Performance Bounds for Synchronized Queueing Networks

Javier Campos Laclaustra

Tesis Doctoral

Departamento de Ingeniería Eléctrica e Informática
Universidad de Zaragoza

October 1990

No es dado a todos aventurarse en la selva y trazar, a fuerza de energía, un camino practicable, pero aun los más humildes podemos aprovecharnos del sendero abierto por el genio, y arrancar, caminando por él, algún secreto a lo desconocido.

Santiago Ramón y Cajal
Los tónicos de la voluntad, 1897

Contents

List of Figures

List of Tables

Preface

Product form queueing networks have long been used for the performance evaluation of computer systems. Their success has been due to their capability of naturally expressing sharing of resources and queueing, that are typical situations of traditional computer systems, as well as to their efficient solution algorithms, of polynomial complexity on the size of the model. Unfortunately, the introduction of synchronization constraints usually destroys the product form solution, so that general concurrent and distributed systems are not easily studied with this class of models.

Petri nets have been proved specially adequate to model parallel and distributed systems. Moreover, they have a well-founded theory of analysis that allows to investigate a great number of qualitative properties of the system.

In the original definition, Petri nets did not include the notion of time, and tried to model only the logical behaviour of systems by describing the causal relations existing among events. This approach showed its power in the specification and analysis of concurrent systems in a way independent of the concept of time. Nevertheless the introduction of a timing specification is essential if we want to use this class of models for the performance evaluation of distributed systems.

One of the main problems in the actual use of timed and stochastic Petri net models for the quantitative evaluation of large systems is the explosion of the computational complexity of the analysis algorithms. In general, exact performance results are obtained from the numerical solution of a continuous time Markov chain, whose dimension is given by the size of the state space of the model. Structural computation of exact performance measures has been possible for some subclasses of nets such as those with state machine topology. These nets, under certain assumptions on the stochastic interpretation are isomorphic to Gordon and Newell's networks, in queueing theory terminology. In the general case, efficient methods for the derivation of performance measures are still needed.

Two complementary approaches to the derivation of exact measures for the analysis of distributed systems are the utilization of approximation techniques and the computation of bounds. Approximate values for the performance parameters are in general more efficiently derived than the exact ones. On the other hand, "exactness" only exists in theory! In other words, numerical algorithms must be applied in practice for the computation of exact values, therefore making errors is inevitable.

Performance bounds are useful in the preliminary phases of the design of a system, in which many parameters are not known accurately. Several alternatives for those parameters should be quickly evaluated, and rejected those that are clearly bad. Exact (and even approximate) solutions would be computationally very expensive. Bounds become useful in these instances since they usually require much less computation effort.

The computation of upper and lower bounds for the steady-state performance of timed and stochastic Petri nets is considered in this work. In particular, we study the throughput of transitions, defined as the average number of firings per time unit. For this measure we try to compute upper and lower bounds in polynomial time on the size of the net model, by means of proper linear programming problems defined from the incidence matrix of the net (in this sense, we develop structural techniques). These bounds depend only on the mean values and not on the higher moments of the probability distribution functions of the random variables that describe the timing of the system. The independence of the probability distributions can be viewed as a useful generalization of the performance results, since higher moments of the delays are usually unknown for real cases, and difficult to estimate and assess.

From a different perspective, the obtained results can be applied to the analysis of queueing networks extended with some synchronization schemes. Monoclass queueing networks can be mapped on stochastic Petri nets. On the other hand, stochastic Petri nets can be interpreted as monoclass queueing networks augmented with synchronization primitives.

Concerning the presentation of this manuscript, it should be mentioned that chapter 1 has been written with the object of giving the reader an outline of the stochastic Petri net model: its definition, terminology, basic properties, and related concepts, together with its deep relation with other classic stochastic network models.

Chapter 2 is devoted to the presentation of the net subclasses considered in the rest of the work. The classification presented here is quite different from the one which is usual in the framework of Petri nets. The reason lies on the fact that our classification criterion, the computability of visit ratios for transitions, is introduced for the first time in the field of stochastic Petri nets in this work. The significance of that criterion is based on the important role that the visit ratios play in the computation of upper and lower bounds for the performance of the models. Nevertheless, classical important net subclasses are identified here in terms of the computability of their visit ratios from different parameters of the model.

Chapter 3 is concerned with the computation of reachable upper and lower bounds for the most restrictive subclass of those presented in chapter 2: marked graphs. The explanation of this fact is easy to understand. The more simple is the model the more accessible will be the techniques an ideas for the development of good results.

Chapter 4 provides a generalization for live and bounded free choice nets of the results presented in the previous chapter. Quality of obtained bounds is similar to that for strongly connected marked graphs: throughput lower bounds are reachable for bounded nets while upper bounds are reachable for 1–bounded nets.

Chapter 5 considers the extension to other net subclasses, like mono-T-semiflow nets, FRT-nets, totally open deterministic systems of sequential processes, and persistent nets. The results are of diverse colours. For mono-T-semiflow nets and, therefore, for general FRT-nets, it is not possible (so far) to obtain reachable throughput bounds. On the other hand, for bounded ordinary persistent nets, tight throughput upper bounds are derived. Moreover, in the case of totally open deterministic systems of sequential processes the exact steady-state performance measures can be computed in polynomial time on the net size.

In chapter 6 bounds for other interesting performance measures are derived from throughput bounds and from classical queueing theory laws. After that, we explore the introduction of more information from the probability distribution functions of service times in order to improve the bounds. In particular, for Coxian service delay of transitions it is possible to improve the throughput upper bounds of previous chapters which held for more general forms of distribution functions. This improvement shows to be specially fruitful for live and bounded free choice nets.

Chapter 7 is devoted to case studies. Several examples taken from literature in the fields of distributed computing systems and manufacturing systems are modelled by means of stochastic Petri nets and evaluated using the techniques developed in previous chapters.

Finally, some concluding remarks and considerations on possible extensions of the work are presented.

Acknowledgements

It is a pleasure to acknowledge a few of my debts. Like all Spanish Petri nets' researchers, I owe everything to Manuel Silva; friend, colleague, and advisor of this thesis, to whom go my foremost thanks, for his constant guidance and help.

I would also like to thank Miguel San Miguel, who first aroused my interest in applied stochastic processes and queueing theory.

I must say how grateful I am to all the colleagues of the *Grupo de Ingeniería de Sistemas e Informática* of the *Universidad de Zaragoza*, and especially to José Manuel Colom, who has offered constructive comments and discussions during the past years.

Other persons have contributed to the derivation of the results presented in this manuscript. Special thanks go to Giovanni Chiola, for his fruitful cooperation.

Finally, to my family and friends. I am eternally grateful for their patience and support.

Javier Campos

Zaragoza, October 1990.

Chapter 1

Synchronized queueing networks and Petri nets

Queueing network models are one of the most popular and classical tools for the performance evaluation of computer systems. With the advent of complex distributed systems, many proposals have been made to extend the modelling power of queueing networks by adding various synchronization mechanisms to the basic model. One of the most important characteristics of basic queueing networks that determined their popularity was the development of efficient (polynomial complexity) algorithms, based on their "product form solution". Unfortunately, the introduction of synchronization mechanisms usually destroys this nice property.

More recently, timed and/or stochastic Petri net models have been introduced as a modelling tool capable of naturally represent synchronization and concurrency. The intimate relation between synchronized queueing networks and stochastic Petri nets is stressed in this chapter. After an historical route through the main hits of queueing networks theory, we justify the necessity of the introduction of synchronization schemes for the performance evaluation of distributed systems. Then, we formally introduce the model of Petri nets, as well as the different implications that the addition of a timing interpretation has in the model. Finally, the close relations between queueing networks with synchronization constraints and stochastic Petri nets are remarked.

1.1 Queueing networks with synchronizations

Queueing network models have been used for performance evaluation since the early work of A. Erlang [Erl09]. Their success for the analysis of computer systems (see, e.g., [Kle76,LZGS84,Lav89]) has been due to their capability of naturally expressing sharing of resources and queueing, that are typical situations of traditional computer systems, as well as to their efficient solution algorithms, of polynomial complexity on the size of the model.

1.1.1 Monoclass queueing networks

A queueing network model of a system is a collection of *service centers* or *stations* and *customers* moving among them. The service centers represent different processing sites while customers represent jobs or processes. Customers can *enter* the system at certain points; after that they *move* from one station to another, queueing up at each for some service; and ocassionally they *depart* from the system.

More formally, a queueing network is a trio $\langle \mathcal{SC}, R, X_0 \rangle$, where

- $\mathcal{SC} = \{1, \ldots, m\}$ is the set of service centers,
- R is the real matrix of *routing probabilities* $r_{ij} \geq 0$; $i, j = 1, \ldots, m$; where r_{ij} is the probability that a customer exiting center i goes to j, and
- X_0 is the vector of external arrival rates $X_{0i} \geq 0$, $i = 1, \ldots, m$, to stations.

If $X_{0i} = 0$ for all station i, the number of customers in the network remains constant, it is denoted as N, and the system is called *closed* network. Otherwise, the network is said to be *open*.

