finally, I would like to thank my parents for their moral support, my friends for their jokes, Millie for all the meals provided and last but not least, my husband and dear friend Tee for his love and friendship.

To my friend, Ed King,

who had the courage to walk with me

on a very hard, but beautiful spiritual journey.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Many present–day systems have become so complex that predicting their performance cannot be done without some form of mathematical modeling. Communication networks, multiprocessor distributed systems, distributed databases, but also traffic control systems, large manufacturing plants and economic systems are simple examples of such problems. A model is a representation, often in mathematical terms, of those features of the original system, which are believed to be important for the study (e.g., for performance analysis). By the manipulation of this representation, new knowledge about the modeled system can be obtained without the danger, cost, or inconvenience of manipulating the real system.

Despite the diversity of these systems, several features are common to most of them. Complex systems are usually composed of separate, interacting components. Each component may itself be a system, but its behavior can be described independently of other components of the system. Often the components of a system exhibit concurrency or parallelism; activities of one component may occur simultaneously with activities of other components. The concurrent nature of activities must be reflected in the modeling process. Since the components of the system interact, it is necessary to synchronize some activities: the transfer of information or materials from one component to another requires that the activities of the involved components be coordinated while the interaction takes place. This may result in one component waiting for another component. The timing of actions of different components may be quite complex, and the resulting interactions difficult to describe.

Petri nets have been proposed specifically to model systems with interacting concurrent components. Petri nets are graph-like models with two types of nodes, places which correspond to conditions (in a very general sense) and transitions representing events; such place/transition nets are also called condition/event systems. Directed arcs connect places with transitions and transitions with places, representing the causality relation between conditions and events. The dynamic behavior is represented by tokens which are assigned to places of a net. If all conditions of an event are satisfied, i.e., if all places connected with a transition contain tokens, an event can occur, and the occurrence of an event removes a single token from each input place of a transition and a single token is deposited in each output place of this transition, creating a new distribution of tokens and a new set of events (or transitions) that can occur. The behavior of such a model can be represented by a collection of all possible distributions of tokens that can be derived in a net.

Concurrent activities are represented in Petri nets by several possible occurrences of events. Synchronization of concurrent activities is obtained by fusion of events (i.e. transitions) which are supposed to occur simultaneously. Consequently, representation of concurrency and synchronization is "natural" in Petri nets, and in recent years different classes of net models have been gaining popularity in modeling and analysis of complex systems.

For the study of system performance the concept of time had to be introduced in the definition of Petri nets and the resulting net is called a timed Petri net. For condition/event systems, time is naturally associated with events. The behavior of systems can be represented either by a detailed deterministic description of system's evolution or probabilistically, by capturing the essence of the system's behavior and avoiding the details which add to the complexity of the model. Thus, time in a Petri net model can be specified in either a deterministic or a stochastic manner.

A system's evolution in time may be described by the chronological sequence of states the system operates in. Thus, the behavior of the system's model is also described by all possible states of the net. A state provides information on token distribution and on the status of events. Reachability analysis is one of the methods which provide detailed information about each state. From this information, performance indices of interest can be obtained very easily. However, the complexity of the reachability method increases very quickly with the size of the system, making it impractical for large systems.

This thesis proposes a less costly method which provides the same detailed information as a reachability analysis. This method uses composition/decomposition of the net state space (or state graph). Thus, complex state graphs, corresponding to the existence of a large number of tokens in the net are being composed from simpler state graphs corresponding to fewer tokens in the net. State information and performance indices for smaller nets are then used for performance analysis of the original net.

The thesis is organized as follows: Chapter 2 presents an introduction to Petri nets and their analysis methods. A survey of current research on analysis methods is included here as well. Chapter 3 formally defines the composition of state graphs and of the performance indices. Chapter 4 presents the application of the method on two real systems for which performance measures are obtained. The numerical results obtained by decomposition of state graphs are validated by the reachability analysis method. Conclusions and discussions on future work are addressed in Chapter 5.

Chapter 2

PETRI NETS

Sections 2.1 and 2.2 recall the fundamental concepts and properties of Petri nets. Section 2.3 introduces Petri nets with time and outlines their analysis. The chapter concludes with an overview of the methods for analyzing timed Petri nets.

2.1 Basic concepts of Petri nets

Several books discussing various aspects of Petri net theory were published recently ([26], [27], [8], [5] and [22]). This section recalls basic concepts of Petri nets which are needed for subsequent chapters; it does not intend to present an exhaustive introduction in Petri net theory.

There are many slightly different ways in which Petri nets can be defined. The approach used in this thesis follows [26], [25], [27] and [8].

Definition 1 ([27]) *A Petri Net is a triple* $\mathcal{N} = (P, T, A)$ *where:*

- P *is a finite nonempty set of places,*
- T *is a finite nonempty set of transitions,*
- A *is a set of directed arcs,* $A \subseteq P \times T \;\cup\; T \times P$*, such that*

$$\forall\, t \in T:\ \exists\, p_i, p_j \in P:\ (p_i, t) \in A\ \wedge\ (t, p_j) \in A.$$

The arcs can have weights, in which case a weight function $w : A \rightarrow \{1, 2, 3, \cdots\}$ is added to the definition of a net. By default the arc weights are equal to 1, and the net with default weights is called *ordinary* or *standard*. Only ordinary nets are considered in this thesis.

Graphically, a Petri net is represented as a directed graph with two types of nodes (i. e., a bipartite graph): places represented by circles and transitions represented by bars, with directed arcs connecting places with transitions and transitions with places. The dynamic behavior of a net is represented by tokens, which are distributed over places. This distribution can change if some conditions are satisfied.

Definition 2 ([8]) *A marking of a net* (P, T, A) *is a mapping which assigns a non-negative number of tokens to each place in* P, $m : P \rightarrow \mathbf{N}$. *A marking is often represented by a vector* $[m(p_1)...m(p_n)]$, *using an (arbitrary) ordering of the set of places* P.

Definition 3 ([8]) *A marked net* $\mathcal{M}$ *is a pair* $\mathcal{M} = (\mathcal{N}, m_0)$ *where:*

- $\mathcal{N}$ *is a net* $\mathcal{N} = (P, T, A)$, *and*
- m_0 *is a marking of* $\mathcal{N}$, $m_0 : P \rightarrow \mathrm{N}$, *called the initial marking.*

Example: Fig. 2.1 represents a Petri net which models the behavior of a simple interactive system. Place p_1 models the queue of waiting jobs. A token in place p_3 means that the server is available. Transitions t_1 and t_2 and place p_2 model the server servicing a job in two stages. Once a job is serviced, the server is freed. Transition t_3 and place p_4 model the thinking time after which a new job joins the waiting queue (place p_1). The initial marking is $m_0 = [3, 0, 1, 0]$ which means that the system has three terminals and one server. □

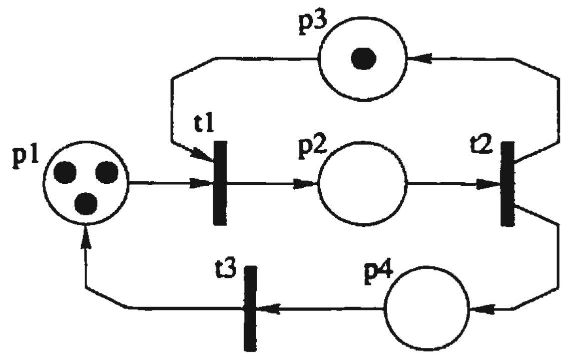

Figure 2.1: A model of an interactive system.

The places which are connected to a transition are called *input places* of this transition and places connected by arcs outgoing from a transition are called *output places* for that transition.

The set of input places of transition t is defined as $Inp(t) = \{p \mid (p,t) \in A\}$, and the set of output places of t as $Out(t) = \{p \mid (t,p) \in A\}$. Similarly, sets $Inp(p) = \{t \mid (t,p) \in A\}$ and $Out(p) = \{t \mid (p,t) \in A\}$, denote the input and output sets of transitions of the place p.

In an ordinary net, a transition t is enabled by a marking m if all its input places contain at least one token. If t is enabled then it can fire and the firing of t will create the successor marking m' (written $m \xrightarrow{t} m'$), defined by:

$$\forall\, p \in P:\; m'(p) = \begin{cases} m(p) - 1, & \text{if } p \in Inp(t) \setminus Out(t), \\ m(p) + 1, & \text{if } p \in Out(t) \setminus Inp(t), \\ m(p), & \text{otherwise.} \end{cases}$$

A marking m which enables no transition is called *dead.*

In our example, the initial marking $m_0 = [3, 0, 1, 0]$, enables transition t_1 and after its firing marking $m_1 = [2, 1, 0, 0]$ is created. This new marking enables transition t_2, and its firing creates marking $m_2 = [2, 0, 1, 1]$. Marking m_2 is called reachable from m_0.

A marking m' is *immediately reachable* in a marked net $(\mathcal{N}, m_0)$ from marking m if it can be obtained by firing a transition enabled by m.

Let m be a marking in $\mathcal{N}$. If $m \xrightarrow{t_1} m_1 \xrightarrow{t_2} \ldots \xrightarrow{t_n} m_n$, then $\sigma = t_1 t_2 \ldots t_n$ is called a firing sequence leading from m to m_n and is written as $m \xrightarrow{\sigma} m_n$. This notion includes the empty sequence ϵ, so $m \xrightarrow{\epsilon} m$ for every marking m. Marking m' is (generally) reachable from marking m, $m \xrightarrow{*} m'$, if there exists a firing sequence σ such that $m \xrightarrow{\sigma} m'$. The set of all markings reachable from m in a net $\mathcal{N}$ is called the set of *reachable markings* and denoted $R(\mathcal{N}, m)$.

Definition 4 *A place $p \in P$ is* shared *iff it belongs to the input set of more than one transition:*

p is shared $\Leftrightarrow\ \exists\, t_i, t_j \in T:\ t_i \neq t_j\ \wedge\ p \in Inp(t_i) \cap Inp(t_j)$.

Definition 5 *A shared place p is* free-choice *iff the input sets of all transitions sharing p are identical:*

p is free-choice $\Leftrightarrow \forall\, t_i, t_j \in T : t_i \neq t_j \wedge p \in Inp(t_i) \cap Inp(t_j)\ \Rightarrow\ Inp(t_i) = Inp(t_j)$.

Definition 6 *A shared place which is not free-choice is a* conflict *place.*

It can be observed that for each marking m, either all transitions sharing a free-choice place are enabled or all are disabled. This is not true for transitions sharing a conflict place, as shown in Fig. 2.2 where p_2 and p_3 are conflict places, and t_1 can be enabled while t_2 is disabled, or t_2 can be enabled while t_1 is disabled.

Let $E(m)$ denote the set of all transitions enabled by m. Enabled transitions t_i and t_j are in conflict at marking m iff:

1) $Inp(t_i) \cap Inp(t_j) \neq \emptyset$, or

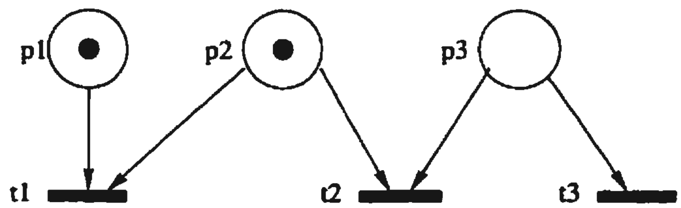

Figure 2.2: An example of a conflict.

2) $\exists\, t_k \in E(m): \; Inp(t_i) \cap Inp(t_k) \neq \emptyset \;\wedge\; t_k$ is in conflict with t_j.

The relation of being in conflict at a marking m is reflexive, symmetric and transitive, so it is an equivalence relation in the set of transitions T; it implies a partition of this set into a collection of disjoint conflict classes:

$$Conf(t,m) = \{T_1, T_2, \ldots, T_k\}. \tag{2.1}$$

It should be observed that classes of transitions sharing free-choice places are conflict classes.

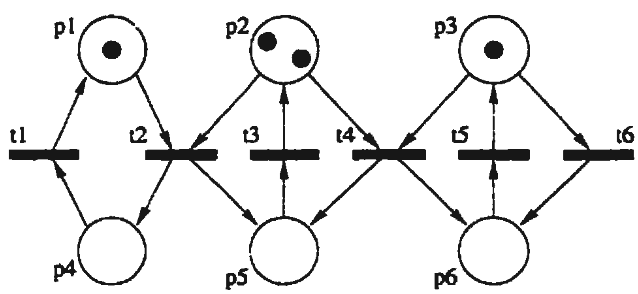

Figure 2.3: Different classes of conflicts.

In Fig. 2.3 transitions t_2 and t_4 are in conflict, and the partition of T is:

$$\{\{t_1\}, \{t_2, t_4, t_6\}, \{t_3\}, \{t_5\}\}$$

For any marking m a selection function $e : T \rightarrow \mathbf{N}$ is any function which determines all those firings which can occur simultaneously ([36]), i.e., any function e for which:

1) there exists a sequence of transitions $w = (t_{i_1}, t_{i_2}, \ldots t_{i_k})$ such that $t_{i_j} \in E(m_{i_{j-1}}), j = 1, 2, \ldots, k$, where $m_{i_0} = m$ and
$$\forall\, p \in P: \; m_{i_j}(p) = m_{i_{j-1}}(p) - \begin{cases} 1, & \text{if } p \in Inp(t_{i_j}); \\ 0, & \text{otherwise.} \end{cases}$$

2) $E(m_{i_k}) = \emptyset$, and

3) $\forall\, t \in T: \; e(t) = count(w, t)$, where $count(w, t)$ is the number of occurrences of transition t in the sequence w.

Example: For the previous example, the two selection functions are:
$$e_1 = [0, 1, 0, 1, 0, 0] \text{ and } e_2 = [0, 1, 0, 0, 0, 1].$$
For e_1, the sequence w is (t_2, t_1) or (t_1, t_2). □

The set of all selection functions for a marking m is denoted $Sel(m)$.

The structure of the net can also be represented by the so called *incidence* (or *connectivity*) matrix of a net.

Definition 7 ([8]) *Let $\mathcal{N} = (P, T, A)$ be a net. The incidence matrix $C : (P \times T) \rightarrow \{-1, 0, 1\}$ is defined as:*

$$\forall\, p \in P : \forall\, t \in T: \; C(p,t) = \begin{cases} -1, & \text{if } (p,t) \in A \;\wedge\; (t,p) \notin A; \\ 1, & \text{if } (p,t) \notin A \;\wedge\; (t,p) \in A; \\ 0, & \text{otherwise.} \end{cases}$$

The matrix representation is unambiguous only for *pure* nets, i.e., nets without self-loops. Fig. 2.4 is an example of a self-loop where the flow of tokens from place p to place p is not represented in the incidence matrix.