A queueing network can be seen as a *directed graph* in which service centers are the nodes. An arc from node i to node j is drawn iff $r_{ij} > 0$. As an example, see the closed network depicted in figure 1.1, that models a simple computer system with virtual memory [GP87]. In this

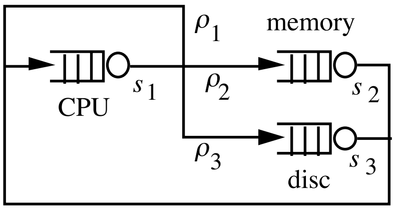


Figure 1.1: A simple computer system with virtual memory.

case, if *CPU*, *memory*, and *disc* are labelled with indexes 1,2, and 3, respectively, we have

$$R = \begin{pmatrix} \rho_1 & \rho_2 & \rho_3 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{1.1}$$

In fact, since each node in the system is a service center with a storage room for queues to form, a queueing network can be seen also as a *bipartite* directed graph. Service centers and storage rooms are the two kinds of nodes. An arc exists from each storage room to its corresponding service center. Finally, an arc from the service center i to the storage room preceding center j is drawn iff $r_{ij} > 0$.

The *state* of the network is defined by a vector $\vec{n} = (n_1, \ldots, n_m)^T$, where n_i is the number of customers at center i (including those being served and those waiting).

In order to completely define the model, the queueing disciplines at each of the storage rooms, the intensity of arrivals from outside, the service requirements of jobs at centers, and specially the *average service time* s_i of each station i must be specified. When all the above parameters are "appropriately" defined the evolution of the system can be modelled by a continuous time Markov chain [Rev84]. In this case the limit, or *stationary*, state distribution can be found, if it exists, by solving a system of linear equations, called *global balance equations*, which, for each state, equates the rate of flow into to the rate of flow out of the state. Unfortunately, the number of states (and therefore the dimension of the system of equations) increases quickly when the number of customers and stations grows.

The following system of equations [Kle75] can be derived from the global balance property:

$$\overline{X}(j) = X_{0j} + \sum_{i=1}^{m} \overline{X}(i) r_{ij} \qquad j = 1, \ldots, m \tag{1.2}$$

where $\overline{X}(i)$ is the limit *throughput* of station i, i.e., the average number of service completions per unit time at station i.

If the network is open (i.e., if there exists a station j with positive external arrival rate, $X_{0j} > 0$), then the above m equations are linearly independent, and the exact throughputs of stations can be derived (independently of the service times). This is not the case for closed networks. If $X_{0j} = 0$, $j = 1, \ldots, m$, then only $m - 1$ equations are linearly independent, and thus only ratios of throughputs can be determined. These *relative throughputs* which are often called *visit ratios*, denoted as v_i for each station i, summarize all the information given by the routing probabilities that is necessary in most cases for the computation of the performance measures. The visit ratios normalized, for instance, for station 1 are defined as:

$$v_i^{(1)} \overset{\text{def}}{=} \frac{\overline{X}(i)}{\overline{X}(1)} \qquad i = 1, \ldots, m \tag{1.3}$$

For a restricted class of networks, called *product form networks*, the solution to the global balance equations can be shown to be a product of terms, one for each station, where the form of each term is explicitly given. This fact occurs when the system satisfies the *local balance equations* [Cha72]. Informally, a local balance equation asserts that for any two adjacent states the effective flow from one to the other must be equal to the effective flow in the other direction.

J. Jackson [Jac63] found the first product form solution in a general network of queues, motivated by manufacturing applications. J. Jackson considered open monoclass networks with a Markovian arrival process dependent on the total population of the network. Service disciplines are FCFS (*first-come first-served*) and service times are exponential (with queue length dependent rates). W. Gordon and G. Newell [GN67] extended Jackson's results to cover closed networks.

The steady-state probability $p(\vec{n})$ of state $\vec{n} = (n_1, \ldots, n_m)^T$ in a closed product form queueing network with m stations and N customers has the form:

$$p(n_1, \ldots, n_m) = \frac{1}{G(N)} \prod_{i=1}^{m} (D_i)^{n_i} \tag{1.4}$$

where D_i is the *average service demand* of customers from station i or *loading* of i, defined as:

$$D_i \stackrel{\text{def}}{=} v_i s_i \quad i = 1, \ldots, m \tag{1.5}$$

and $G(N)$ is a *normalization constant* defined so that the $p(\vec{n})$ sum to 1:

$$G(N) \stackrel{\text{def}}{=} \sum_{\vec{n} \in S(N,m)} \prod_{i=1}^{m} (D_i)^{n_i} \tag{1.6}$$

with $S(N, m)$ the set of *feasible* or *reachable* states for the network, i.e., those states $\vec{n}$ with $n_i \geq 0$ and $\sum_i n_i = N$.

We remark that the knowledge of average service demands D_i is crucial for the computation of exact measures of product form queueing networks. On the other hand, the normalization constant $G(N)$ depends on the number of states, therefore the direct "brute force" algorithm for its computation can be really inefficient.

In 1975, F. Baskett, K. Chandy, R. Muntz, and F. Palacios [BCMP75] proved the product form solution for *mixed networks with multiple chains* (meaning networks with both open and closed chains, or types of jobs). BCMP networks include also new service disciplines like *infinite-servers* (IS), *processor sharing* (PS), *last-come first-served preemptive-resume* (LCFSPR), in addition to FCFS. Moreover, general service times distributions are allowed at IS, PS, and LCFSPR service stations.

A somewhat different, but in many ways equivalent, formulation of product form networks was introduced by F. Kelly [Kel76b], based on *quasi-reversibility* of stochastic processes, an equivalent property to that of local balance.

Although several extensions of BCMP networks have been presented in the last years, these extensions have not been found to be particularly useful in applications. Therefore the BCMP formulation has become the most widely used in practice. The main reason for this success is probably the existence of *efficient algorithms* for deriving the solution from product form equations. J. Buzen [Buz73] was the first to develop a computationally efficient algorithm, called the *convolution algorithm*, focussed on the computation of the normalization constant, for Gordon and Newell's class of closed networks. The algorithm was extended to BCMP networks by M. Reiser and H. Kobayashi in [RK75]. The next major advance in product form queueing networks was the *mean value analysis* algorithm, presented in [RL80,RL81] by M. Reiser and S. Lavenberg. It is not centred in the computation of the normalization constant. Instead of this, a recursive relationship that gives performance parameters at population N in terms of those at population $N-1$ is used. It was first developed for closed networks with fixed rate and queue length dependent rates only, but has been extended to cover a broader range of product form networks.

1.1.2 Addition of synchronization schemes

One major limitation of product form queueing networks for computer performance modelling applications is that, in such real systems, cooperation and competence relationships among different processes are usual. Product form queueing models do not allow the explicit representation of *synchronization primitives*, that are necessary for expressing the previously mentioned relationships. In particular, in models of computer systems, certain *blocking phenomena* frequently arise because a job requires more than one resource before it can be processed. The following examples are mentioned in [HL84]:

1) Holding a channel and a disk drive before data transfer can occur.

2) Obtaining a memory partition before job processing can occur.

3) Obtaining a database lock before the data item can be read from disk.

The limitation of queueing networks for the modelling of the above examples lies on the fact that this model has not a general construct for representing synchronization. Many extensions (e.g., [Kel76a,SMK82,

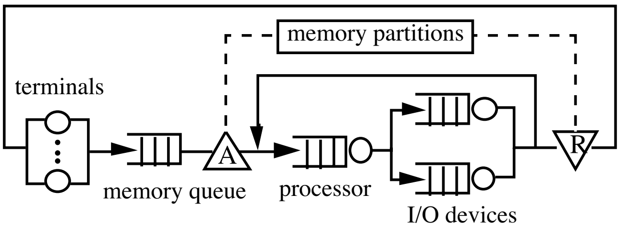


Figure 1.2: Queueing network model of a multiprogramming memory limited system.

Mai87]) have been proposed to introduce *synchronization primitives* into the queueing network formalism, in order to allow the modelling of distributed asynchronous systems: *passive resources*, *fork and join*, *customer splitting*, etc.

As an example, let us consider the model in figure 1.2. It represents a memory limited system in which each process requests a fixed amount of memory before entering the central subsystem composed by the CPU and several I/O devices. This memory (which is called a passive resource) is needed for loading the program to be run as a result of the command issued by a terminal user. After the end of the command execution, the allocated memory is released. Product form solution cannot be derived for this model, because of the blocking caused by the memory limitation.

Recent operating systems, like the one of Cray X-MP and the IBM's MVS, contain a multitasking feature that allows a job to spawn multiple tasks that can execute in parallel on a multiprocessor system. A spawning job waits until all of its spawned tasks have completed service before resuming the execution. Single chain closed queueing networks including *fork* and *join* primitives constitute a suitable model for the representation of this multitasking feature. This kind of model is considered in [HT83]. An example is depicted in figure 1.3, using RESQ representation [SMK82] for the fork and join primitives.

Some very restricted forms of synchronization, such as, for instance, some special use of *passive resources* [AMBCD86,LB86], preserve the local balance property that allows efficient algorithms to be used for the computation of exact product form solution. In general, however, these extensions destroy the local balance property so that queueing models

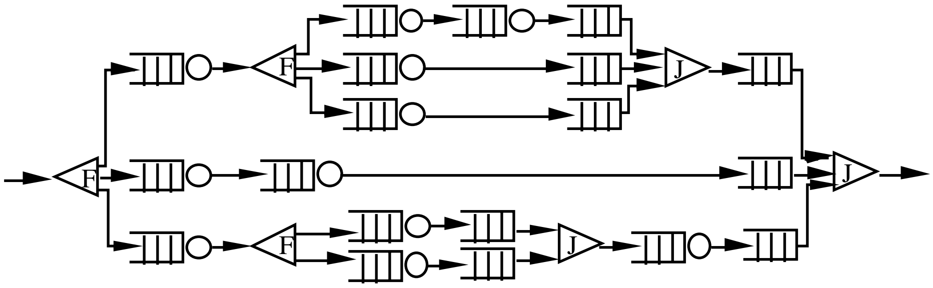


Figure 1.3: Queueing network model of a fork/join multitasking process.

extended with synchronizations are used mainly as system descriptions for simulation experiments [SMK82].

A common factor to all above referred extensions of networks with synchronization primitives is that a formal unified definition in graph theory terms is not as easy to derive as in the case of classic queueing networks. In the next section, *stochastic Petri nets are proposed as a unified model including several extensions of queueing networks with synchronization schemes and preserving a formal and simple interpretation in graph terms* which allows the use of graph theory and linear algebra techniques for their analysis.

1.2 Stochastic Petri nets

The interest in parallel and distributed systems grows constantly according to their new domains of application. One of the main problems arising from these systems is their complexity that implies a stressed necessity for analysis techniques of properties of good behaviour before the implementation.

Petri nets have been proved specially adequate to model parallel and distributed systems. Moreover, they have a well-founded theory of qualitative analysis that allows to investigate a great number of properties of the system.

In this section, after an informal introduction to Petri nets, we recall fundamental definitions, some of the basic properties, as well as the timing specification of the model that converts it to a suitable performance analysis tool for distributed and concurrent systems.

1.2.1 Introducing nets

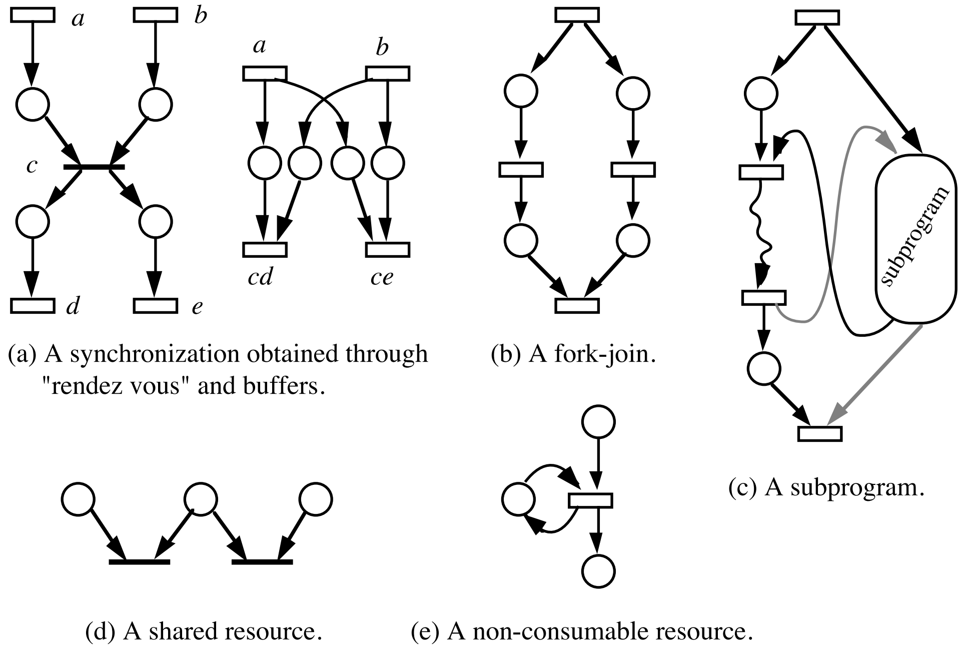
Petri nets [Pet66] are a well-known formalism for describing concurrent discrete event systems with synchronizations (see, e.g., [Pet81,Sil85, Mur89]). Petri net model is oriented to the description of both *states* of a system and *actions* producing evolution through the states. In this sense, it differs from other formal models of concurrent systems which usually are state-based or action-based. Petri nets treat states and actions on equal footing. In fact, the *structure* of a Petri net can be seen as a *bipartite graph* whose two different kind of nodes, *places* and *transitions*, correspond with states and actions of the system. Certain similarity with queueing models can be observed at this point. Storage rooms and service stations of queueing networks represent also states and actions, respectively.

In queueing models the state of the system is represented by means of a given distribution of customers at storage rooms (queues). In an analogous way, a *marking* or distribution of *tokens* (marks) over the places of the Petri net defines the state of the system. Therefore, as for queueing networks, the representation of a state is *distributed.*

The behaviour of a queueing network is governed by the departures of customers from stations, after finishing service, and the movement towards other storage rooms. The *token game* is the analogue in Petri net models. Tokens are stored at places and the *firing* of a transition produces a change of the distribution of tokens (new marking).

The first main property of Petri net model for the description of concurrent systems is its *simplicity.* A very few and simple mathematical entities are necessary for the formal definition of nets. This fact constitute a great advantage, mainly in the modelling of concurrent systems which are enough complicated per se.

In spite of the simplicity of the model, its *generality* must be remarked. The three basic schemes in the modelling of concurrent systems can be included in the Petri net structure: *sequencing*, *choice*, and *concurrency.* Moreover, other typical and well-known elements in the modelling of distributed systems, as “rendez-vous”, shared resources, fork-joins. . . , can be easily derived by combination of the basic schemes



(a) A synchronization obtained through "rendez vous" and buffers.

(b) A fork-join.

(c) A subprogram.

(d) A shared resource.

(e) A non-consumable resource.

Figure 1.4: Typical schemes in the modelling of distributed systems.

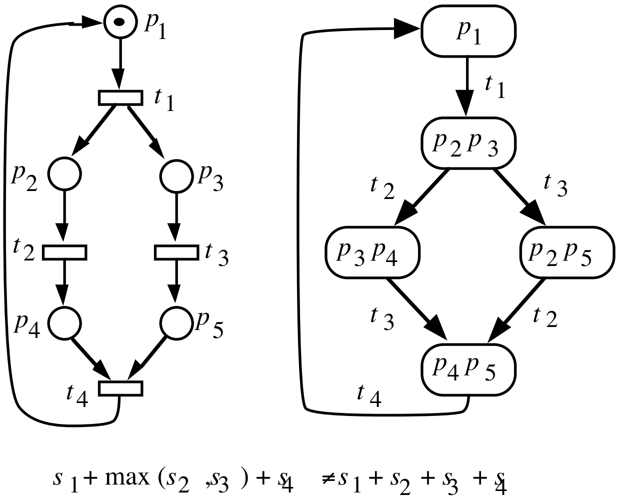


Figure 1.5: Partial ordel formalism and temporal realism.

(see figure 1.4). In this direction, Petri nets improve clearly the modelling power of classic queueing networks, for which synchronizations are difficult or impossible to express, except for some extended formalisms (see section 1.1.2).

One aspect of the *adequacy* of Petri net models is their possibility of expressing all basic semantics of concurrency, *interleaving*, *step*, and *partial order semantics*, which can be compared within the Petri net formalism. In this sense, Petri nets are capable of modelling *"true concurrency"*. The importance of true concurrency in a performance oriented concurrent model can be explained from the *temporal realism* that provides step and partial order semantics of concurrent events. Let us briefly describe these considerations with the use of the net depicted in figure 1.5. Activities modelled with transitions t_2 and t_3 are truly concurrent. This means that the completion time of both is $\max\{\gamma_2, \gamma_3\}$ if γ_2 and γ_3 are their respective random service times, and not $\gamma_2 + \gamma_3$ that would be obtained with interleaving semantics (and could be thought at first glance from a direct interpretation of the reachability graph, which represents a complete sequentialization of the behaviour of the model).

Locality of states and actions constitutes another aspect of adequacy for the modelling of concurrent systems. It provides the possibility of progressive modelling by using *stepwise refinements* (top-down) or *modular composition* (bottom-up modelling).

As in the case of queueing network models, the *graphical representation* of Petri nets is being crucial for the increasing interest of systems designers in this model. However, distributed and concurrent systems are complex and difficult to master for designers by nature. Therefore, desirable "good properties" must be formally defined and the model must be *validated* for these properties. In this sense, *qualitative analysis* of Petri nets is important before going on the implementation. A wide range of techniques for checking *synchronic* (*lead*, *distance*, *places bounds*, *places mutual exclusions*...) and *activity* properties (*deadlock-freeness*, *liveness*, *home states*...) are reasonably known.

Reachability analysis, based on the construction of the state space of the model, provides a complete knowledge of all its properties if the net is bounded (i.e., if the number of reachable states is finite). However, the exponencial temporal and spatial computational complexity originated from the *state explosion* reduces the applicability of this enumeration technique.