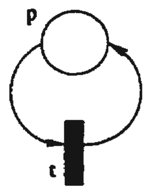

Figure 2.4: An example of a self-loop.

For the net in Fig. 2.1, the incidence matrix is:

$$C = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

Lemma 1 (Marking Equation Lemma [8]) *For every finite firing sequence w and any markings m and m' of a net $\mathcal{N}$ such that $m \xrightarrow{w} m'$, the following marking equation holds:*

$$m' = m + c \times e \tag{2.2}$$

where markings m and m' are represented by column vectors and, for all $t \in T$, $e(t) = count(w, t)$.

For the previous example (Fig. 2.1), let $w = t_1 t_2$ and $e = [1, 1, 0]$. Using the Marking Equation Lemma, we have:

$$m' = \begin{bmatrix} 3 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

which yields marking $m' = [2, 0, 1, 1]$, the marking obtained after firing transitions t_1 and t_2.

2.1.1 Properties of Petri nets

An important advantage of using Petri nets is their support for analysis of many properties of concurrent systems. There are two types of properties that can be studied with a Petri net model: *behavioral* properties, which depend on the initial marking of the net, and *structural properties*, which are independent of the initial marking.

Behavioral properties

The most popular behavioral properties include: *liveness*, *boundedness* and *reachability*.

Definition 8 ([8]) *A marked net is* live *if, for every reachable marking m and every transition t, there exists a marking m' reachable from m which enables t. If $\mathcal{M} = (\mathcal{N}, m_0)$ is a live net, then also m_0 is called a live marking of $\mathcal{N}$.*

Definition 9 ([26]) *A marking m of net $\mathcal{M} = (\mathcal{N}, m_0)$ is live iff*

$$\forall\, t \in T : \exists\, m' \in R(\mathcal{N}, m) : \; t \in E(m').$$

Proposition 1 ([26]) *A marked net $\mathcal{M}$ is live if and only if all markings $m \in R(\mathcal{N}, m_0)$ are live.*

Liveness of a system is closely related to the absence of dead markings in the set of reachable markings. If a net is live, then from any reachable marking m, any transition in the net can be fired either directly or through some firing sequences. If a net is live, then the system modeled by the net can always continue to operate.

In many cases, the number of tokens which can be associated with a place at any time is finite. Such a place is called *bounded.*

Definition 10 ([8]) *A marked net is* bounded *if for every place p there is a natural number k such that $m(p) \leq k$ for every reachable marking m. If a net $(\mathcal{N}, m_0)$ is bounded, m_0 is also called a bounded marking of $\mathcal{N}$.*

The bound of a place p in a bounded net $(\mathcal{N}, m_0)$ is defined as:

$$\max\{m(p) \mid m \in R(\mathcal{N}, m_0)\}. \tag{2.3}$$

A net with places whose bound is not greater than k is called *k-bounded.*

Proposition 2 ([8]) *Every bounded net is k-bounded for some $k \in \mathbf{N}$. Every bounded net has a finite set of reachable markings.*

Definition 11 *A Petri net is* safe *if it is 1-bounded.*

The net in Fig. 2.1 is live but not safe; it is 3-bounded.

Structural properties

Structural properties depend only on the structure of the net and hold for any initial marking. They can often be characterized by the incidence matrix.

More detailed information on structural properties can be found in [25], [26] and [22]. Some structural properties used in this thesis are presented below.

Definition 12 (structural liveness [25]) *A Petri net $\mathcal{N}$ is* structurally live *if there exists a live initial marking for $\mathcal{N}$.*

Definition 13 (structural boundedness [25]) *A Petri net $\mathcal{N}$ is* structurally bounded *if it is bounded for any finite initial marking m_0.*

Definition 14 (conservativeness [25]) *A Petri net $\mathcal{N}$ is partially* conservative *if there exists a place p and a positive integer $y(p)$ such that for every $m \in R(\mathcal{N}, m_0)$ and for any fixed initial marking m_0, the weighted sum of tokens:*

$$m^T \times y = m_0^T \times y = const$$

where m^T denotes the transpose of m, and m and y are represented by vectors.

Net invariants

The set of reachable markings set for the net shown in Fig. 2.1 is as follows:

$R(\mathcal{N}, m_0) = \{[3, 0, 1, 0], [2, 1, 0, 0], [2, 0, 1, 1], [1, 1, 0, 1], [1, 0, 1, 2], [0, 1, 0, 2], [0, 0, 1, 3]\}$.

It can be observed that the total number of tokens in places p_1, p_2 and p_4 is always 3 for any reachable marking and the total number of tokens in places p_2 and p_3 is always 1. These sets of places are *net invariants* called *place invariants* or P-invariants.

Also, the marking obtained by firing the sequence $t_1t_2t_3$ is the initial marking. The set of transition in this firing sequence represents another net invariant, called *transition invariant*, or T-invariant, which indicates how many times, starting from one marking, each transition has to fire to reproduce that marking.

The following definitions, lemmas and theorems considering properties of P- and T-invariants are taken mostly from [26]. [8] presents some alternative definitions of place and transition invariants. [23] gives a simple and quick algorithm for finding out these invariants for a given net. More detailed descriptions of net invariants can be found in [8], [16], [5] and [22].

In the following definitions $\mathbf{Z}$ denotes the set on non-negative integers.

Definition 15 ([26]) *Let $\mathcal{N} = (P, T, A)$. A non-zero card(P)-element vector s, $s : P \to \mathbf{Z}$, is called P-invariant iff $C^T \times s = 0$, where C is the incidence matrix of $\mathcal{N}$.*

It can be observed that any linear combination of P-invariants is also a P-invariant.

The following is an alternative of the *conservativeness* property:

Theorem 1 ([26]) *Let $\mathcal{N} = (P, T, A)$ be a conservative net. Then for each P-invariant s of $\mathcal{N}$ and for each marking m reachable from the initial marking m_0, the following holds:*

$$s \times m_0 = s \times m. \tag{2.4}$$

The converse of this theorem is true only if every transition fires at least once [26].

Definition 16 ([26]) *A net $\mathcal{N} = (P, T, A)$ is covered by place invariants iff for each place $p \in P$ there exists a P-invariant s of $\mathcal{N}$ such that $s(p) > 0$.*

If a net is covered by P-invariants then there exists an invariant s such that $s(p) > 0$ for all places $p \in P$.

Fig. 2.5 represents the net of Fig. 2.1 with its place invariants indicated by broken lines. The place invariants are: $s_1 = [1, 1, 0, 1]$ and $s_2 = [0, 1, 1, 0]$. The invariant $s = s_1 + s_2 = [1, 2, 1, 1]$ has all its components positive, therefore the net is covered by P-invariants, its reachability set is finite, and the net is bounded.

Definition 17 ([26]) *Let $\mathcal{N} = (P, T, A)$. A non-zero card(T)-element vector $v : T \to \mathbf{Z}$ is called T-invariant iff $C \times v = 0$.*

As for P-invariants, any linear combination of T-invariants is also a T-invariant.

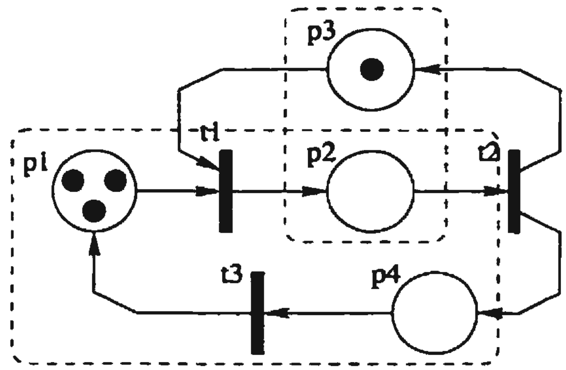

Figure 2.5: Place invariants.

Definition 18 ([26]) *A net $\mathcal{N}$ is covered by T-invariants iff for each transition $t \in T$ there exists a T-invariant v of N such that $v(t) > 0$.*

Proposition 3 ([26]) *If a net $\mathcal{N}$ is covered by T-invariants then there exists a T-invariant v of $\mathcal{N}$ such that $v(t) > 0$ for all $t \in T$.*

Theorem 2 ([26]) *Every net which is finite, live and bounded, is covered by T-invariants.*

The net in Fig. 2.1 has only one T-invariant consisting of all transitions in the net: $v = [1, 1, 1]$. The net is covered by a T-invariant, therefore it is live and bounded. The net in Fig. 2.6 is an example of a net covered by two T-invariants: $v_1 = [1, 0, 1, 0]$ and $v_2 = [0, 1, 0, 1]$. It is also covered by two P-invariants: $s_1 = [1, 0, 1, 1]$ and $s_2 = [0, 1, 0, 1]$. This net is also live and bounded.

The approach presented in [8] finds the net invariants without using the connectivity matrix.

Definition 19 ([8]; an alternative definition of P-invariants) *Let $\mathcal{N} = (P, T, A)$.*

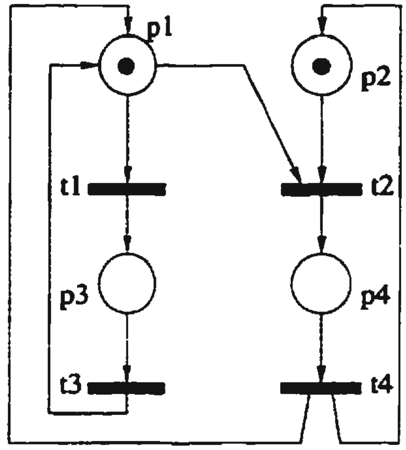

Figure 2.6: Net covered by 2 T-invariants and 2 P-invariants.

A mapping $i : P \to \mathbf{Z}$ *is a* P*-invariant iff for every transition* t*:*

$$\sum_{p \in Inp(t)} i(p) = \sum_{p \in Out(t)} i(p). \tag{2.5}$$

Definition 20 ([8]; an alternative definition of T-invariants) *Let* $\mathcal{N} = (P, T, A)$*. A mapping* $j : T \to \mathbf{Z}$ *is a* T*-invariant iff for every place:*

$$\sum_{t \in Inp(p)} j(t) = \sum_{t \in Out(p)} j(t). \tag{2.6}$$

2.2 Analysis methods

Petri net analysis methods can be grouped in two categories: *reachability* methods which can also be used in combination with *reduction* or *decomposition* techniques ([25, 5, 22]) and *structural* methods.

2.2.1 Reachability analysis

This method performs exhaustive analysis of all reachable markings for a given net. It can be applied to any bounded net, but its use is limited to cases when the number of reachable markings is not too large.

Given a marked net $(\mathcal{N}, m_0)$, the reachability set $R(\mathcal{N}, m_0)$ can be represented by a graph, starting from the initial marking m_0 and adding all markings which are immediately reachable from m_0, and so forth.

Definition 21 ([25]) *The* reachability graph *of a Petri net* $(\mathcal{N}, m_0)$ *is a labeled graph* $\mathcal{G} = (V, E)$. *The set of nodes* V *is the set of all reachable markings, and the set of arcs* E *represents the relation of immediate reachability, that is:*

$$(m_i, m_j) \in E \Leftrightarrow m_i \xrightarrow{t} m_j.$$

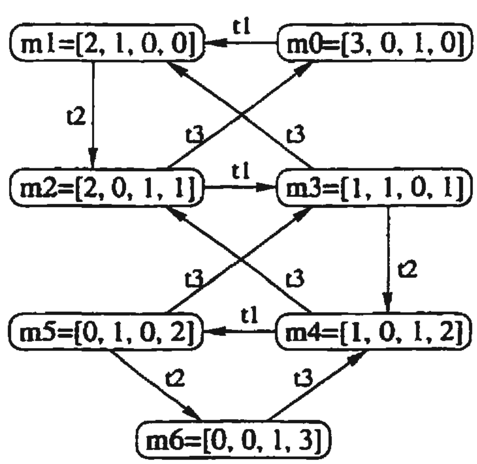

Figure 2.7: Reachability graph for the initial marking $m_0 = [3, 0, 1, 0]$.

Example: Fig. 2.7 shows the reachability graph for the net of Fig. 2.1 with the initial marking $m_0 = [3, 0, 1, 0]$. □

Reduction or decomposition techniques are applied together with the reachability analysis; if a large net can be reduced to a simpler net, the complexity of reachability analysis can be reduced. However, the reduction techniques must preserve the properties of the original net and, as a consequence, the applicability of this method is limited to special classes of nets.

The following reduction rules preserve the properties of liveness ([25]):

- *fusion of series places* Fig. 2.8 (a),
- *fusion of series transitions*, Fig. 2.8 (b),
- *fusion of parallel places*, Fig. 2.8 (c),
- *fusion of parallel transitions*, Fig. 2.8 (d).

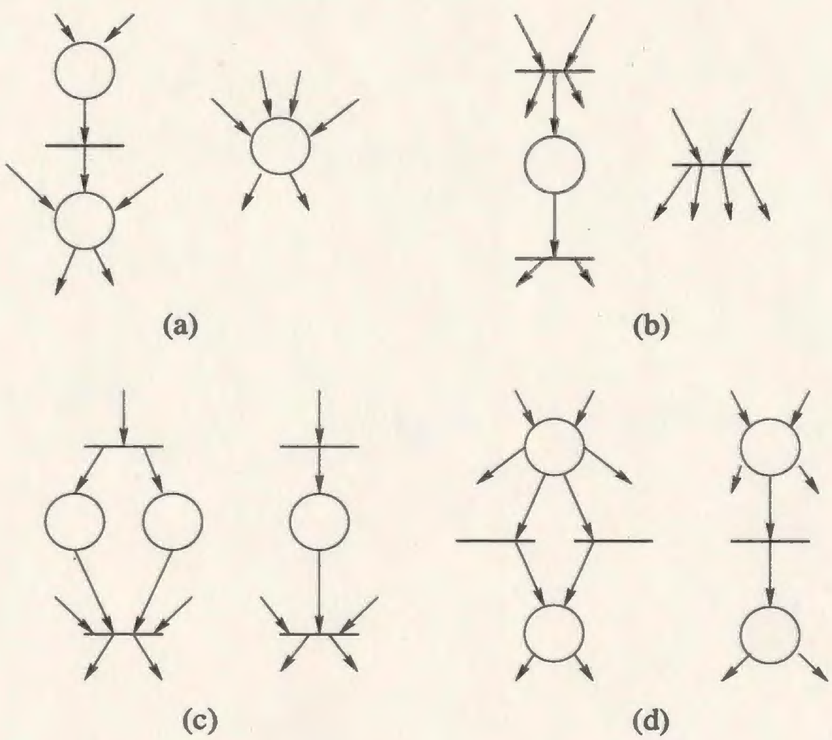

Figure 2.8: Some reduction rules.

2.2.2 Structural methods

Structural methods are used to investigate structural properties of the net.

The incidence matrix and marking equation have already been defined along with the equations for obtaining the place and transition invariants of a given net. Reachable marking can be determined by using the Marking Equation Lemma. Using the net invariants, structural properties can be derived, e.g.: if the net is covered by P-invariants then the net is structurally bounded. The upper bound on the number of tokens in a place can be determined due to the coverage of P-invariants. For example, in the net shown in Fig.2.1, places $\{p_1, p_2, p_4\}$ are 3-bound and place p_3 is 1-bound. Also, in the case when the net is covered by P-invariants, and at least one transition is not included in any T-invariants, then the net is not live.

Definition 22 ([22]) *A* deadlock *is such a subset of places $P_D \subset P$ that the set of its input transitions is a subset of the set of its output transitions, that is:*

$$\bigcup_{p \in P_D} Inp(p) \subset \bigcup_{p \in P_D} Out(p). \tag{2.7}$$

Definition 23 ([22]) *A* trap *is such a subset of places $P_T \subset P$ that the set of its output transitions is a subset of the set of its input transitions, that is:*

$$\bigcup_{p \in P_T} Out(p) \subset \bigcup_{p \in P_T} Inp(p). \tag{2.8}$$

Traps and deadlocks can be used to study properties of the nets: there is no way an empty deadlock can increase its number of tokens, and therefore all transitions which have input places in P_D cannot fire, so the net cannot be live. Also, it is known that a trap, if it is marked by the initial marking, will never loose its tokens; therefore a deadlock will not loose its tokens if it contains a marked trap.

More information on structural methods can be found in [25], [26], [8] and [22].

2.3 Petri Nets with time

Time can be introduced into Petri nets in several different ways. The characteristics associated with introducing time into a Petri net are as follows:

- the net elements (places or transitions) with which timing is associated,
- the rules applied for transition firing,
- the nature of the temporal specification (deterministic or probabilistic).

Time associated with places

In nets in which time is associated with places, a token that reaches a place becomes *available* only after a delay θ. Therefore, any token can be in one of the two states: *available* or *unavailable*, and only available tokens can enable a transition. An unavailable token models the time the system spends performing an activity. It is known [32] that Petri nets with time associated with places can be transformed into equivalent timed Petri net models in which time is associated with transitions. Only this other class of timed Petri nets is considered in this thesis.

Time associated with transitions

The association of time with transitions seemed to many authors as more natural [2]:

> "interpreting PN as state/event models, time is naturally associated with activities that induce state changes, and hence with the delays incurred before firing transitions".

There are, in the literature, two ways in which a timed transition fires. The firing is either instantaneous or not. In the former case, once a transition is enabled, the tokens spend the time associated with this transition in its input places, and after this time has elapsed, the transition fires removing the tokens from its input places and depositing them in its output places in one instantaneous operation. In the latter case, the transition initiates its firing absorbing the tokens from input places, and after the time associated with the transition has elapsed, the tokens are deposited into the output places of the transition. In [20], the Petri nets in which time is associated with transitions are called Timed Transition Petri Nets (TTPN).

The behavior of systems can be represented either by a detailed deterministic behavior of the system or probabilistically, by capturing the essence of the system's behavior and avoiding the details which add to the complexity of the model. Thus, time in a Petri net model can be specified in either a deterministic or stochastic manner. For the latter, random durations are characterized by a probabilistic distribution function, and the most popular of such functions is the (negative) exponential distribution.

A TTPN in which time is specified in a stochastic manner and the firing of a transition is an instantaneous event is called a stochastic Petri net ([21, 19, 20, 2, 3] and others). On the other hand, a TTPN in which the firing of a transition is not instantaneous and time is specified in either a deterministic or stochastic manner is called a timed Petri net ([34], [35], [14], [36] and [37]).

2.3.1 Stochastic Petri nets

Stochastic Petri nets were introduced as an extension of Petri nets for performance evaluation of modeled systems. This extension introduces exponentially distributed holding times (or sojourn times) associated with the markings. These holding times are obtained by assigning 'firing rates' to transitions of the net. The holding time of a marking is determined by the transitions enabled by it.

By taking a standard Petri net and associating with each of its transitions an exponentially distributed random variable which models the time between the enabling and the firing of a transition, a Stochastic Petri Net (SPN) is obtained.

Definition 24 ([21]) *A stochastic Petri net is a 5-tuple:* $\mathcal{S} = (P, T, A, m_0, r)$ *where:*

- P, T, A, *and* m_0 *are as for standard PN;*
- $r : T \to \mathbf{R}^+$ *is a firing time function which assigns the rate of firings to each transition of the net. Firing rates can be marking dependent, and then* r *is defined as* $r : T \times R(\mathcal{N}, m) \to \mathbf{R}^+$.

Consider an SPN with a marking m which enables several transitions. It is known that a random variable which is equal to the smaller of two exponentially distributed random variables with parameters r_1 and r_2 is an exponentially distributed random variable, and its rate is equal to $(r_1 + r_2)$. Therefore, the sojourn time associated with marking m is

$$\frac{1}{\sum_{t_i \in E(m)} r(t_i)} \tag{2.9}$$

where $E(m)$ is the set of transitions enabled by marking m, and r is the firing rate function.

If m' is the marking obtained by firing transition t, some of the transitions enabled by m' might still be enabled. Because the exponential distribution yields a residual time distribution equal to the distribution of the firing delay itself (the memoryless property), the activity associated with each transition can be considered as restarted in a new marking.

The reachability graph of the basic SPN model is isomorphic with the reachability graph of the underlying standard Petri Net. Therefore, all the results of structural analysis of standard Petri nets can be applied to stochastic nets. Molloy [24] has shown that SPNs are isomorphic to continuous–time Markov chains.

The continuous–time Markov chain corresponding to an SPN can be determined as follows [21]:

1. the state space S of the Markov chain is the reachability set $R(\mathcal{N}, m_0)$ of the standard Petri net modeling the stochastic net;

2. the rate q_{ij} of transitions from state i (marking m_i) to state j (marking m_j) is equal to:

$$q_{ij} = \sum_{t \in E_j(m_i)} r(t), \tag{2.10}$$

   where $E_j(m_i)$ is the set of transitions enabled by marking m_i whose firings generate marking m_j.

The steady–state (stationary or equilibrium) probabilities $\pi(i)$ for each state i are obtained by solving the system of linear equations:

$$\begin{cases} \pi \times Q = 0 \\ \sum \pi = 1 \end{cases} \tag{2.11}$$

where Q is the transition probability matrix with elements q_{ij}, as defined above.

2.3.2 Timed Nets

In timed nets the firing times can be either deterministic or stochastic. The class of nets with deterministic firing times is called D-timed Petri nets. For stochastic firing times, the only case considered in the literature is when the firing times are exponentially distributed random variables ([36]), due to the memoryless property of this distribution which simplifies the state description of the corresponding nets. This class of nets is called M-timed Petri nets.

Zuberek ([34], [35], [36]) proposed solutions for several classes of timed nets: conflict-free, free-choice, inhibitor nets, extended (with interrupt arcs) nets and enhanced Petri nets. Inhibitor M-timed Petri nets provide a mechanism to model priorities. Extended M-timed Petri nets use interrupt arcs to discontinue the firing of transitions. Enhanced M-timed Petri nets have two classes of transitions: immediate and timed. One can see them as being a combination of an ordinary (without time) free-choice Petri nets and timed extended free-choice bounded M-timed Petri nets. The modeling power is increased in this way.

More detailed information on timed Petri nets, their analysis and applicability to modeling different systems can be found in [34], [35], [14], [36], [37]. A collection of software tools for analysis of timed Petri nets, called TPN-tools, is under development ([39] and [38]).

M-timed Petri nets

Definition 25 ([36]) *An M-timed net $\mathcal{T}$ is a quadruple $\mathcal{T} = (\mathcal{N}, m_0, c, r)$ where:*

- *$\mathcal{N}$ is a standard net, $\mathcal{N} = (P, T, A)$, and m_0 is the initial marking, $m_0 : P \to \mathbf{N}$;*
- *c is a conflict–resolution function which assigns a positive relative frequency of firings to each transition $t \in T$, $c : T \to \mathbf{R}^+$;*
- *r is a firing–rate function which assigns a rate of exponentially distributed firing times to each transition $t \in T$, $r : T \to \mathbf{R}^+$.*

The behavior of timed Petri nets is described by states and state transitions. Each state describes the distribution of tokens in places and also the firings of transitions.

Definition 26 ([36]) *A state s of a timed net $\mathcal{T} = (\mathcal{N}, m_0, c, r)$, $\mathcal{N} = (P, T, A)$, is a pair of functions, $s = (m, f)$, $m : P \to \mathbf{N}$, $f : T \to \mathbf{N}$, where m describes the marking of a net (in state s), and f describes the number of (active) firings of transitions.*

Definition 27 ([36]) *An initial state of a net $\mathcal{T} = (\mathcal{N}, m_0, c, r)$ is any state $s_i = (m_i, f_i)$ such that:*

1) $f_i \in Sel(m_0)$, and

2) $\forall\ p \in P:\ m_i(p) = m_0 - \sum_{t \in Out(p)} f_i(t)$.

Example: The net in Fig. 2.6 is converted into an M-timed net by associating firing rates with all transitions. For the initial marking $m_0 = [1, 1, 0, 0]$, the two possible initial states are, $s_1 = [0, 0, 0, 0; 0, 1, 0, 0]$ and $s_2 = [0, 1, 0, 0; 1, 0, 0, 0]$. □

Definition 28 ([36]) *A state $s_j = (m_j, f_j)$ is directly (t_k, e_l)-reachable from a state $s_i = (m_i, f_i)$, $s_i \xrightarrow{(t_k, e_l)} s_j$, iff:*

1) $f_i(t_k) > 0$;

2) $e_l \in Sel(m_{ij})$;

3) $\forall\ p \in P:\ m_j(p) = m_{ij}(p) - \sum_{t \in Out(p)} e_l(t)$;

4) $\forall\ t \in T:\ f_j(t) = f_{ij} - e_l(t)$,

where:

5) $\forall\ p \in P:\ m_{ij}(p) = m_i(p) + \begin{cases} 1, & \text{if } t_k \in Inp(p); \\ 0, & \text{otherwise;} \end{cases}$

6) $\forall\ t \in T:\ f_{ij}(t) = f_i(t) - \begin{cases} 1, & \text{if } t = t_k; \\ 0, & \text{otherwise.} \end{cases}$

For each state $s = (m, f)$, the holding time (or sojourn time) is:

$$h(s) = \frac{1}{\sum_{t \in T} f(t) * r(t)} \tag{2.12}$$

If a state $s_j = (m_j, f_j)$ is directly (t_k, e_l) reachable from a state $s_i = (m_i, f_i)$, then the rate of transitions from s_i to s_j is:

$$q(s_i, s_j) = \frac{a(t_k) * b(e_l)}{h(s_i)} \tag{2.13}$$

where the probability that transition t_k will terminate is:

$$a(t_k) = \frac{r(t_k)}{\sum_{t \in T} f_i(t) * r(t)}, \tag{2.14}$$

and

$$b(e_l) = \sum_{1}^{n} \prod_{n=1}^{len(w)} d(m_{i_n}, t_{i_n}) \tag{2.15}$$

where n represents all firing sequences w such that $count(w) = e_l$, $len(w)$ is the length of w, e_l is the selection function used, $d(m, t)$ is the probability that transition t will be selected for firing and *Conf(m, t)* denotes the conflict class of t at marking m:

$$d(m, t) = \frac{c(t)}{\sum_{t' \in Conf(m,t)} c(t')}. \tag{2.16}$$

Definition 29 ([34]) *A state graph $\mathcal{G}$ of an M-timed Petri net $\mathcal{T}$ is a labeled directed graph $\mathcal{G}(\mathcal{T}) = (S, E, h, q)$ where:*

- *S is a set of vertices which is equal to the set of reachable states of the net $\mathcal{T}$,*
- *E is a set of directed arcs, $E \subseteq S \times S$, such that $(s_i, s_j) \in E$ iff s_j is directly reachable from s_i in $\mathcal{T}$,*
- *h is the holding time function which assigns the sojourn time of a state s to*

each $s \in S$, $h : S \to \mathbf{R}^+$, where:

$$h(s) = \frac{1}{\sum_{t \in T} f(t) * r(t)}, \tag{2.17}$$

- *q is a transition–rate function which assigns the rate of transitions from state s_i to state s_j to each arc $(s_i, s_j) \in E$, $q : E \to \mathbf{R}^+$.*

The state graphs of bounded M-timed Petri nets are finite continuous–time homogeneous Markov chains ([34]). The steady-state probabilities can be obtained by solving the system of linear equations:

$$\begin{cases} \sum_{1 \le j \le K} q(s_j, s_i) * \pi(s_j)/h(s_j) = \pi(s_i)/h(s_i); \quad i = 1, \ldots, K-1, \\ \sum_{1 \le i \le K} \pi(s_i) = 1, \end{cases}$$

where K in the total number of states.

Example: Table 2.1 shows the states and the stationary probabilities of states for the net of Fig. 2.1 with the following firing rates of transitions: $r(t_1) = 10$, $r(t_2) = 10$, $r(t_3) = 20$. □

state	$s_i = (m_i, f_i)$	π_i
1	[2, 0, 0, 0; 1, 0, 0, 0]	0.11538
2	[2, 0, 0, 0; 0, 1, 0, 0]	0.21429
3	[1, 0, 0, 0; 1, 0, 1, 0]	0.23077
4	[1, 0, 0, 0; 0, 1, 1, 0]	0.19780
5	[0, 0, 0, 0; 0, 1, 1, 0]	0.13187
6	[0, 0, 0, 0; 0, 1, 2, 0]	0.06593
7	[0, 0, 1, 0; 0, 0, 3, 0]	0.04396

Table 2.1: Stationary probabilities for net of Fig. 2.1.

Different performance measures can be obtained from the stationary probabilities, e.g. the average number of jobs waiting in the queue or the utilization of a server.

The average number of jobs waiting in the queue can be obtained by interpreting the states. Place p_1 models the queue of waiting jobs. In states s_1 and s_2, there are two jobs in the queue, in states s_3 and s_4 there is one job in the queue, and in the remaining states the queue is empty. The average number of waiting jobs in the queue is thus:

$$n(p_2) = 2 * (\pi_1 + \pi_2) + (\pi_3 + \pi_4) = 1.08791$$

D-timed Petri nets

For D-timed Petri nets, the firing times of transitions are deterministic. so the memoryless property does not apply to such models. Therefore the description of such nets must include the history of the firings, i.e., a state description must contain the remaining firing time for each transition's firing. The state graph for a D-timed net is a discrete-state discrete-time semi-Markov process ([36]), and thus, the steady-state probabilities can be obtained in a similar manner as for M-timed Petri nets. A detailed description of the behavior of different classes of D-timed Petri nets is presented in [35] and [36].