In order to avoid the state explosion, *reduction/transformation* and *structural* techniques have been developed. The first are based on the application of local rules for the simplification of nets, preserving some of the desirable properties. On the other hand, structural techniques allow to conclude about some properties of the model just from the net structure and using mathematical tools taken from *graph theory*, *linear algebra*, *convex geometry*, or *linear programming*.

Regarding *quantitative analysis* of Petri nets with timing interpretation, the most commonly used technique consists on the derivation of exact performance measures from the reachability graph of the model (if bounded) which is identified with a Markov chain, under certain assumptions on the stochastic specification. As in the case of qualitative reachability analysis, the explosion of the computational complexity is the main problem in the actual use of this technique for the performance evaluation of large models.

Alternative methods for the quantitative evaluation of Petri net models have been tried out. As in the case of queueing networks, *approximation techniques* and the computation of *bounds* constitute an option instead of exact analysis. The study of the second one has been our choice!

1.2.2 Some terminology

The purpose of this section is just to introduce some notations and terminology to be extensively used in the sequel. The reader is assumed to be familiar with basic Petri nets concepts.

1.2.2.1 Net structure

A Petri net is a 4-tuple $\mathcal{N} = \langle P, T, Pre, Post \rangle$, where

- P is the set of *places* ($|P| = n$),
- T is the set of *transitions* ($|T| = m$, $P \cap T = \emptyset$, $P \cup T \neq \emptyset$),
- Pre ($Post$) is the *pre- (post-) incidence function* representing the input (output) arcs, $Pre\colon P \times T \to \mathbb{N} = \{0, 1, 2, \ldots\}$ ($Post\colon P \times T \to \mathbb{N}$).

A Petri net can be seen as a *bipartite directed graph* in which places and transitions are the two kinds of nodes. Places are usually drawn as circles while transitions are depicted as bars or boxes.

Ordinary nets are Petri nets whose pre and post incidence functions take values in $\{0, 1\}$. The incidence function of a given arc in non-ordinary nets is called *weight* or *multiplicity.*

The *pre-* and *post-sets* of a transition $t \in T$ are defined respectively as ${}^\bullet t = \{p | Pre(p, t) > 0\}$ and $t^\bullet = \{p | Post(p, t) > 0\}$. The *pre-* and *post-sets* of a place $p \in P$ are defined respectively as ${}^\bullet p = \{t | Post(p, t) > 0\}$ and $p^\bullet = \{t | Pre(p, t) > 0\}$.

The *incidence matrix* of the net $C = [c_{ij}]$, $i = 1, \ldots, n$, $j = 1, \ldots, m$, is defined by $c_{ij} = Post(p_i, t_j) - Pre(p_i, t_j)$. Similarly the *pre- and post-incidence matrices* are defined as $PRE = [a_{ij}]$ and $POST = [b_{ij}]$, where $a_{ij} = Pre(p_i, t_j)$ and $b_{ij} = Post(p_i, t_j)$.

1.2.2.2 Token game

A function $M\colon P \to \mathbb{N}$ (usually represented in vector form) is called *marking*. A *marked Petri net* $\langle \mathcal{N}, M_0\rangle$ is a Petri net $\mathcal{N}$ with an *initial marking* M_0.

A transition $t \in T$ is *enabled* at marking M iff $\forall p \in P$: $M(p) \geq Pre(p,t)$. A transition t enabled at M can *fire* yielding a new marking M' (*reached* marking) defined by $M'(p) = M(p) - Pre(p,t) + Post(p,t)$ (it is denoted by $M[t\rangle M'$).

A sequence of transitions $\sigma = t_1 t_2 \ldots t_n$ is a *firing sequence* of $\langle \mathcal{N}, M_0\rangle$ iff there exists a sequence of markings such that $M_0[t_1\rangle\ M_1[t_2\rangle\ M_2 \ldots [t_n\rangle M_n$. In this case, marking M_n is said to be *reachable* from M_0 by firing σ, and this is denoted by $M_0[\sigma\rangle M_n$. Expresion $M[\sigma\rangle$ denotes a firable sequence σ from marking M.

The function $\vec{\sigma}\colon T \to \mathbb{N}$ is the *firing count vector* or Parikh vector [Par66] of the firable sequence σ, i.e., $\vec{\sigma}[t]$ represents the number of occurrences of $t \in T$ in σ. If $M_0[\sigma\rangle M$, then we can write in vector form $M = M_0 + C \cdot \vec{\sigma}$, which is referred to as the *linear state equation* of the net. A marking M is said to be *potentially reachable* iff $\exists \vec{X} \geq 0$ such that $M = M_0 + C \cdot \vec{X} \geq 0$.

1.2.2.3 Basic properties

The *reachability set* $R(\mathcal{N}, M_0)$ is the set of all markings reachable from the initial marking. Denoting by $PR(\mathcal{N}, M_0)$ the set of all potentially reachable markings we have the following relation: $R(\mathcal{N}, M_0) \subseteq PR(\mathcal{N}, M_0)$. $L(\mathcal{N}, M_0)$ is the set of all firing sequences and their suffixes in $\langle \mathcal{N}, M_0\rangle$: $L(\mathcal{N}, M_0) = \{\sigma \mid M[\sigma\rangle \text{ with } M \in R(\mathcal{N}, M_0)\}$.

A place $p \in P$ is said to be *k–bounded* iff $\forall M \in R(\mathcal{N}, M_0)$, $M(p) \leq k$. A marked net $\langle \mathcal{N}, M_0\rangle$ is said to be (marking) k–bounded iff each of its places is k–bounded. A net $\mathcal{N}$ is *structurally bounded* iff $\forall M_0$ the marked nets $\langle \mathcal{N}, M_0\rangle$ are k–bounded for some $k \in \mathbb{N}$.

Given an initial marking, an *implicit* place is one which never is the unique that restricts the firing of its output transitions. Let $\mathcal{N}$ be any net and $\mathcal{N}^p$ be the net resulting from adding an implicit place p to $\mathcal{N}$. Therefore, the firing sequences in $\langle \mathcal{N}, M_0\rangle$ and $\langle \mathcal{N}^p, M_0^p\rangle$ are identical.

A transition $t \in T$ is *live* in $\langle \mathcal{N}, M_0\rangle$ iff $\forall M \in R(\mathcal{N}, M_0)$: $\exists M' \in R(\mathcal{N}, M)$ such that M' enables t. The marked net $\langle \mathcal{N}, M_0\rangle$ is live iff all its transitions are live (i.e., liveness of the net guarantees the possibility of an infinite activity of all transitions). A net $\mathcal{N}$ is *structurally live* iff $\exists M_0$ such that the marked net $\langle \mathcal{N}, M_0 \rangle$ is live. The marked net $\langle \mathcal{N}, M_0 \rangle$ is *deadlock-free* iff $\forall M \in R(\mathcal{N}, M_0)$: $\exists t \in T$ such that M enables t. A marked net has a *total deadlock* iff it is not deadlock-free.

A *consistent component* (or *T-semiflow*) is a function (vector) $X\colon T \to \mathbb{N}$ such that $X \neq 0$ and $C \cdot X = 0$. A *conservative component* (or *P-semiflow*) is a function (vector) $Y\colon P \to \mathbb{N}$ such that $Y \neq 0$ and $Y^T \cdot C = 0$. The *support* of (T- and P-) semiflows is defined by $||X|| = \{t \in T | X(t) > 0\}$ and $||Y|| = \{p \in P | Y(p) > 0\}$. A (T- or P-) semiflow I has *minimal support* iff there exist no other semiflow I' such that $||I'|| \subset ||I||$. A (T- or P-) semiflow is *canonical* iff the greatest common divisor of its components is 1. A (T- or P-) semiflow is *elementary* iff it is canonical and has minimal support.

A net $\mathcal{N}$ is *consistent* iff there exists a T-semiflow $X \geq \mathbb{1}$. A net $\mathcal{N}$ is *conservative* iff there exists a P-semiflow $Y \geq \mathbb{1}$.

$M \in R(\mathcal{N}, M_0)$ is a *home state* iff $\forall M' \in R(\mathcal{N}, M_0) : M \in R(\mathcal{N}, M')$. $M \in R(\mathcal{N}, M_0)$ is a *transient state* iff it is not a home state. A marked net is *reversible* iff its initial marking is a home state.

1.2.3 On stochastic Petri nets

In the original definition, Petri nets did not include the notion of time, and tried to model only the logical behaviour of systems by describing the causal relations existing among events. This approach showed its power in the specification and analysis of concurrent systems in a non-interleaved way, independent of the concept of time. Nevertheless the introduction of timing specification is essential if we want to use this class of models for an evaluation of the performance of distributed systems [TPN85,PNPM87,PNPM89].

1.2.3.1 Timing and firing process

Since Petri nets are bipartite graphs, historically there have been two ways of introducing the concept of time in them, namely, associating a time interpretation (deterministic or stochastic) with either places [Sif78] or transitions [Ram74]. Since transitions represent activities that change the state (marking) of the net, it seems natural to associate a duration with these activities (transitions). The latter has been our choice. In other words, from a queueing theory perspective, the service stations are represented by timed transitions, and we denote by s_i the *average service time* of transition t_i.