Holliday and Vernon ([14]) proposed an extension of timed nets called a Generalized Timed Petri Net (GTPN), which imposes no restrictions on the net apart from the finite state space. GTPNs can also include geometric and deterministic transition firing times, as well as state-dependent firing times.

2.3.3 Other types of Petri nets

An SPN with immediate transitions is called a Generalized SPN and a TPN with immediate transitions is called an Enhanced TPN. In both cases immediate transitions have priority over the timed ones. The analysis of these nets is more complex due to the existence of two different kinds of states or markings: *vanishing* and *tangible*. The *vanishing* states are those in which at least one immediate transition is enabled, and therefore the sojourn times of such states are zero; the *tangible* states are those in which only timed transitions are enabled.

In the case of GSPNs, the underlying process is semi-Markovian, so the solution is based on extracting the embedded Markov chain ([3]). All vanishing states can be eliminated either during the generation of the state space or just before the solution of the steady-state equation.

2.3.4 Performance measures

As was shown previously, stationary probabilities of states can be used very conveniently to obtain the performance characteristics of the modeled system. This is true not only for timed Petri nets, but for any discrete-state model. The average number of waiting tokens in a place, the utilization of a transition or a place or the average waiting time of a token in place can be used for the performance evaluation of the modeled system. The formulae for performance indices are different for stochastic and timed nets.

For a stochastic Petri net (P, T, A, m_0, r) some of the performance indices are as follows:

- transition utilization $U(t_j)$ is given by the steady state probability that transition t_j is enabled:

$$\forall\, p \in Inp(t_j) :\ U(t_j) = Prob(t_j, \text{enabled}) = \sum_{m(p)>0} \pi(m) \tag{2.18}$$

- the stationary probability that there are at least k tokens in place $p \in P$:

$$Prob(m(p) \geq k) = \sum_{m(p) \geq k} \pi(m) \tag{2.19}$$

- the throughput $\Theta(t_i)$ of a transition $t_i \in T$ represents the average number of firings of t_i per unit of time:

$$\Theta(t_i) = U(t_i) * r(t_i) \tag{2.20}$$

where $m \in R(\mathcal{N}, m_0)$.

For a timed Petri net $\mathcal{T} = (\mathcal{N}, m_0, c, r)$, $\mathcal{N} = (P, T, A)$, the same indices are defined slightly differently due to the different firing rules:

- the stationary probability that a transition $t_i \in T$ is firing in a state $s = (m, f)$ is given by:

$$\forall\ (m, f) \in S(\mathcal{T}):\ Prob(f(t_i) > 0) = \sum_{f(t_i) > 0} \pi(m, f) \tag{2.21}$$

- the utilization of transition $t_i \in T$ is defined by:

$$\forall\ (m, f) \in S(\mathcal{T}):\ U(t_i) = \sum_{f(t_i)=1}^{n} f(t_i) * \pi(m, f) \tag{2.22}$$

where n is the upper bound on the number of firings of transition t_i.

- the throughput of transition $t_i \in T$ is given by:

$$\Theta(t_i) = U(t_i) * r(t_i). \tag{2.23}$$

Example: For the M-timed Petri net in Fig. 2.1, the following performance measures are an example of indices that can be calculated from stationary probabilities π ([31], [34]):

the utilization of the server in the first stage:

$$u_1 = \pi_1 + \pi_3 + \pi_5 = 0.47802$$

the utilization of the server in the second stage :

$$u_2 = \pi_2 + \pi_4 + \pi_6 = 0.47802$$

the average number of jobs serviced per time unit:

$$\theta(t_1) = \theta(t_2) = u_1 * \frac{1}{r(t_1)} = u_2 * \frac{1}{r(t_2)} = 0.047802$$

the average number of waiting jobs in the queue:

$$n(p_1) = 2 * (\pi_1 + \pi_2) + \pi_3 + \pi_4 = 1.08791 \ \Box$$

2.4 Approaches to analysis of Petri net models

The basic approaches to analysis of timed Petri nets include: reachability analysis and structural analysis. Since reachability analysis performs an exhaustive analysis of the state space, its time and space requirements can increase very quickly with the size of the model (number of places and number of tokens in the system). Structural reduction methods can be used to reduce the size of the net (the number of places and transitions) but the application of such methods is limited to special classes of nets.

Structural methods determine various properties of the model without generating the state space.

Simulation is also widely used, and many simulation tools are available: TPN-tools ([39], [38]), GreatSPN, ALPHA/Sim, etc. ([1]). However, one of the drawbacks of simulation methods is that the states which have very small stationary probabilities might never occur even in very long simulation runs, and sometimes such rarely occurring states are the most interesting ones, e.g. the probability of transmission error over a fiber optic link, or a deadlock in an operating system.

In [6], Berthomieu and Menasche reduced the size of the state space by aggregating states into a state classes. A *state class* contains all those states which are reachable by a firing sequence sequence rather than a single transition. All such classes are enumerated and relations between classes are determined. Reachability analysis is then applied to the graph of state classes rather than states.

In recent years, work has been done on applying results from queueing theory to Petri nets ([18], [21], [11], and [5]). A special class of stochastic Petri nets has been defined, which has the property of having product form solution (PFS). The term "product form solution" is understood less strictly than for queueing networks. The steady-state solution for PFS nets has a product form composed of a normalization constant and as many terms as the number of places in the net. Stochastic Petri nets which have this property are called Product Form Stochastic Petri Nets (PF SPN). The other queueing theory technique applied for SPN is the Mean Value Analysis (MVA) ([15], [31], [28], and [29]), which reduces the complexity of computing the normalization constant for PFS. As is the case for queueing networks, if a net has a product form solution, then MVA can be used to obtain the performance measures of interest.

The first application of the queueing network Product Form Solution to stochastic Petri nets is given in [10] where the authors observed that the underlying Petri net is a bounded stochastic Petri net. The solution is similar to the product form for queueing networks using matrix and vectors instead of scalars. The underlying Petri nets are ordinary, bounded and strongly connected (live). The firing rates of transitions are assumed to be independent of the markings.

Live and 1-bounded (safe) stochastic Petri nets are considered in [17] where the necessary conditions are given for the equilibrium state probabilities to satisfy the local balance equations. It is known that for a queueing network with product form solution, the state diagram is decomposable into elementary building blocks. The approach presented in [17] is based on Theorem 2, which is equivalent to the state graph (reachability graph) being decomposable into building blocks. At a more careful inspection, Theorem 2 also gives the structural constraints which SPNs have to satisfy in order to have PFS. Recently, a particular class of stochastic Petri nets which display Product Form like Solutions has been investigated ([12], [13], [33],[30], [7], [4], [31], [28] and [29]). The generation of the reachability set for these nets is reduced to the analysis of the Markov chain associated with the transitions, and called *routing process*. For the routing process to be well defined, the input sets of any two transitions $t_i, t_j \in T$ must be different. For a positive solution of this process, the flow condition ([13]) given in various forms in [12], [33], [30] and others, has to be satisfied:

$$\forall\ t_i \in T : \exists\ t_j \in T :\ Inp(t_i) = Out(t_j).$$

In [13], the class of nets is extended and more flexible than in [6]. The authors considered alternating *enabling* and *firing time* points. At these points, a product form solution for the equilibrium probability that the net has a marking m is given. Tokens are absorbed by transitions just before firing points and deposited into output places just before enabling points. The change in state is due to the firing of only one transition, and no two transitions can have the same input sets.

By observing the model at different points, Henderson and Taylor ([13]) were able to derive product form like solutions for two different cases: when transition firings are instantaneous (this corresponds to Stochastic Petri Nets) and when there is zero time between an enabling point and the following firing point (this corresponds to timed Petri nets).

Henderson *et all* ([12]) proved that for a stochastic Petri net which satisfies the condition given above the equilibrium probabilities π are given by:

$$\pi(m) = K * \Phi(m) * g(m) \tag{2.24}$$

where :

K is a normalization constant;

$\Phi(.)$ is a given positive function ([12]);

$g : R(\mathcal{N}, m_0) \rightarrow \mathbf{R}$ is the solution of the routing process.

The two approaches presented in [17] and [12] and [13] are compared in [9]. In [17], the PFS is determined by inspecting the structure of the reachability graph, while [12] derives conditions for PFS to exist from the structure of the net. The equivalence between the two approaches is given by Lemma 6 in [9] which states that the state transition diagram is completely decomposable into building blocks (the approach used in [17]) if the SPN is covered by T-invariants, the sets of transitions corresponding to invariants satisfy the flow condition, and the initial marking enables at least one transition (these conditions are also given in [12]).

In [33], Ziegler and Szczerbicka extended the PFS to Generalized SPN's. The considered class of GSPN's has no inhibitor arcs and no marking dependent arc cardinalities. [33] presents an algorithm which checks if the GSPN can have a product form solution, and if it does, the solution is computed. First, structural transformations are performed in order to obtain a net which satisfies the structural constraints for a product form solution. These transformations eliminate all immediate transitions and remove identical input sets of transitions. If the net has some subnets which do not satisfy the structural conditions for a PF solution, they can be replaced by a block transition. The algorithm then performs all the necessary steps for computing the product form solution of the same form as in [12].

One of the problems is the computation of the normalization constant. [30] presents an algorithm to recursively compute this constant, and obtains the steady state probabilities for PF SPN's. The recursion is both on the number of tokens and the number of places in the net. The algorithm is polynomial in the number of places and the number of tokens in the initial marking. Sereno and Balbo prove that for the same conditions as in [12], the equilibrium distribution of the SPN is given by:

$$\pi(m) = \frac{1}{G} * h(m) \tag{2.25}$$

where $m \in R(\mathcal{N}, m_0)$, and $h(m)$ contains as many terms as there are places or transitions in the net, and under certain conditions, has the form:

$$h(m) = \prod_{p \in P} f_p^{m(p)} \tag{2.26}$$

where f_p is a function given in [30].

Coleman [7] proposed a new algorithm to calculate the normalization constant for PF SPN. This algorithm depends only on the structure of the net and not on the size of the reachability graph (as is the case of the algorithm presented by Sereno and Balbo [30]). Hence, the time complexity of Coleman's algorithm is much simpler.

Balbo *et al.* [4] analyze the same class of nets as in [30], and derive a formula to compute the mean sojourn time of a token in a place. The approach is similar to arrival theorems for queuing theory. The formula is recursive, depending on the average number of tokens in the same place, but for a smaller number of tokens in the initial marking. This is the basis for deriving and applying the Mean Value Analysis (MVA) algorithm to solve a PF SPN.

MVA or Approximate MVA are used for the derivation of performance measures in [31], [28], [29], without obtaining the PFS. The basis for MVA for SPN is the recursive computation of the normalization constant derived in [30]. For MVA, the nets have to satisfy the same conditions as for PFS. The advantage of Approximate MVA is that it can be applied to more general classes of nets.

In [31], the authors present the derivations of different performance indices like: utilization of a place, of a transition, the mean sojourn time of a token in place, the mean number of tokens in a given place. These performance indices are then used for deriving the performance measures of the modeled system. The formulae used are recursive on the number of tokens in the net and the place invariants are considered for the reduction of tokens.

An iterative Approximate MVA algorithm for stochastic nets which do not have product form solution is proposed in [28] and [29] for special classes of nets, and in particular, marked graphs. Marked graphs (MGs) are Petri nets in which each place has only one input transition and only one output transition. MGs seem quite simple but they can model synchronization and the fork–join constructs. It is well known that synchronizations are in general non-product features, and therefore MG are not PF SPNs. As in the case of queueing networks, one of the difficulties of Approximate MVA is the inability to provide bounds for the performance measures obtained this way. To quote Sereno [29]:

> "the quality of the results is stated by means of experimental evidence."

This thesis proposes a new way to derive performance measures for timed Petri nets. This is a hybrid method that uses both reachability and structural analysis. Independent and self-independent initial markings are identified and used for composition of state graphs. In this way, complex state graphs for nets with a large number of tokens can be composed from simpler state graphs of the same net. The markings which need to be analyzed exhaustively are quite simple, their reachability analysis is straightforward and cannot be affected by the "state explosion" problem. Consequently, the analysis of large state graphs is reduced by their decomposition, and the component state graphs provide results which are combined into the values of interest for the original system. Both, stationary probabilities and performance indices of interest can be obtained in this way.

Chapter 3

STATE SPACE DECOMPOSITION

An overview of the methods used to analyze different classes of Petri nets "with time" concludes that the reachability analysis becomes impractical for complex models due to time and space complexity. Therefore, research has been focused on methods which do not require the generation of the state space for obtaining the values of performance indices. However, only a few such methods have been proposed for timed Petri nets; stochastic nets have been studied and used more extensively.

This chapter explores the state graphs of M-timed Petri nets and proposes a method of decomposing state graphs with a large number of states into smaller state graphs. A composition of state graphs is then defined so that more complex state graphs can be constructed from simpler state graphs. These simpler graphs are generated for simple initial markings. Reachability analysis of these initial markings is quite straightforward and stationary probabilities of states and values of performance indices can be easily derived. These results are then composed into the stationary probabilities and performance measures of the original net.

The sections that follow use the concepts of state graph and labeled state graph. The state graph of a net $(\mathcal{N}, m, c, r)$ is denoted by $\mathcal{G}(\mathcal{N}, m, c, r) = (S, E)$ where S is the set of nodes and E the set of directed arcs. A labeled state graph is

$\mathcal{G}(\mathcal{N}, m, c, r) = (S, E, h, q)$ where h is a state labeling function, $h : S \rightarrow \mathbf{R}$, specifying the holding times of states, and q is an arc labeling function, $q : E \rightarrow \mathbf{R}$, describing the probabilities of transitions between states.

3.1 Composition of state graphs

The state graph of a timed net describes the behavior of the net. The composition of two state graphs is defined in such a way that it describes a change of state in only one of the composed graphs.