In the case of timed transition models, two different firing rules have been defined:

1) "timed firing" of transitions in three phases which changes the firing rule of Petri nets introducing a timed phase in which the transition is "working" after having removed tokens from the input and before adding tokens to the output places, or a

2) "timed enabling" followed by an atomic firing which does not affect the usual Petri net firing rule.

These different timing interpretations have different implications on the resolution of *conflicts* [AMBB$^+$89]. On the one hand, using timed transition models with three phases firing we can define a policy for conflict resolution independent of the time specification but we cannot model *pre-emption*. On the other hand, using timed transition models with single phase firing we can model pre-emption but we cannot define conflict resolution policies independent of the timing specification (the conflicts are usually resolved in this case with *race policy*, i.e., the transition which samples the minimum service time is the one whose firing determines the change of marking).

In order to avoid the coupling between resolution of conflicts and duration of activities, we suppose that transitions in conflict are *immediate* (they fire in zero time). Decisions at these conflicts are taken according to *routing rates* associated with immediate transitions (*generalized stochastic Petri nets* [AMBC84,AMBCC87a]). In this way, preemption cannot be modelled. In other words, each subset of transitions $\{t_1, \ldots, t_k\} \subset T$ that are in conflict in one or several reachable markings are considered immediate, and the constants $r_1, \ldots, r_k \in \mathbb{N}^+$ are explicitly defined in the net interpretation in such a way that when $t_1, \ldots, t_k$ are enabled, transition t_i ($i = 1, \ldots, k$) fires with probability (or with long run rate, in the case of deterministic conflicts resolution policy) $r_i/(\sum_{j=1}^{k} r_j)$. Note that the routing rates are assumed to be strictly positive, i.e., all possible outcomes of any conflict have a non-null probability of firing. This fact guarantees a *fair* behaviour for the non-autonomous Petri nets that we consider (a marked net is said to be fair iff all transitions that are simultaneously enabled infinitely many times will fire infinitely often).

In summary, we model service stations by means of (deterministic or stochastic) timed transitions, routing by means of immediate transitions in conflict, and both kinds of transitions, timed and immediate, can be used as fork (split) nodes and join (synchronization) nodes.

1.2.3.2 Single versus multiple server semantics

Another possible source of confusion in the definition of the timed interpretation of a Petri net model is the concept of *degree of enabling* of a transition (or re-entrance). In the case of timing associated with places, it seems quite natural to define an unavailability time which is independent of the total number of tokens already present in the place, an this can be interpreted as an *infinite-server* policy from the point of view of queueing theory. In the case of time associated with transitions, it is less obvious a-priori whether a transition enabled k times in a marking should work at conditional speed 1 or k times that it would work in the case it was enabled only once. In the case of stochastic Petri nets with exponentially distributed service times associated with transitions, the usual implicit hypothesis is to have *single-server* semantics (see, e.g., [Mol82,FN85a]), and the case of *multiple-server* is handled as a case of service rate dependent on the marking; this trick cannot work in the case of more general probability distributions. This is the reason why people working with deterministic timed transitions Petri nets prefer an infinite-server semantics (see, e.g., [RP84,HV85,Zub85]). Of course an infinite-server transition can always be constrained to a "k–server" behaviour by adding one place that is both input and output (self-loop with multiplicity 1) for that transition and marking it with k tokens. Therefore, the infinite-server semantics appears to be the most general one, and for this reason it is adopted in this work.

The maximum number of servers working in parallel at a given transition will be characterized with the *enabling bound* concept.

Definition 1.2.1 (Enabling bound) *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked Petri net. The enabling bound of a given transition t of $\mathcal{N}$ is*

$$E(t) \overset{\text{def}}{=} \max\{\ k \mid \exists M \in R(\mathcal{N}, M_0) :\ M \geq kPRE[t]\ \}$$

Since in this work we are interested in the steady-state performance of a model, one can ask the question how many servers are available in transitions in steady-state condition. The answer is the definition of the *liveness bound* concept.

Definition 1.2.2 (Liveness bound) *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked Petri net. The liveness bound of a given transition t of $\mathcal{N}$ is:*

$$L(t) \overset{\text{def}}{=} \max\{\ k \mid \forall M' \in R(\mathcal{N}, M_0),\ \exists M \in R(\mathcal{N}, M') :\ M \geq kPRE[t]\ \}$$

The above definitions allow to generalize the classical concepts of enabling and liveness of a transition. In particular, a transition t is live if and only if $L(t) > 0$, i.e., if there is at least one working server associated with it in steady-state conditions. The following is also obvious from the definitions.

Property 1.2.1 *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked Petri net, then for all transition t of $\mathcal{N}$, $E(t) \geq L(t)$.*

A case of strict inequality in this property can be interpreted as a generalization of the concept of non-liveness: there exist transitions containing "potential servers" that are never used in the steady-state; these additional servers might only be used in a transient phase, so they "die" during the evolution of the model. See, as an example, the net in figure 1.6. For transition t_1 we have: $E(t_1) = 2 > L(t_1) = 1$.

Since for any reversible net (i.e., such that M_0 is a home state) the reachability graph (which is a directed labelled graph with the reachable markings as nodes) is strongly connected, the following can be stated:

Property 1.2.2 *Let $\langle \mathcal{N}, M_0 \rangle$ be a reversible marked Petri net, then for all transition t of $\mathcal{N}$, $E(t) = L(t)$.*

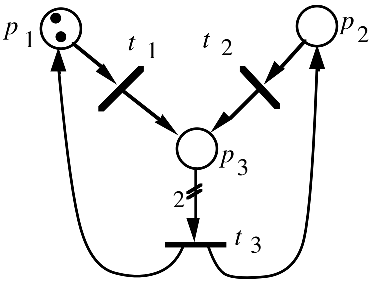


Figure 1.6: A net with enabling bound greater than liveness bound for transition t_1.

The definition of enabling bound refers to a behavioural property that depends on the reachability graph of a Petri net. Since we are looking for computational techniques at the structural level, we can also introduce the structural counterpart of the enabling bound concept. Structural net theory has been developed from two complementary points of view: graph theory [Bes87] and mathematical programming (or more specifically linear programming and linear algebra) [SC88]. Let us introduce our structural definition from the mathematical programming point of view; essentially in this case the reachability condition is substituted by the (in general) weaker (linear) constraint that markings satisfy the net state equation: $M = M_0 + C \cdot \vec{\sigma}$, with $M, \vec{\sigma} \geq 0$.

Definition 1.2.3 (Structural enabling bound) *Let $\mathcal{N}$ be a Petri net. The structural enabling bound of a given transition t of $\mathcal{N}$ is*

$$SE(t) \overset{\text{def}}{=} \begin{array}{ll} \textit{maximize} & k \\ \textit{subject to} & M = M_0 + C \cdot \vec{\sigma} \geq k \; PRE[t] \\ & \vec{\sigma} \geq 0 \end{array} \qquad \text{(LPP1)}$$

Note that the definition of structural enabling bound reduces to the formulation of a linear programming problem [Mur83].

Now let us remark the relation between behavioural and structural enabling bound concepts that follows from the implication "$M \in R(\mathcal{N}, M_0) \;\Rightarrow\; M = M_0 + C \cdot \vec{\sigma} \wedge \vec{\sigma} \geq 0$".

Property 1.2.3 *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked Petri net, then for all transition t of $\mathcal{N}$, $SE(t) \geq E(t)$.*

1.2.3.3 Ergodicity and measurability

In order to compute the steady-state performance of a system we have to assume that some kind of "average behaviour" can be estimated on the long run of the system we are studying. The usual assumption in this case is that the system model must be *ergodic* [Ros83], meaning that at the limit when the observation period tends to infinity, the estimates of average values tend (almost surely) to the theoretical expected values of the (usually unknown) probability distribution functions that characterize the performance indexes of interest.

This assumption is very strong and difficult to verify in general; moreover, it creates problems when we want to include the deterministic case as a special case of a stochastic model, since the existence of the theoretical limiting expected value can be hampered by the periodicity of the model. Thus we introduce the concept of *weak ergodicity* that allows the estimation of long run performance also in the case of deterministic models.

Definition 1.2.4 (Weak and strong ergodicities)

1. *A (not necessarily stochastic) process* Z_τ*, where* $\tau \geq 0$ *represents the time, is said to be weakly ergodic (or measurable in long run) iff the following limit exists:*

$$\lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau Z_u \, du < \infty \tag{1.7}$$

2. *A stochastic process* Z_τ*, where* $\tau \geq 0$ *represents the time, is said to be (strongly) ergodic iff the following condition holds:*

$$\lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau Z_u \, du = \lim_{\tau\to\infty} E[Z_\tau] < \infty \ \textit{(a.s.)} \tag{1.8}$$

For stochastic Petri nets, weak ergodicity of the marking and the firing processes can be defined in the following terms:

Definition 1.2.5 (Weak ergodicity of marking and firing) *The marking process* M_τ*, where* $\tau \geq 0$ *represents the time, of a stochastic marked net is weakly ergodic iff the following limit exists:*

$$\overline{M} \overset{\text{def}}{=} \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau M_u \, du < \vec{\infty} \tag{1.9}$$

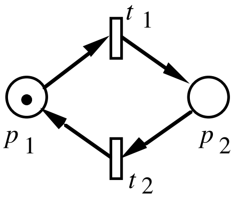


Figure 1.7: A trivial weakly but non-strongly marking ergodic deterministic net.

and $\overline{M}$ is called the limit average marking.