Definition 30 *Let $\mathcal{G}_1 = (S_1, E_1, h_1, q_1)$ and $\mathcal{G}_2 = (S_2, E_2, h_2, q_2)$ be two state graphs of the same timed net with two initial markings, m_1 and m_2.* The composition of state graphs *$\mathcal{G}_1$ and $\mathcal{G}_2$, $\mathcal{G}_1 \otimes \mathcal{G}_2$, is a labeled graph $\mathcal{G} = (S, E, h, q)$ where:*

- *S is the set of nodes:*

$$S = S_1 \otimes S_2 = \{(m_i + m_j, f_i + f_j) \mid (m_i, f_i) \in S_1 \wedge (m_j, f_j) \in S_2\} \tag{3.1}$$

- *E is a set of directed arcs, $E \subseteq S \times S$:*

$$\begin{aligned}(s_i, s_j) \in E \;\Leftrightarrow\; \exists\, s'_i, s'_j \in S_1 \;\wedge\; s''_i, s''_j \in S_2 : s_i = s'_i \otimes s''_i \;\wedge\; s_j = s'_j \otimes s''_j \;\wedge \\ ((s'_i, s'_j) \in E_1 \;\wedge\; s''_i = s''_j) \;\wedge\; (s'_i = s'_j \;\wedge\; (s''_i, s''_j) \in E_2)\end{aligned} \tag{3.2}$$

- *h is the holding time function labeling the nodes:*

$$\forall\, s = (m, f) \in S \;:\; \frac{1}{h(s)} = \sum_{f(t)>0} f(t) * r(t) = \frac{1}{h_1(s_i)} + \frac{1}{h_2(s_j)} \tag{3.3}$$

where $s = s_i \otimes s_j$,

- *q is the transition probability function labeling the arcs:*

$$\forall\ s_i, s_j \in S:\ q(s_i, s_j) = \begin{cases} q_1(s_i', s_j'), & \text{if } s_i = s_i' \otimes s_i'' \wedge s_j = s_j' \otimes s_j'' \wedge s_i'' = s_j''; \\ q_2(s_i'', s_j''), & \text{otherwise.} \end{cases} \tag{3.4}$$

Lemma 2 *Let $\mathcal{G} = (S, E, h, q) = \mathcal{G}_1 \otimes \mathcal{G}_2$, where $\mathcal{G}_1 = (S_1, E_1, h_1, q_1)$ and $\mathcal{G}_2 = (S_2, E_2, h_2, q_2)$ are state graphs of the same timed net with two initial markings, m_1 and m_2, respectively. Then the stationary probabilities of composed states are determined by the stationary probabilities of the components:*

$$\forall\ s \in S:\ \pi(s) = \sum_{s = s_i \otimes s_j} \pi_1(s_i) * \pi_2(s_j) \tag{3.5}$$

where $s_i \in S_1$, $s_j \in S_2$, π_1 and π_2 describe the stationary probabilities of states in S_1 and S_2, respectively.

Proof: It is straightforward to check that this solution satisfies the balance equations that describe the stationary probabilities of states. □

3.2 Independent Markings

Definition 31 *An initial marking m in a net $(\mathcal{N}, m, c, r)$ is* self-independent *if and only if the labeled state graphs $\mathcal{G}(\mathcal{N}, m+m, r)$ and $\mathcal{G}(\mathcal{N}, m, c, r) \otimes \mathcal{G}(\mathcal{N}, m, c, r)$ are isomorphic:*

$$\mathcal{G}(\mathcal{N}, m+m, c, r) \equiv \mathcal{G}(\mathcal{N}, m, c, r) \otimes \mathcal{G}(\mathcal{N}, m, c, r) \tag{3.6}$$

Definition 32 *The remainder marking set, $Rem(\mathcal{N}, m)$, is defined as:*

$$Rem(\mathcal{N}, m) = \{m_e' \mid e \in Sel(\mathcal{N}, m)\} \tag{3.7}$$

where:

$$\forall\, p \in P:\ m'_e(p) = m(p) - \sum_{t \in Out(p)} e(t). \tag{3.8}$$

Lemma 3 *If m is the initial marking in $(\mathcal{N}, c, r)$ then:*

$$\forall\, m' \in Rem(m):\ Sel(\mathcal{N}, m) = Sel(\mathcal{N}, m + m'). \tag{3.9}$$

Proof: From the definition of selection functions, $E(m') = \emptyset$ and for all $p \in P$, $m'(p) > 0 \Rightarrow m(p) > 0$, therefore $E(m + m') = E(m)$, and then $Sel(\mathcal{N}, m) = Sel(\mathcal{N}, m + m')$. □

Lemma 4 *Let m be a self-independent marking in a timed net in $(\mathcal{N}, m, c, r)$. Then:*

(a) *The selection functions can be obtained by composition:*

$$Sel(\mathcal{N}, 2 * m) = Sel(\mathcal{N}, m) \otimes Sel(\mathcal{N}, m). \tag{3.10}$$

(b) *The set S_0 of initial states of $(\mathcal{N}, 2 * m, c, r)$ can be obtained by composition of the set of initial states of $(\mathcal{N}, m, c, r)$ with itself:*

$$S_0(\mathcal{N}, 2 * m, c, r) = S_0(\mathcal{N}, m, c, r) \otimes S_0(\mathcal{N}, m, c, r). \tag{3.11}$$

Proof: Part **(a)** is a consequence of lemma 3, and part **(b)** is a consequence of part **(a)**. □

Theorem 3 *If m is self-independent in $(\mathcal{N}, c, r)$, then:*

$$Sel(\mathcal{N}, k * m) = Sel(\mathcal{N}, m) \otimes Sel(\mathcal{N}, (k-1) * m). \tag{3.12}$$

Proof: Since $E(k*m) = E(m)$, $Sel(k*m)$ can be expressed as follows:

$$\begin{aligned} Sel(k*m) &= \bigcup_{t\in E(m)} Sel(\mathcal{N}, k*m - Inp(t)) \otimes 1_t \\ &= \bigcup_{t\in E(m)} Sel(\mathcal{N}, (k-1)*m \otimes Rem(\mathcal{N}, m))) \otimes 1_t \\ &= Sel(\mathcal{N}, (k-1)*m) \otimes (\bigcup_{t\in E(m)} 1_t) \\ &= Sel(\mathcal{N}, (k-1)*m) \otimes Sel(m) \end{aligned}$$

where $Inp(t)$ is represented as a vector, and 1_t is a $card(T)$–element vector with 1 as the t-th element and zeros elsewhere:

$$\forall\, t' \in T: \; 1_t(t') = \begin{cases} 1, & \text{if } t = t'; \\ 0, & \text{otherwise;} \end{cases}$$

and T is the set of transitions of the net $(\mathcal{N}, m, c, r)$. □

Definition 33 *Initial markings m_i and m_j are* independent *in a net $(\mathcal{N}, c, r)$ if and only if:*

$$\mathcal{G}(\mathcal{N}, m_i + m_j, c, r) \equiv \mathcal{G}(\mathcal{N}, m_i, c, r) \otimes \mathcal{G}(\mathcal{N}, m_j, c, r). \tag{3.13}$$

Lemma 5 *Let m_i and m_j be independent markings in $(\mathcal{N}, c, r)$. Then:*

(a) *The set of selection functions of $(\mathcal{N}, m_i + m_j, c, r)$ can be obtained by composition of the sets of selection functions for m_i and m_j:*

$$Sel(\mathcal{N}, m_i + m_j) = Sel(\mathcal{N}, m_i) \otimes Sel(\mathcal{N}, m_j). \tag{3.14}$$

(b) *The set of initial states of $(\mathcal{N}, m_i + m_j, c, r)$ can be obtained by composition of the sets of initial states of $(\mathcal{N}, m_i, c, r)$ and $(\mathcal{N}, m_j, c, r)$:*

$$S_0(\mathcal{N}, m_i + m_j, c, r) = S_0(\mathcal{N}, m_i, c, r) \otimes S_0(\mathcal{N}, m_j, c, r). \tag{3.15}$$

Proof: The lemma is a straightforward modification of lemma 4 with self–independence replaced by independence. □

Theorem 4 *If m_i and m_j are independent markings in $(\mathcal{N}, c, r)$ and m_i is self-independent in $(\mathcal{N}, c, r)$, then:*

$$Sel(\mathcal{N}, k * m_i + m_j) = Sel(\mathcal{N}, (k-1) * m_i + m_j) \otimes Sel(\mathcal{N}, m_i). \tag{3.16}$$

Proof: Same as for the theorem 3. □

Theorem 5 *If m is self-independent in $(\mathcal{N}, c, r)$, then $2 * m$ and m are independent in $(\mathcal{N}, c, r)$.*

Proof: By induction on reachable states.

Step 1: The sets of initial states are identical:

$$S_0(\mathcal{N}, 3 * m, c, r) = S_0(\mathcal{N}, 2 * m, c, r) \otimes S_0(\mathcal{N}, 2 * m, c, r)$$

because

$$Sel(\mathcal{N}, 3 * m) = Sel(\mathcal{N}, 2 * m) \otimes Sel(\mathcal{N}, m).$$

Step 2: Sets $Inp(t)$ and $Out(t)$ are used here in vector representation.

Assumption: For each state $s_i = (m_i, f_i) \in S(\mathcal{N}, 3 * m, c, r)$ there exist states $s_i'' = (m_i'', f_i'') \in S(\mathcal{N}, 2 * m, c, r)$ and $s_i' = (m_i', f_i') \in S(\mathcal{N}, m, c, r)$ such that $s_i = (m_i'' + m_i', f_i'' + f_i')$. Then for a firing of a transition t which can terminate:

$$\forall\, t \in T: \; f_i(t) > 0 \Rightarrow f_i''(t) > 0 \;\vee\; f_i'(t) > 0$$

and:

$$E(\mathcal{N}, m_i + Out(t)) = \begin{cases} m_i'' + Out(t), & \text{if } f_i''(t) > 0; \\ m_i' + Out(t), & \text{if } f_i'(t) > 0; \end{cases}$$

and:

$$Sel(\mathcal{N}, m_i + Out(t)) = \begin{cases} Sel(\mathcal{N}, m_i'' + Out(t)), & \text{if } f_i''(t) > 0; \\ Sel(\mathcal{N}, m_i' + Out(t)), & \text{if } f_i'(t) > 0. \end{cases}$$

Therefore:

$$\forall\ e_l \in Sel(\mathcal{N}, m_i + Out(t)) :\ s_j = (m_j, f_j) \in S(\mathcal{N}, 3 * m, c, r) \ \wedge$$
$$\begin{cases} s_j'' = (m_j'', f_j'') \in S(\mathcal{N}, 2 * m, c, r), & \text{if } f_i''(t) > 0, \\ s_j' = (m_j', f_j') \in S(\mathcal{N}, m, c, r), & \text{if } f_i'(t) > 0. \end{cases}$$

Once the firing of transition t terminates, other firings can start. The change in markings and firings is expressed as follows:

$$\begin{cases} m_j = m_i + d, \\ m_j'' = m_i'' + d, \\ m_j' = m_i' + d, \end{cases}$$

and

$$\begin{cases} f_j = f_i + e_l - 1_t, \\ f_j'' = f_i'' + e_l = 1_t, \\ f_j' = f_i' + e_l - 1_t, \end{cases}$$

where vector d is defined as:

$$\forall\ p \in P:\ d(p) = -\textstyle\sum_{t \in Out(p)} e_l(t) + \begin{cases} 1, & \text{if } t \in Inp(p); \\ 0, & \text{otherwise.} \end{cases}$$

The new state $s_j \in S$ is thus defined as:

$$s_j = \begin{cases} (m_i' + m_j'', f_i' + f_j''), & \text{if } f_i''(t) > 0, \\ (m_i'' + m_j', f_i'' + f_j'), & \text{if } f_i'(t) > 0. \end{cases}$$

Consequently, for any state in $S(\mathcal{N}, 3 * m, c, r)$, there exists a corresponding state in $S(\mathcal{N}, 2 * m, c, r) \otimes S(\mathcal{N}, m, c, r)$. $\square$

Lemma 6 *If m is a self-independent marking in a net $(\mathcal{N}, c, r)$, then $k * m$ and m are also independent for $k > 1$.*

Proof: This is a straightforward extension of lemma 3. □

Theorem 6 *If m_i and m_j are independent markings in $(\mathcal{N}, c, r)$ and m_i is self-independent in $(\mathcal{N}, c, r)$, then m_i and $m_i + m_j$ are also independent in $(\mathcal{N}, c, r)$.*

Proof: By induction on the states, similarly to the previous theorem.

Step 1: The sets of initial states are the same:

$$S_0(\mathcal{N}, 2 * m_i + m_j, c, r) = S_0(\mathcal{N}, m_i, c, r) \otimes S_0(\mathcal{N}, m_i + m_j, c, r),$$

because:

$$Sel(\mathcal{N}, 2 * m_i + m_j) = Sel(\mathcal{N}, m_i) \otimes Sel(\mathcal{N}, m_i + m_j).$$

Step 2 follows the same line as the previous proof. In conclusion,

$$\mathcal{G}(\mathcal{N}, 2 * m_i + m_j, c, r) \equiv \mathcal{G}(\mathcal{N}, m_i, c, r) \otimes \mathcal{G}(\mathcal{N}, m_i + m_j, c, r). \ \square$$

The following lemma is a generalization of the previous steps:

Lemma 7 *If marking m_i and m_j are independent in $(\mathcal{N}, c, r)$ and both m_i and m_j are self-independent then for $k > 0$ and $l > 0$:*

$$\mathcal{G}(\mathcal{N}, k * m_i + l * m_j, c, r) \equiv \mathcal{G}(\mathcal{N}, k * m_i, c, r) \otimes \mathcal{G}(\mathcal{N}, l * m_j, c, r). \tag{3.17}$$

Proof: The lemma can be shown using induction on k and the results of Theorems 5 and 6 and Lemma 6. □

Definition 34 *Initial markings m_i and m_j in $(\mathcal{N}, c, r)$ are equivalent if and only if:*

$$\mathcal{G}(\mathcal{N}, m_i, c, r) \equiv \mathcal{G}(\mathcal{N}, m_j, c, r). \tag{3.18}$$

Definition 35 *Initial marking m is reducible in $(\mathcal{N}, c, r)$ if there exists another marking m' such that $m' < m$ and:*

$$\mathcal{G}(\mathcal{N}, m, c, r) \equiv \mathcal{G}(\mathcal{N}, m', c, r).$$

Example: Consider the net in Fig. 2.6. Transition firing rates are given in Table 3.1. The initial marking $m'_0 = [1, 1, 0, 0]$ is self-independent. That means that:

$$\mathcal{G}(\mathcal{N}, 2 * m'_0, c, r) = \mathcal{G}(\mathcal{N}, m'_0, c, r) \otimes \mathcal{G}(\mathcal{N}, m'_0, c, r).$$

transition t	firing rate $r(t)$
t_1	10
t_2	15
t_3	20
t_4	25

Table 3.1: Transition firing rates.

The set of reachable states $S(\mathcal{N}, m'_0, c, r)$ and the stationary probabilities are given in Table 3.2. The set $S(\mathcal{N}, 2 * m'_0, c, r)$ is obtained by the composition of state graphs. The results of the composition and the set of reachable states for the same initial marking obtained through reachability analysis are given in Table 3.4 (states and stationary probabilities).

state s	(m, f)	$\pi(s)$
1	$[0, 1, 0, 0; 1, 0, 0, 0]$	0.17647
2	$[0, 0, 0, 0; 0, 1, 0, 0]$	0.17647
3	$[0, 1, 0, 0; 0, 0, 1, 0]$	0.35294
4	$[0, 0, 0, 0; 0, 0, 0, 1]$	0.29412

Table 3.2: State space and stationary probabilities for $m_0 = [1, 1, 0, 0]$.

Fig. 3.1 and Fig 3.2 show the state graphs for $(\mathcal{N}, m_0', c, r)$ and $(\mathcal{N}, 2 * m_0', c, r)$, while Table 3.3 details the correspondence between the state sets obtained through the composition and through reachability analysis.

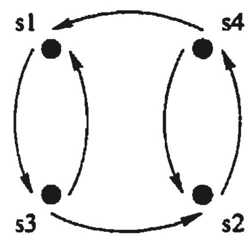

Figure 3.1: State graph for $m_0 = [1, 1, 0, 0]$.

state	(m, f)	state composition
1	[0, 2, 0, 0; 2, 0, 0, 0]	s1+s1
2	[0, 1, 0, 0; 1, 1, 0, 0]	s1+s2=s2+s1
3	[0, 0, 0, 0; 0, 2, 0, 0]	s2+s2
4	[0, 2, 0, 0; 1, 0, 1, 0]	s1+s3=s3+s1
5	[0, 1, 0, 0; 0, 1, 1, 0]	s2+s3=s3+s2
6	[0, 1, 0, 0; 1, 0, 0, 1]	s1+s4=s4+s1
7	[0, 0, 0, 0; 0, 1, 0, 1]	s2+s4=s4+s2
8	[0, 2, 0, 0; 0, 0, 2, 0]	s3+s3
9	[0, 1, 0, 0; 0, 0, 1, 1]	s3+s4=s4+s3
10	[0, 0, 0, 0; 0, 0, 0, 2]	s4+s4

Table 3.3: States for $m_0 = [2, 2, 0, 0]$ obtained by composition and reachability.

On the other hand it can be observed that markings $m_0' = [1, 1, 0, 0]$ and $m_0'' = [1, 0, 0, 0]$ are not independent. State graphs for m_0' and m_0'' are shown in Fig. 3.1 and Fig. 3.3 respectively. Fig. 3.4 shows the state graph for the initial marking $m = m_0' + m_0''$. The dotted arc from state s_2 to state s_5 exists in graph $\mathcal{G}(\mathcal{N}, m_0' + m_0'', c, r)$ but does not exist in the composition $\mathcal{G}(\mathcal{N}, m_0', c, r) \otimes \mathcal{G}(\mathcal{N}, m_0'', c, r)$.

state	reachability analysis	composition
1	0.03114	0.03114
2	0.06228	0.06228
3	0.03114	0.03114
4	0.12457	0.12456
5	0.12457	0.12456
6	0.10381	0.10380
7	0.10381	0.10380
8	0.12457	0.12456
9	0.20761	0.20761
10	0.08651	0.08650

Table 3.4: Stationary probabilities of states for $m_0 = [2, 2, 0, 0]$.

Initial markings $[1, 1, 0, 0]$ and $[0, 1, 1, 0]$ are equivalent and the state graph which is generated by them is shown in Fig. 3.1. It can be observed that the initial marking $[1, k, 0, 0]$, where $k > 0$, is reducible to marking $[1, 1, 0, 0]$, as they generate the same state graph as in Fig. 3.1. □

3.3 Performance measures

Definition 36 *An initial marking m is* decomposable *in net $(\mathcal{N}, c, r)$ if and only if there exist independent markings m_i and m_j such that $m = m_i + m_j$.*

Theorem 7 *If an initial marking m is decomposable into m_i and m_j in $(\mathcal{N}, c, r)$, the performance properties of $(\mathcal{N}, m, c, r)$ can be determined from nets $(\mathcal{N}, m', c, r)$ and $(\mathcal{N}, m'', c, r)$. In particular:*

$$\forall\ t \in T:\ \theta(t) = \theta'(t) + \theta''(t) \tag{3.19}$$

where $\theta_m(t)$ is the throughput of transition t for initial marking m.

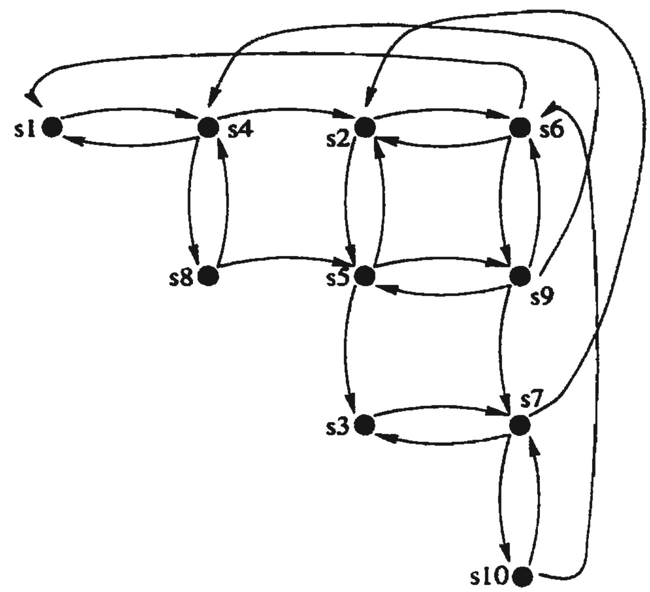

Figure 3.2: State graph for initial marking $2 * m_0$.

Proof: From operational analysis, throughput is defined as:

$$\theta(t) = u(t) * r(t) \tag{3.20}$$

where $u(t)$ is the utilization of a transition and $r(t)$ is the firing rate of that transition. The utilization of a transition is given by (see Chapter 2):

$$u(t) = \sum_{s \in S} \pi(s) * f(t). \tag{3.21}$$

Therefore:

$\theta'(t) = r(t) * \sum_{s'=(m',f') \in S_1} \pi_1(s')$

$\theta''(t) = r(t) * \sum_{s''=(m'',f'') \in S_2} \pi_2(s'')$

For the composed state graph:

Figure 3.3: State graph for initial marking $[1, 0, 0, 0]$.

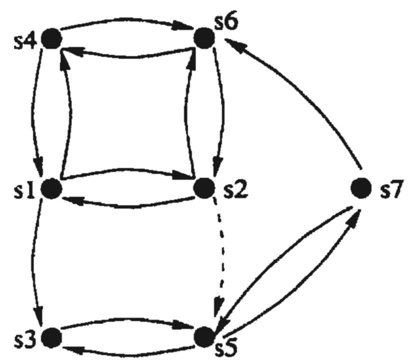

Figure 3.4: State graph for $m = [1, 1, 0, 0]$ and the composition $\mathcal{G}(\mathcal{N}, m_0') \otimes \mathcal{G}(\mathcal{N}, m_0'')$.

$$
\begin{aligned}
\theta(t) &= r(t) * \textstyle\sum_{s=(m,f)\in S} \pi(s) * f(t) \\
&= r(t) * \textstyle\sum_{s'=(m',f')\in S_1} \sum_{s''=(m'',f'')\in S_2} \pi_1(s') * \pi_2(s'') * (f'(t) + f''(t)) \\
&= r(t) * \textstyle\sum_{s'\in S_1} \pi_1(s') * (\sum_{s''\in S_2} \pi_2(s'') * f(t) + \sum_{s''\in S_2} \pi_2(s'') * f''(t)) \\
&= r(t) * \textstyle\sum_{s'\in S_1} \pi_1(s') * f'(t) * \sum_{s''\in S_2} \pi(s'') + \\
&\quad\; r(t) * \textstyle\sum_{s''\in S_2} \pi_2 * f''(t) * \sum_{s'\in S_1} \pi_1(t)
\end{aligned}
$$

From the definition of stationary probabilities,

$$\textstyle\sum_{s\in S} \pi(s) = 1$$

it follows that

$$
\begin{aligned}
\theta(t) &= r(t) * \textstyle\sum_{s'\in S_1} \pi_1(s') * f'(t) + r(t) * \sum_{s''\in S_2} \pi_1(s'') * f''(t) \\
&= \theta'(t) + \theta''(t). \;\square
\end{aligned}
$$

transition	$\theta(t)$ for $[1,1,0,0]$	$\theta(t)$ for $[2,2,0,0]$ (reachability)	$\theta(t)$ for $[2,2,0,0]$ (composition)
1	0.01765	0.03529	0.0353
2	0.01176	0.02353	0.02352
3	0.01765	0.03529	0.0353
4	0.01176	0.02353	0.02352

Table 3.5: Throughputs.

Example: For the previous example of a self-independent marking, Table 3.5 lists the throughput of each transition for initial marking $[1, 1, 0, 0]$. For initial marking $[2, 2, 0, 0]$, the values obtained with the proposed method are compared with the values obtained through reachability analysis. □

Chapter 4

EXAMPLES

This chapter presents the application of the proposed method (presented in Chapter 3) for two different cases. The performance measures computed according to the proposed method are validated by the calculation of the equilibrium probabilities and applying the formulae for performance indices discussed in Chapter 2. The generation of the state graph and the calculation of equilibrium probabilities were performed with the TPN-tools package ([39. 38]).

4.1 Example 1

The net $(\mathcal{N}, c, r)$ in Fig. 4.1 is similar to Example 1 from [31] with few modifications. In our case the net is an M-timed Petri net, and it depicts a slightly different system. The system modeled by the net in Fig. 4.1 is composed of servers servicing two types of jobs: first class and second class jobs. A second class can be serviced only by interrupting a first class job. Other second class jobs can then be serviced as well after which the interrupted first class job resumes. If there are no first class jobs waiting for it, the processor is running a diagnostic repair sequence to detect any possible problems and to solve them.

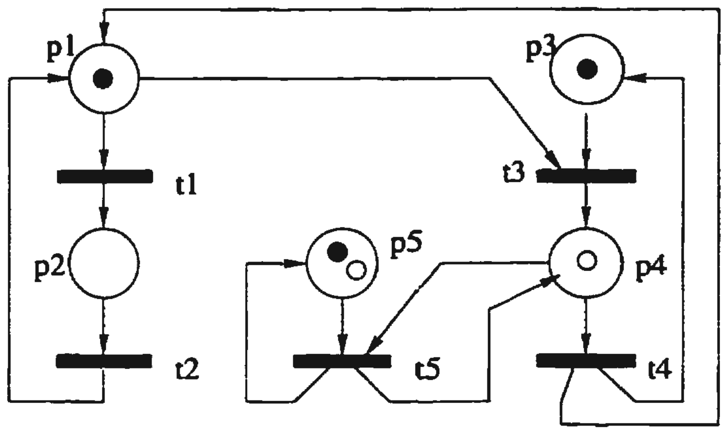

Figure 4.1: Servers with two classes of jobs.

Tokens in place p_1 model the servers being idle. From here they either start servicing a first class job or start on the diagnostic repair sequence. Transitions t_1, t_2 and place p_2 represent the internal activities of the servers. Tokens in places p_3 and p_5 model the waiting queues for first class jobs and respectively for second class jobs. Transition t_3 represents the server running a first class job until it is interrupted by a second class job. If it is not interrupted, then it continues immediately with the activity modeled by transition t_4. Transition t_5 represents a second class job service. Transitions have the following firing rates: $R = \{1, 2, 3, 5, 4\}$ and conflicts are solved by probabilities on the associated arcs: $Prob(p_1, t_1) = 0.2$, $Prob(p_1, t_3) = 0.8$, $Prob(p_4, t_5) = 0.3$ and $Prob(p_5, t_5) = 0.7$.

Initial markings $m_1 = [1, 0, 1, 0, 1]$ and $m_2 = [0, 0, 0, 1, 1]$ are independent and are represented by the black, respectively white tokens. Fig. 4.2 and 4.3 show the state graphs for each initial marking and tables 4.1 and 4.2 give the states and stationary probabilities. The states for the composed graph are given in Table 4.3 and the graph for the composed marking $m_1 + m_2$ is shown in Fig. A.1. The two graphs

state	$s'_i = (m_i, f_i)$	$\pi(s'_i)$
1	$[0,0,1,0,1;1,0,0,0,0]$	0.00997
2	$[0,0,0,0,1;0,0,1,0,0]$	0.11960
3	$[0,0,1,0,1;0,1,0,0,0]$	0.01993
4	$[0,0,0,0,1;0,0,0,1,0]$	0.19934
5	$[0,0,0,0,0;0,0,0,0,1]$	0.65116

Table 4.1: Stationary probabilities for marking $[1,0,1,0,1]$.

state	$s''_i = (m_i, f_i)$	$\pi(s''_i)$
1	$[0,0,0,0,1;0,0,0,1,0]$	0.19934
2	$[0,0,0,0,0;0,0,0,0,1]$	0.65116
3	$[0.0,1,0,1;1,0,0,0,0]$	0.00997
4	$[0,0,0,0,1;0,0,1,0,0]$	0.11960
5	$[0,0,1,0,1;0,1,0,0,0]$	0.01993

Table 4.2: Stationary probabilities for marking $[0,0,0,1,1]$.

state	$s_i = (m_i, f_i)$	composition of states	$\pi(s_i)$
1	$[0,0,1.0,2;1,0,0,1,0]$	$s'_1 + s''_1, s'_4 + s''_3$	0.0039735
2	$[0,0,1,0,1;1,0,0,1,0]$	$s'_1 + s''_2, s'_5 + s''_3$	0.0129800
3	$[0,0,0,0,2;0,0,1,1,0]$	$s'_2 + s''_1, s'_4 + s''_4$	0.0476816
4	$[0,0,0,0,1;0,0,1,0,1]$	$s'_2 + s''_2, s'_5 + s''_4$	0.1557599
5	$[0,0,1,0,2;0,1,0,1,0]$	$s'_3 + s''_1, s'_4 + s''_5$	0.0079469
6	$[0,0,2,0,2;2,0,0,0,0]$	$s_1 + s''_3$	0.0000993
7	$[0,0,1,0,2;1,0,1,0,0]$	$s'_1 + s''_4, s'_2 + s''_3$	0.0023841
8	$[0,0,1,0,1;0,1,0,0,1]$	$s'_3 + s''_2, s'_5 + s''_5$	0.0259600
9	$[0,0,0,0,2;0,0,0,2,0]$	$s'_4 + s''_1$	0.0397347
10	$[0,0,0,0,1;0,0,0,1,1]$	$s'_4 + s''_2, s'_5 + s''_1$	0.2595998
11	$[0,0,0,0,2;0,0,2,0,0]$	$s'_2 + s''_4$	0.0143045
12	$[0,0,0,0,0;0,0,0,0,2]$	$s'_5 + s''_2$	0.4240130
13	$[0,0,2,0,2;1,1,0,0,0]$	$s'_1 + s''_5, s'_3 + s''_3$	0.0003973
14	$[0,0,1,0,2;0,1,1,0,0]$	$s'_3 + s''_4, s'_2 + s''_5$	0.0047682
15	$[0,0,2,0,2;0,2,0,0,0]$	$s'_3 + s''_5$	0.0003973

Table 4.3: State composition.