The firing process $\vec{\sigma}_\tau$, where $\tau \geq 0$ represents the time, of a stochastic marked net is weakly ergodic iff the following limit exists:

$$\overline{X} \overset{\text{def}}{=} \lim_{\tau \to \infty} \frac{\vec{\sigma}_\tau}{\tau} < \vec{\infty} \tag{1.10}$$

and $\overline{X}$ is the limit vector of transition throughputs (or limit firing flow vector).

The usual (i.e., strong) ergodicity concepts [FN85a] are defined in the obvious way taking into consideration definition 1.2.4.2.

Figure 1.7 shows a trivial example of a Petri net in which the marking process is weakly but not strongly ergodic when transitions t_1 and t_2 are associated with deterministic service times s_1 and s_2: indeed $E[M_\tau] = M_\tau$ is in this case a periodic function of time, so that $\lim_{\tau\to\infty} E[M_\tau]$ does not exist even if $\overline{M} = (s_2/(s_1+s_2), s_1/(s_1+s_2))^T$ in definition 1.2.5.

Ergodicity of the marking and of the firing processes are, in general, unrelated properties. Let us consider an $M|M|1$ queue modelled by means of the Petri net depicted in figure 1.8. If time-dependent rates $\lambda\tau$ and $\mu\tau$, with $\lambda < \mu$ are considered for arrival and service distributions (i.e., for service time of t_1 and t_3, respectively), the marking process is strongly ergodic while the firing process is not (even weakly) ergodic. In what follows, we do not consider any more time-dependent distributions. On the other hand, if the arrival and the service rates are λ and μ, respectively, with $\lambda > \mu$, then the firing process is strongly ergodic but the marking process is non (even weakly) ergodic, because the marking of the place p_2 tends to infinity, almost surely.

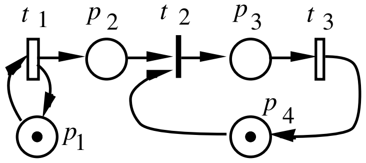


Figure 1.8: A net with home states but possibly non-ergodic marking process.

Other cases in which the firing process is weakly non-ergodic can be obtained when all the input places of a given transition are marking unbounded or when a transition has not any input place. In these cases, the liveness bound of that transition is infinite and, since infinite-server semantics is assumed, the throughput of the transition tends to infinity, unless the stochastic interpretation makes the limit marking bounded. Other "pathological behaviour" of the firing process leading to weak non-ergodicity can occur when there exits a circuit in the net including only immediate transitions. In this case, if no synchronization exists braking the firing speed of transitions in that circuit, their throughput tends to infinity. In what follows, we discard the previous undesirable cases.

For bounded nets, ergodicity of the firing process does not imply marking ergodicity. This can be the case if after an initial transient phase, the model can reach different closed subsets of the state space. Even in those cases in which there does not exist a "true" mean marking (i.e., the limit marking for $\tau \to \infty$ is not unique), it makes sense to compute upper and lower bounds on transition throughputs.

Related to marking ergodicity, if a bounded Petri net has a home state then its associated state space is finite and has a unique closed subset of markings. Therefore, the next result follows:

Theorem 1.2.1 *If a bounded marked net has a home state then its marking process is weakly ergodic.*

The above result provides an interesting example of possible interleaving between qualitative (home state concept) and quantitative (ergodicity concept) analysis for stochastic Petri nets.

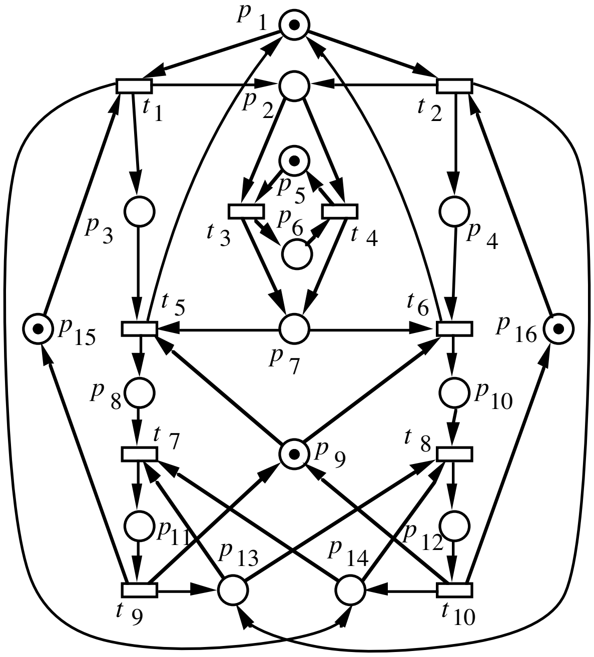


Figure 1.9: A live and bounded net without home states.

Markovian nets [FN85a] are stochastic Petri nets such that their related marking processes are Markov processes [Rev84]. Markovian Petri nets are obtained, e.g., using exponential distributions for transition service times. For Markovian bounded nets with home state, even strong marking ergodicity is assured:

Theorem 1.2.2 *If a Markovian bounded marked net has a home state then its marking process is strongly ergodic.*

Proof. The set of home states of the net is the unique recurrent class of the underlying Markov process. If the net is bounded, this class is positive recurrent and the marking process is strongly ergodic. ∎

The conditions of this theorem cannot be relaxed. An unbounded net can have home states but non-ergodic marking process if the mean marking of a place tends to infinity. On the other hand, nets can have bounded marking mean values and be non-ergodic because of the