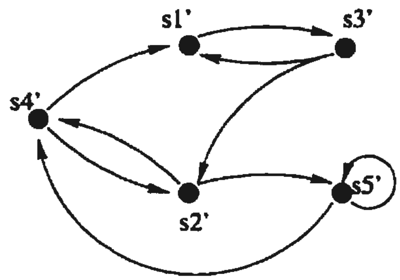

Figure 4.2: State graph for $m_1 = [1, 0, 1, 0, 1]$.

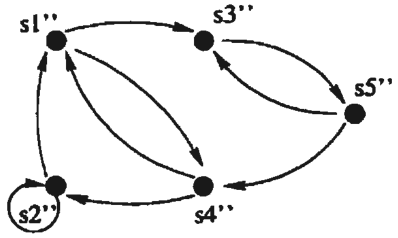

Figure 4.3: State graph for $m_2 = [0, 0, 0, 1, 1]$.

$\mathcal{G}(\mathcal{N}, m_1 + m_2)$ and $\mathcal{G}(\mathcal{N}, m_1) \otimes \mathcal{G}(\mathcal{N}, m_2)$ are isomorphic as the set of states and arcs are equivalent. Appendix A presents detailed information on the set of arcs of the composed graph. Stationary probabilities for the composed graph are obtained from stationary probabilities for the smaller graphs as defined in Chapter 3. For example:

$$\pi(s_1) = \pi(s'_1) * \pi(s''_1) + \pi(s'_4) * \pi(s''_3) = 0.0097 * 0.19934 + 0.19934 * 0.00997 = 0.0039748$$

and

$$\pi(s_6) = \pi(s'_1) * \pi(s''_3) = 0.00997 * 0.00997 = 0.0000994$$

and so forth.

It can also be observed that both m_1 and m_2 are equivalent and self-independent. According to Corollary 10:

transition	$\theta(t)$ for $[3,0,3,2,5]$	$\theta(t)$ (proposed method)
t_1	0.04983	0.04985
t_2	0.04983	0.04985
t_3	0.19934	0.19935
t_4	0.19934	0.19935
t_5	0.46512	0.46510

Table 4.4: Throughputs of transitions for net of Fig. 4.1 with $m_0 = [3, 0, 3, 2, 5]$.

$$\forall\, k, l \in \{1, 2, 3, \ldots\} :\ \mathcal{G}(\mathcal{N}, k * m_1 + l * m_2) = \mathcal{G}(\mathcal{N}, k * m_1) \otimes \mathcal{G}(\mathcal{N}, l * m_2)$$

and

$$\forall\, t \in T :\ \theta_{k * m_1 + l * m_2}(t) = k \cdot \theta_{m_1}(t) + l \cdot \theta_{m_2}(t).$$

Table 4.4 shows this equality for all transitions $t \in T$ and for $k = 3$ and $l = 2$. The results obtained through the proposed method are also validated by the reachability analysis method. It can be observed that the error is less than 0.05%. As was shown in Chapter 3, the performance indices of the net can be obtained from the stationary probabilities which are shown in Table 4.3. Having these probabilities any performance measure of interest can be obtained.

4.2 Example 2

The net shown in Fig. 4.4 is more complex. The system modeled consists of two servers which share a common resource. Whenever the resource is obtained, the service can begin. A first server services two classes of jobs while the other offers the same service for all jobs.

Places p_1 and p_3 represent the two servers, idle, waiting for the resource which is modeled by place p_2. Transition t_2 and t_6 represent the service for one of the servers.

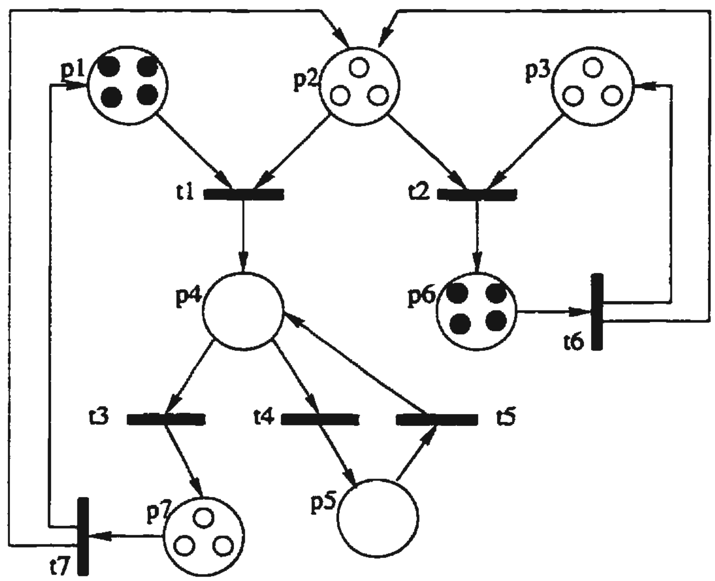

Figure 4.4: System with 2 servers and a common resource.

Transition t_1 represents the amount of time a first class job can run without being interrupted by a second class job. Transitions t_4 and t_5 model the second class job service. At this point it is possible for more than one second class job to "steal" the processor. Transition t_3 models the end of first class job service and the common resource is freed again. It should be noted that the servers are infinite servers and are limited only by the number of available common resources represented by the token count in place p_2.

The firing rates for transitions have the following values: $R = \{10, 3, 15, 5, 6, 7, 2\}$. When transitions t_1 and t_2 are enabled at the same time and a conflict arises, it is solved by the associated probabilities (same for transitions t_3 and t_4): $Prob(p_2, t_1) = 0.4$, $Prob(p_2, t_2) = 0.6$, $Prob(p_4, t_4) = 0.2$ and $Prob(p_4, t_3) = 0.8$.

Markings $m_1 = [1, 1, 1, 0, 0, 0, 0]$ and $m_2 = [1, 0, 0, 0, 0, 1, 0]$ are independent and self-independent. It follows that the net can be analyzed for any linear combination m of marking m_1 and m_2. Thus performance indices for the marking $m = k * m_1 + l * m_2$

for the net in Fig. 4.4 can obtained. Numerical results for transition throughputs are given in Table 4.5 for the case when $k = 2$ and $l = 4$.

transition	$\theta(t) for$ [6, 2, 2, 0, 0, 4, 0] (RA)	$\theta(t)$ (proposed method)
t_1	0.13408	0.13410
t_2	0.20112	0.20112
t_3	0.13408	0.13410
t_4	0.03352	0.03354
t_5	0.03352	0.03354
t_6	0.20112	0.20112
t_7	0.13408	0.13410

Table 4.5: Throughputs of transitions for net of Fig. 4.4 with $m_0 = [4, 2, 2, 0, 0, 4, 2]$.

Chapter 5

CONCLUSIONS

An approach based on decomposition of state graphs is proposed as a method for analysing M-timed Petri nets and obtaining their performance indices. The proposed approach avoids the state explosion problem of reachability analysis by decomposing large state spaces into several smaller components. An overview of recent work on applying some of the analysis methods from queueing theory, like the Product Form Solution and Mean Value Analysis, is also given.

The proposed method is a combination of reachability analysis and state graph composition. The composition of state graphs for the same net and different initial markings is formally defined. Independent and self-independent markings are then introduced, using state graph composition. Independent markings provide the basis for decomposition of large state spaces into simpler components which can be analyzed in isolation. Equations for obtaining performance indices, and in particular transition throughputs, are derived from transition throughputs for component markings. Stationary probabilities of states for the composed state graph are also obtained from the stationary probabilities of component graphs. Since the component nets are quite simple, reachability analysis is straightforward and is not affected by the state explosion problem. The proposed method is thus much more efficient than the reachability analysis directly applied to the net. The approach can be automated by developing software tools for checking independence and composition of the state graphs.

There are many issues which need further investigation. They include:

- How are the markings determined as being (self)-independent without the generation of the state graph for the combined marking? Are there any structural properties that a net has to satisfy for its markings to be independent? The examples used in this thesis are covered by P and T-invariants, but this is clearly insufficient.

- Let $(\mathcal{N}, c, r)$ be an M-timed net and m_1, m_2 and m_3 be mutually independent markings in $\mathcal{N}$. Is it true that:

  $$\mathcal{G}(\mathcal{N}, m_1 + m_2 + m_3, c, r) \equiv \mathcal{G}(\mathcal{N}, m_1, c, r) \otimes \mathcal{G}(\mathcal{N}, m_2, c, r) \otimes \mathcal{G}(\mathcal{N}, m_3, c, r)$$

  Numerical results indicate a positive answer, but a formal proof is needed.

- Can the proposed approach be applied to D-timed nets? The state description of a deterministic timed net is more complicated as it has to consider the history of firings. Some experimental work done with D-timed nets seems to indicate that graph composition is possible, but again, formal proof is needed.

- Can the proposed approach be extended to high level nets, in particular colored nets? The intuition is very simple; if the colored tokens do not "mix", the corresponding colored layers of the net should be independent but again, formal support of this observation is needed.

Independence of markings, used in this thesis for a decomposition of the state space, is a very restrictive property which requires behavioural equivalence between composed state graphs and state graphs of composed markings. It is felt that a less restrictive relation, allowing a "controlled" interaction of components, would be very helpful for further decomposition of state spaces. The approach presented in this thesis can thus be regarded as a starting point for more investigations in this direction.

Bibliography

[1] "http://www.daimi.aau.dk/ petrinet/tools/."

[2] G. Balbo, "On the Success of Stochastic Petri Nets," *Proceedings of the 6-th International Workshop on Petri Nets and Performance Models*, pp. 2–9, 1995.

[3] G. Balbo, "Stochastic Petri Nets: Achievements and Open Problems." *Proceedings of Computer Performance and Dependability Symposium (IPDS)*, pp. 51–60, 1995.

[4] G. Balbo, S. Bruell, and M. Sereno, "Arrival Theorems for Product Form Stochastic Petri Nets," *Proceedings 1994 ACM SIGMETRICS Conference*, pp. 87–97, May 1994.

[5] F. Bause and P. S. Kritzinger, *Stochastic Petri Nets - An Introduction to the Theory.* Verlag Vieweg, 1996.

[6] B. Berthomieu and M. Menasche, "An Enumerative Approach for Analysing Time PN," *Information Processing 83*, pp. 41–45, 1983.

[7] J. L. Coleman, "Algorithms for Product Form Stochastic Petri Nets - A New Approach," *Proceedings of the 5th International Workshop on Petri Net and Performance Models*, pp. 108–116, Oct. 1993.

[8] J. Desel and J. Esparza, *Free Choice Petri Nets.* Cambridge University Press, 1995.

[9] S. Donatelli and M. Sereno, "On the Product Form Solution for Stochastic Petri Nets," in: *Application and Theory of Petri Nets 1992 (Lecture Notes in Computer Science 616)*, pp. 154–172, 1992.

[10] G. Florin and S. Natkin, "Generalization of Queueing Network Product Form Solution to Stochastic Petri Nets," *IEEE Transactions on Software Engineering*, vol. SE-17, pp. 99–107, Feb. 1991.

[11] D. Gross and C. Harris, *Fundamentals of Queueing Theory.* John Wiley & Sons, 1985.

[12] W. Henderson, D. Lucic, and P. Taylor, "A Net Level Performance Analysis of Stochastic Petri Nets," *Journal of Australian Mathematical Society*, pp. 176–187, 1989.

[13] W. Henderson and P. G. Taylor, "Embedded Processes in Stochastic Petri Nets," *IEEE Transactions on Software Engineering*, vol. SE-17, pp. 109–116, Feb. 1991.

[14] M. Holliday and M. Vernon, "A Generalized Timed Petri Net Model for Performance Analysis," *IEEE Transactions on Software Engineering*, vol. SE-13, pp. 1297–1310, 1987.

[15] K. Kant, *Fundamentals of Queueing Theory.* McGraw-Hill, Inc., 1992.

[16] K. Lautenbach and H. Ridder, "Liveness in Bounded Petri Nets Which Are Covered by T-invariants," in: *Application and Theory of Petri Nets 1994 (Lecture Notes in Computer Science 815)*, pp. 358–375, 1994.

[17] A. A. Lazar and T. G. Robertazzi, "Markovian Petri Net Protocols with Product Form Solution," *Performance Evaluation*, vol. 12, pp. 67–77, 1991.

[18] E. Lazowska, J. Zhorjan, G. Graham, and K. C. Sevcik, *Quantitative System Performance - Computer System Analysis using Queueing Network Models*. Prentice-Hall, 1984.

[19] M. A. Marsan, "Stochastic Petri nets - An Elementry Introduction," in: *Advances in Petri Nets 1989 (Lecture Notes in Computer Science 424)*, pp. 1–29, 1989.

[20] M. A. Marsan and G. Chiola, "On Petri nets with Deterministic and Exponential Firing Times," in: *Advances in Petri Nets 1987 (Lecture Notes in Computer Science 266)*, pp. 132–145, 1987.

[21] M. A. Marsan, G. Balbo, and G. Conte, *Performance Models of Multiprocessor Systems*. The MIT Press, 1986.

[22] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G.Franceschinis, *Modelling with Generalized Petri Nets*. John Willey & Sons, 1995.

[23] J. Martinez and M. Silva, "A Simple and Fast Algorithm to obtain all Invariants of a Generalized Petri net," in: *Applications and Theory of Petri Nets (Informatik Fachberichte 52)*, pp. 301–310, 1982.

[24] M. Molloy, *On the Integration of Delay and Throughput Measures in Distributed Processing Models*. PhD thesis, University of California, Los Angeles, 1981.

[25] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, pp. 544–580, Apr. 1989.

[26] W. Reisig, *Petri Nets - An Introduction*. Springer-Verlag, 1985.

[27] W. Reisig, *A Primer in Petri Net Design*. Springer-Verlag, 1992.

[28] M. Sereno, "Approximate Mean Value Analysis Technique for non-Product Form Solution Stochastic Petri Nets: an Application to Stochastic Marked Graphs," *6-th International Workshop on Petri Nets and Performance Models*, pp. 42–51, 1995.

[29] M. Sereno, "Approximate Mean Value Analysis for Stochastic Marked Graphs," *IEEE Transactions on Software Engineering*, vol. SE 22, pp. 654–664, Sept. 1996.

[30] M. Sereno and G. Balbo, "Computational Algorithms for Product Form Stochastic Petri Nets," *Proceedings of the 5th Workshop on Petri Net Performance Models*, pp. 98–107, 1993.

[31] M. Sereno and G. Balbo, "Mean Value Analysis of Stochastic Petri Nets," *Performance Evaluation*, vol. vol. 29, pp. 35–62, 1997.

[32] J. Sifakis, "Use of Petri nets for performance evaluation"; in: *Measuring, modelling and evaluating computer systems*, pp. 75–93, North–Holland, 1977.

[33] P. Ziegler and H. Szczerbicka, "Towards An Algorithm to Compute the Product Form Solution of GSPN models," *Proceedings of the 1995 Summer Computer Simulation Conference*, pp. 24–26, 1995.

[34] W. M. Zuberek, "M-Timed Petri Nets, Priorities, Preemptions, and Performance Evaluation of Systems," in: *Advances in Petri Nets 1985 (Lecture Notes in Computer Science 222)*, pp. 478–498, 1985.

[35] W. M. Zuberek, "D-Timed Petri Nets and Modelling of Timeouts and Protocols," *Transactions of The Society for Computer Simulation*, vol. 4, no. 4, pp. 331–357, 1987.

[36] W. M. Zuberek, "Timed Petri Nets - Definitions, Properties, and Applications," *Microelectronics and Reliability*, vol. 31, pp. 627–644, 1991.

[37] W. M. Zuberek, "Application of Timed Petri Nets to Modeling and Analysis of Flexible Manufacturing Cells," Tech. Rep. 9503, Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada, 1995.

[38] W. M. Zuberek, "Modeling Using Timed Petri Nets - Event-Driven Simulation," Tech. Rep. 9602, Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada, Sept. 1996.

[39] W. M. Zuberek, "Modeling Using Timed Petri Nets - Model Description and Representation," Tech. Rep. 9601, Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada, Aug. 1996.

Appendix A

Graph Composition: Arc Composition

The graph composition for the net shown in Fig. 4.1 is presented in detail here. The arc composition for markings $m_1 = [1, 0, 1, 0, 1]$ and $m_2 = [0, 0, 0, 1, 1]$ is detailed in Table A.1. The state graph for the combined marking $m_1 + m_2$ is represented in Fig. A.1. An examination of the Table and the graph reveals the equality of the two sets of arcs.

Table A.1: Arc composition.

arc in E	composition of arcs
$s_1 \rightarrow s_5$	$(s'_1, s''_1) \rightarrow (s'_3, s''_1) \wedge (s'_4, s''_3) \rightarrow (s'_4, s''_5)$
$s_1 \rightarrow s_6$	$(s'_4, s''_3) \rightarrow (s'_1, s''_3) \wedge (s'_1, s''_1) \rightarrow (s'_1, s''_3)$
$s_1 \rightarrow s_7$	$(s'_4, s''_3) \rightarrow (s'_2, s''_3) \wedge (s'_1, s''_4) \rightarrow (s'_1, s''_1)$
$s_2 \rightarrow s_8$	$(s'_1, s''_2) \rightarrow (s'_3, s''_2) \wedge (s'_5, s''_3) \rightarrow (s'_5, s''_5)$
$s_2 \rightarrow s_1$	$(s'_5, s''_3) \rightarrow (s'_4, s''_3) \wedge (s'_1, s''_2) \rightarrow (s'_1, s''_1)$
$s_2 \rightarrow s_2$	$(s'_5, s''_3) \rightarrow (s'_5, s''_3) \wedge (s'_1, s''_2) \rightarrow (s'_1, s''_2)$
$s_3 \rightarrow s_9$	$(s'_2, s''_1) \rightarrow (s'_4, s''_1) \wedge (s'_4, s''_4) \rightarrow (s'_4, s''_1)$
$s_3 \rightarrow s_{10}$	$(s'_2, s''_1) \rightarrow (s'_5, s''_1) \wedge (s'_4, s''_4) \rightarrow (s'_4, s''_2)$

Table A.1: Arc composition.

arc in E	composition of arcs
$s_3 \to s_7$	$(s'_4, s''_4) \to (s'_1, s''_4) \wedge (s'_2, s''_1) \to (s'_1, s''_3)$
$s_3 \to s_{11}$	$(s'_4, s''_4) \to (s'_2, s''_4) \wedge (s'_2, s''_1) \to (s'_2, s''_4)$
$s_4 \to s_{10}$	$(s'_2, s''_2) \to (s'_4, s''_2) \wedge (s'_5, s''_4) \to (s'_5, s''_1)$
$s_4 \to s_{12}$	$(s'_2, s''_2) \to (s'_5, s''_2) \wedge (s'_4, s''_2) \to (s'_4, s''_5)$
$s_4 \to s_3$	$(s'_5, s''_4) \to (s'_4, s''_4) \wedge (s'_2, s''_1) \to (s'_2, s''_2)$
$s_4 \to s_4$	$(s'_5, s''_4) \to (s'_5, s''_4) \wedge (s'_2, s''_2) \to (s'_2, s''_2)$
$s_5 \to s_1$	$(s'_3, s''_1) \to (s'_1, s''_1) \wedge (s'_5, s''_3) \to (s'_5, s''_4)$
$s_5 \to s_3$	$(s'_3, s''_1) \to (s'_1, s''_1) \wedge (s'_5, s''_3) \to (s'_5, s''_4)$
$s_5 \to s_{13}$	$(s'_4, s''_5) \to (s'_1, s''_5) \wedge (s'_3, s''_1) \to (s'_3, s''_3)$
$s_5 \to s_{14}$	$(s'_4, s''_5) \to (s'_2, s''_5) \wedge (s'_3, s''_1) \to (s'_3, s''_4)$
$s_6 \to s_{13}$	$(s'_1, s''_3) \to (s'_3, s''_3) \wedge (s'_3, s''_5) \to (s'_3, s''_1)$
$s_7 \to s_{14}$	$(s'_1, s''_4) \to (s'_3, s''_4) \wedge (s'_3, s''_5) \to (s'_3, s''_2)$
$s_7 \to s_1$	$(s'_2, s''_3) \to (s'_4, s''_3) \wedge (s'_1, s''_4) \to (s'_1, s''_1)$
$s_7 \to s_2$	$(s'_2, s''_3) \to (s'_5, s''_3) \wedge (s'_4, s''_2) \to (s'_4, s''_1)$
$s_8 \to s_2$	$(s'_3, s''_2) \to (s'_1, s''_2) \wedge (s'_5, s''_3) \to (s'_5, s''_5)$
$s_8 \to s_4$	$(s'_3, s''_2) \to (s'_2, s''_2) \wedge (s'_5, s''_4) \to (s'_5, s''_5)$
$s_8 \to s_5$	$(s'_5, s''_5) \to (s'_4, s''_5) \wedge (s'_2, s''_1) \to (s'_2, s''_3)$
$s_8 \to s_8$	$(s'_5, s''_5) \to (s'_5, s''_5) \wedge (s'_3, s''_2) \to (s'_3, s''_2)$
$s_9 \to s_1$	$(s'_4, s''_1) \to (s'_1, s''_1) \wedge (s'_1, s''_3) \to (s'_1, s''_4)$
$s_9 \to s_3$	$(s'_4, s''_1) \to (s'_2, s''_1) \wedge (s'_1, s''_4) \to (s'_1, s''_4)$
$s_{10} \to s_2$	$(s'_4, s''_2) \to (s'_1, s''_2) \wedge (s'_5, s''_1) \to (s'_5, s''_3)$
$s_{10} \to s_4$	$(s'_4, s''_2) \to (s'_2, s''_2) \wedge (s'_5, s''_1) \to (s'_5, s''_4)$
$s_{10} \to s_9$	$(s'_5, s''_1) \to (s'_4, s''_1) \wedge (s'_4, s''_2) \to (s'_4, s''_1)$

Table A.1: Arc composition.

arc in E	composition of arcs
$s_{10} \rightarrow s_{10}$	$(s'_5, s''_1) \rightarrow (s'_5, s''_1) \wedge (s'_4, s''_2) \rightarrow (s'_4, s''_2)$
$s_{11} \rightarrow s_3$	$(s'_2, s''_4) \rightarrow (s'_4, s''_4) \wedge (s'_4, s''_1) \rightarrow (s'_4, s''_2)$
$s_{11} \rightarrow s_4$	$(s'_2, s''_4) \rightarrow (s'_5, s''_4) \wedge (s'_4, s''_2) \rightarrow (s'_4, s''_2)$
$s_{12} \rightarrow s_{10}$	$(s'_5, s''_2) \rightarrow (s'_4, s''_2) \wedge (s'_5, s''_2) \rightarrow (s'_5, s''_1)$
$s_{12} \rightarrow s_{12}$	$(s'_5, s''_2) \rightarrow (s'_5, s''_2) \wedge (s'_5, s''_2) \rightarrow (s'_5, s''_2)$
$s_{13} \rightarrow s_{15}$	$(s'_1, s''_5) \rightarrow (s'_3, s''_5) \wedge (s'_3, s''_3) \rightarrow (s'_3, s''_5)$
$s_{13} \rightarrow s_6$	$(s'_3, s''_3) \rightarrow (s'_1, s''_3) \wedge (s'_1, s''_5) \rightarrow (s'_1, s''_3)$
$s_{13} \rightarrow s_7$	$(s'_3, s''_3) \rightarrow (s'_2, s''_3) \wedge (s'_1, s''_5) \rightarrow (s'_1, s''_4)$
$s_{14} \rightarrow s_7$	$(s'_3, s''_4) \rightarrow (s'_1, s''_4) \wedge (s'_2, s''_5) \rightarrow (s'_2, s''_3)$
$s_{14} \rightarrow s_{11}$	$(s'_3, s''_4) \rightarrow (s'_2, s''_4) \wedge (s'_2, s''_5) \rightarrow (s'_2, s''_4)$
$s_{14} \rightarrow s_5$	$(s'_2, s''_5) \rightarrow (s'_4, s''_5) \wedge (s'_3, s''_4) \rightarrow (s'_3, s''_1)$
$s_{14} \rightarrow s_8$	$(s'_2, s''_5) \rightarrow (s'_5, s''_5) \wedge (s'_3, s''_4) \rightarrow (s'_3, s''_2)$
$s_{15} \rightarrow s_{13}$	$(s'_3, s''_5) \rightarrow (s'_1, s''_5) \wedge (s'_3, s''_5) \rightarrow (s'_3, s''_3)$
$s_{15} \rightarrow s_{14}$	$(s'_3, s''_5) \rightarrow (s'_2, s''_5) \wedge (s'_3, s''_5) \rightarrow (s'_3, s''_4)$

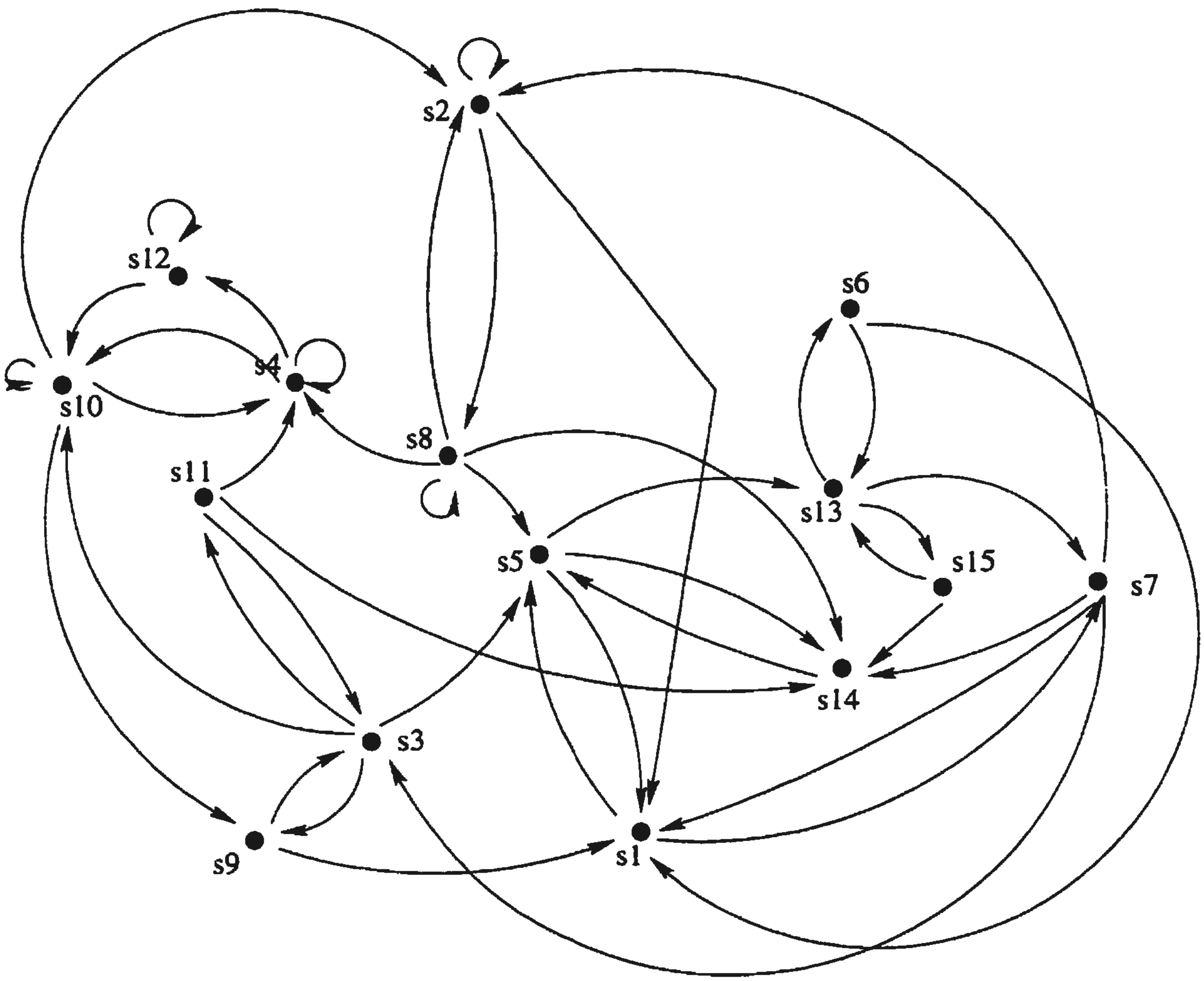

Figure A.1: State graph for the composed marking $[1, 0, 1, 1, 2]$.