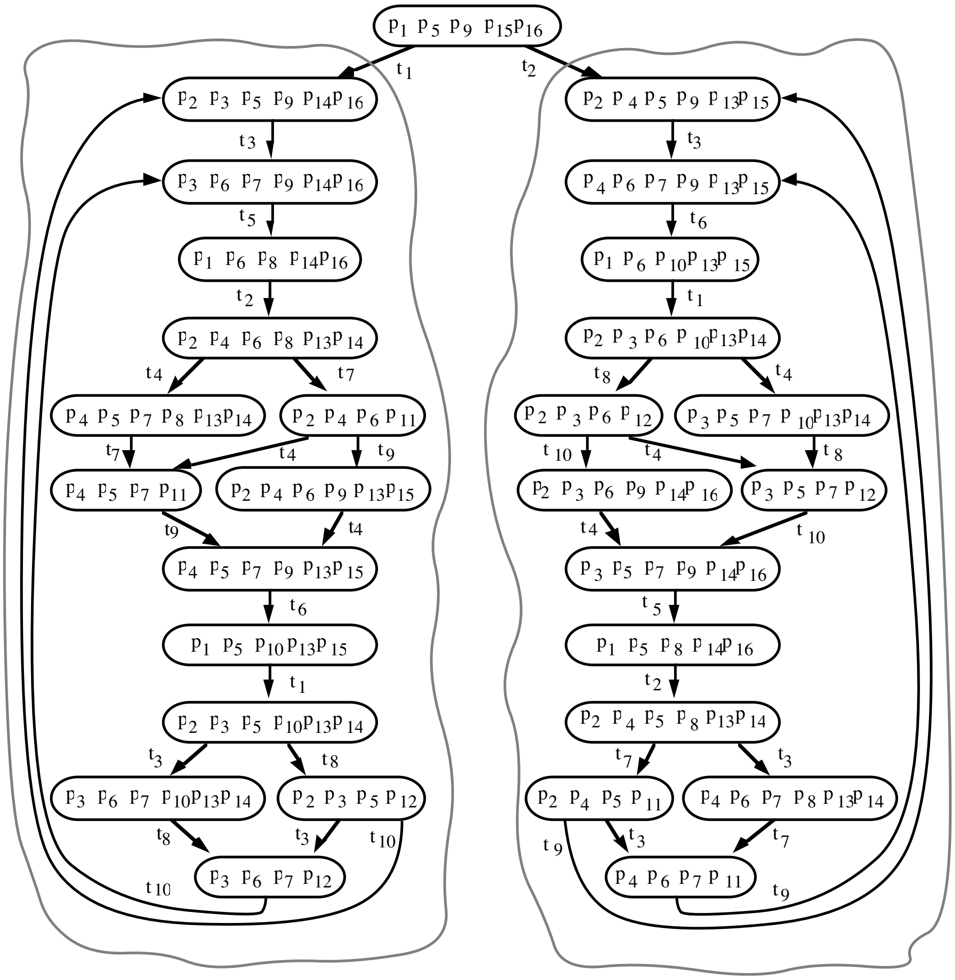

$p_1\ p_5\ p_9\ p_{15} p_{16}$
t_1
t_2
$p_2\ p_3\ p_5\ p_9\ p_{14} p_{16}$
t_3
$p_3\ p_6\ p_7\ p_9\ p_{14} p_{16}$
t_5
$p_1\ p_6\ p_8\ p_{14} p_{16}$
t_2
$p_2\ p_4\ p_6\ p_8\ p_{13} p_{14}$
t_4
t_7
$p_4\ p_5\ p_7\ p_8\ p_{13} p_{14}$
$p_2\ p_4\ p_6\ p_{11}$
t_7
t_4
t_9
$p_4\ p_5\ p_7\ p_{11}$
$p_2\ p_4\ p_6\ p_9\ p_{13} p_{15}$
t_9
t_4
$p_4\ p_5\ p_7\ p_9\ p_{13} p_{15}$
t_6
$p_1\ p_5\ p_{10} p_{13} p_{15}$
t_1
$p_2\ p_3\ p_5\ p_{10} p_{13} p_{14}$
t_3
t_8
$p_3\ p_6\ p_7\ p_{10} p_{13} p_{14}$
$p_2\ p_3\ p_5\ p_{12}$
t_8
t_3
t_{10}
t_{10}
$p_3\ p_6\ p_7\ p_{12}$
$p_2\ p_4\ p_5\ p_9\ p_{13} p_{15}$
t_3
$p_4\ p_6\ p_7\ p_9\ p_{13} p_{15}$
t_6
$p_1\ p_6\ p_{10} p_{13} p_{15}$
t_1
$p_2\ p_3\ p_6\ p_{10} p_{13} p_{14}$
t_8
t_4
$p_2\ p_3\ p_6\ p_{12}$
$p_3\ p_5\ p_7\ p_{10} p_{13} p_{14}$
t_{10}
t_4
t_8
$p_2\ p_3\ p_6\ p_9\ p_{14} p_{16}$
$p_3\ p_5\ p_7\ p_{12}$
t_4
t_{10}
$p_3\ p_5\ p_7\ p_9\ p_{14} p_{16}$
t_5
$p_1\ p_5\ p_8\ p_{14} p_{16}$
t_2
$p_2\ p_4\ p_5\ p_8\ p_{13} p_{14}$
t_7
t_3
$p_2\ p_4\ p_5\ p_{11}$
$p_4\ p_6\ p_7\ p_8\ p_{13} p_{14}$
t_9
t_3
t_7
$p_4\ p_6\ p_7\ p_{11}$
t_9

Figure 1.10: Reachability graph of the net in figure 1.9

presence of more than one closed subset in the state space. This is the case, for instance, for the net in figure 1.9 (taken from [BV84]). It is live and bounded and has two different closed subsets in its state space (depicted in figure 1.10). However, marking ergodicity does not imply the existence of a home state: for the net in figure 1.9 (which has not home state) an exponential distribution timing can be associated with transitions (e.g., taking the same value for the rates of all transitions) such that the related marking process is ergodic anyway.

Let us now give a structural necessary condition for the ergodicity of the marking process of a live Markovian Petri net.

Theorem 1.2.3 *Live and marking ergodic Markovian Petri nets are consistent.*

Proof. If a Markovian net is marking ergodic then, in particular, $\sup_\tau E[M_\tau] < \infty$, thus:

$$\limsup_{\tau\to\infty} \frac{E[M_\tau]}{\tau} = 0 \tag{1.11}$$

For all Markovian nets:

$$\limsup_{\tau\to\infty} \frac{E[\vec{\sigma}_\tau]}{\tau} < \infty \tag{1.12}$$

Now, from the linear state equation of the net:

$$\limsup_{\tau\to\infty} \frac{E[M_\tau]}{\tau} = \limsup_{\tau\to\infty} \frac{M_0 + E[C \cdot \vec{\sigma}_\tau]}{\tau} = \limsup_{\tau\to\infty} \frac{C \cdot E[\vec{\sigma}_\tau]}{\tau} \tag{1.13}$$

Then, from (1.11), (1.12), and (1.13):

$$C \cdot X = 0, \text{ where } X = \limsup_{t\to\infty} \frac{E[\vec{\sigma}_t]}{t} \tag{1.14}$$

Finally, the liveness of the net assures that vector X is such that $X \geq \mathbb{1}$. Therefore, the net is consistent. ■

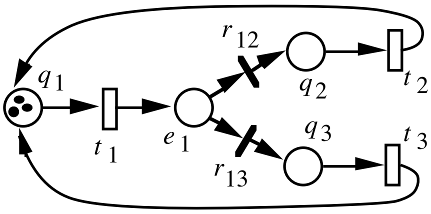


Figure 1.11: An example of stochastic Petri net representing a network of delay stations.

1.3 Mapping between monoclass synchronized queueing networks and stochastic Petri nets

In [VZL87] a comparison has been proposed between synchronized queueing networks and stochastic Petri nets, showing that *the two formalisms are roughly equivalent from a modelling point of view.*

Here we show how the different queueing network models with synchronizations presented in section 1.1 can be uniformely represented with a Petri net formalism.

An infinite-server queue [Kle75] (i.e., a pure delay node) can be represented by a Petri net containing one place to model the number of customers in the system and a timed transition connected with the place through an input arc to model departures. A queueing network containing only pure delay nodes can be modelled, as depicted in the example of figure 1.11, by a stochastic Petri net. Persistent timed transitions represent service times of the nodes, while conflicting immediate transitions model the routing of customers moving from one node to the other.

A monoclass single-server station [Kle75] can be modelled by a subnet of the type depicted in figure 1.12. Monoclass queueing networks containing both delay and finite-server nodes are thus naturally modelled by stochastic Petri nets of the type depicted in the example of figure 1.13 (t_1 is a delay, while t_2 and t_3 are single-server stations). Also in this more general context conflicting immediate transitions model the routing of customers among the stations, while persistent timed

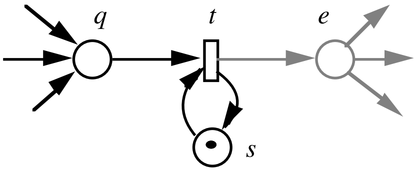


Figure 1.12: A Petri net representation of a monoclass single-server queue.

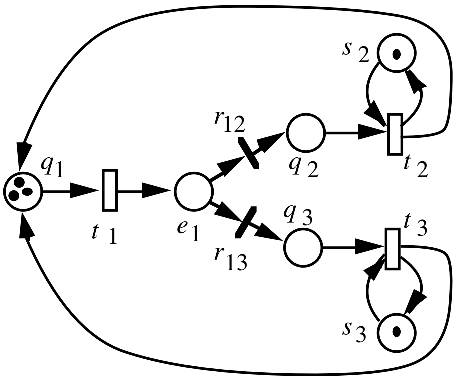


Figure 1.13: A Petri net representation of a queueing network.

transitions model the service times.

On the other hand, stochastic nets can assume forms much more complex than the one illustrated in the example of figure 1.13. Figure 1.14 illustrates a more general stochastic Petri net that cannot be mapped onto a product form queueing network. In fact, this net can be mapped on an extended queueing network [SMK82], in which such constructs as fork, join, and passive resources are used to map the effect of the pairs of transitions t_2–t_7 and t_9–t_{10}, respectively. These examples show how, using a Petri net formalism, extensions of product form queueing networks are represented with an analogous level of structural complexity of BCMP networks.

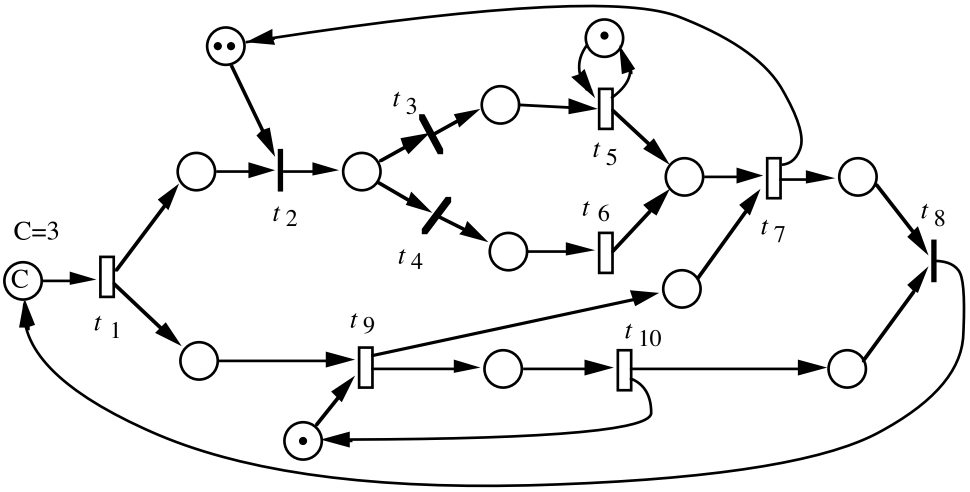
In section 1.1, extended queueing network models were presented for the modelling of a multiprogramming memory limited system (figure 1.2) and a fork/join multitasking process (figure 1.3). The corresponding Petri net models are depicted in figures 1.15 and 1.16, respectively.

The reader is noticed that "unclever" use of synchronizations in queueing networks can lead to pathological cases as unbounded number of customers or total deadlock (see figure 1.17), that need to be carefully studied.

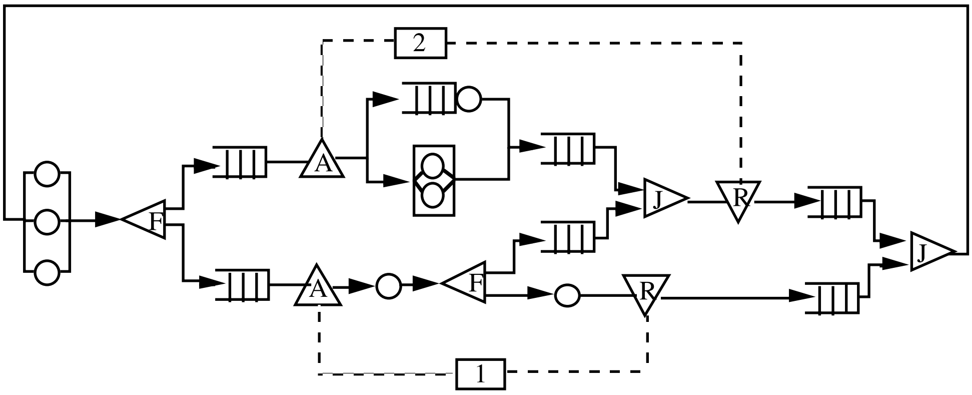
Finally, let us remark that stochastic Petri nets with weighted arcs (i.e., non-ordinary nets) can be used for the modelling of *bulk arrivals* and *bulk services* [Kle75], with *deterministic size of batches* (given by the weights of arcs). As an example, transition t_3 of Petri net in figure 1.6 is a bulk service system which accepts a batch of exactly two tokens (customers) from the place p_3, and serves them collectively.

1.4 Analytical techniques for synchronized queueing networks

One of the main problems in the actual use of timed and stochastic Petri net models for the performance evaluation of large systems is the explosion of the computational complexity of the analysis algorithms. In general, *exact performance results* are obtained from the numerical solution of a continuous time Markov chain [BT81,Mol81,FN85b]. This exact computation is only possible for bounded nets (finite state space),



(a) Stochastic Petri net representation.



(b) Extended queueing network representation.

Figure 1.14: A more general stochastic net and the corresponding synchronized queueing network.

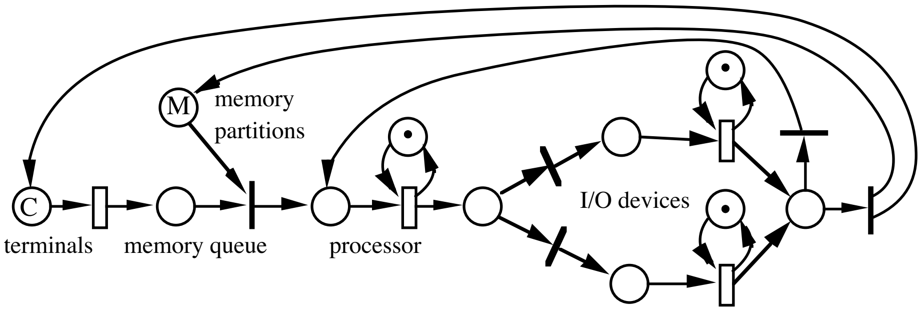


Figure 1.15: Petri net model of a multiprogramming memory limited system.

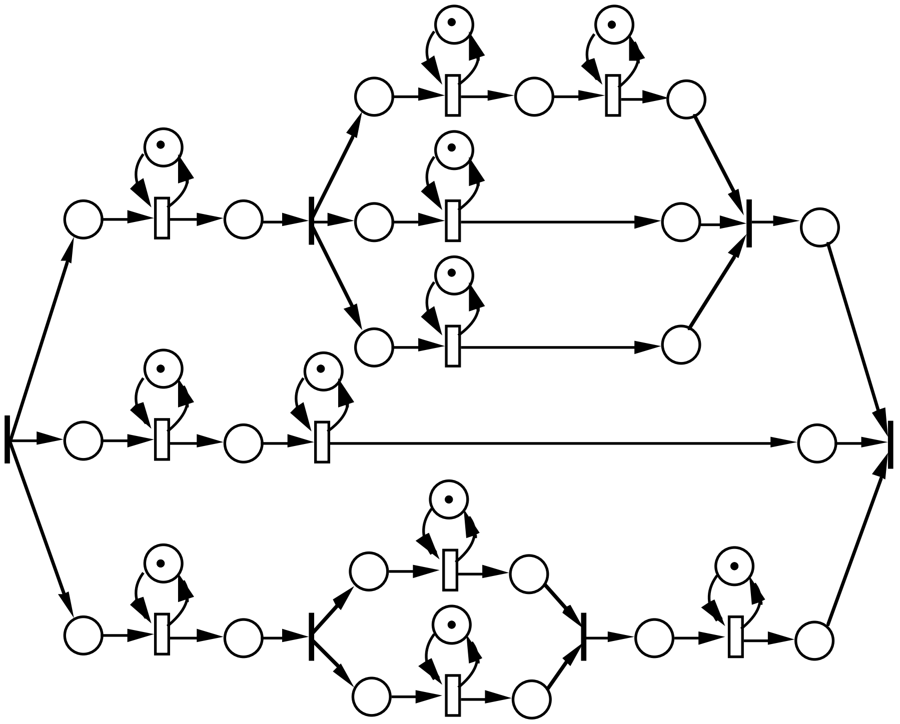
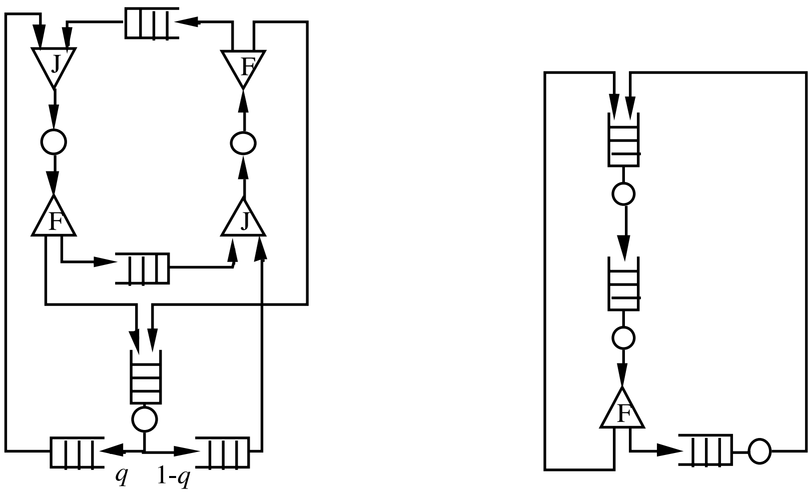


Figure 1.16: Petri net model of a fork/join multitasking process.



(a) A total deadlock will be reached sooner or later, even for q =1/2.

(b) Any infinite behaviour will lead to an infinite number of customers.

Figure 1.17: Pathological cases of synchronized queueing networks.

and under exponential assumption for the service time of transitions. And the worst of it is that the dimension of the state space of the embedded Markov chain grows exponentially with the net size.

The same problem arose in the framework of queueing networks before the work of J. Jackson, and it was solved by means of the introduction of *product form* equations [Jac63,GN67,BCMP75], and *efficient algorithms* for their solution [Buz73,RK75,RL80,BB80]. Unfortunately, the generalization of these results to more complex stochastic models with synchronization features seems to be very difficult, and a very few number of results have been already published.

Related with open networks, a *matrix product form solution* is known only for stochastic Petri nets with at most one place unbounded [FN86]. In [FN89a], G. Florin and S. Natkin presented the first general product form expresion in matrix form for closed (i.e., bounded) ordinary stochastic Petri nets with strongly connected reachability graph. The great difference between scalar (Gordon-Newell result for closed queueing networks) and matrix product forms appears in numerical computation. Solving synchronized queueing networks implies much more complex algorithms than classical ones. The problem of computing the *normalization constant* in the scalar product form solution is replaced by the computation of a constant vector obtained solving a system of linear equations, which is *ill-conditioned*. This is the reason why the paper of Florin and Natkin can be considered mainly of theoretical significance. Other works dealing with this problem [AMBCD86,LR87] consider only very restrictive subclasses of Petri nets. Therefore, efficient computational methods are still needed.

Approximation techniques have been developed in the framework of non-product form queueing networks for overcoming the practical limitations of exact solutions. The *"flow equivalent" server decomposition method* is probably the most used in practice [Lav89]. In this method, a subnetwork is replaced by a server with exponentially distributed service times and queue length dependent service rates. The rates are obtained by solving the throughput of the isolated subnetwork once for each possible value of number of customers in the subnetwork. The aggregated system consisting of this flow equivalent server and the rest of the original network is then solved.

Two different theoretical justifications for the fitness of the flow equivalent server method can be given. The first is that it yields exact results for single chain product form networks [CHW75]. This result is called *Norton's theorem for product form queueing networks* due to its analogy with Norton's theorem for electrical circuits (in which a subsystem is replaced by a current source and parallel resistance that are equivalent to the original subsystem in terms of their effect on the rest of the system). This exact result for product form queueing networks suggests that the flow equivalent server method may yield fairly accurate approximations for networks that are "almost product form".

The second justification for the use of this method was performed by P. Courtois [Cou77] within the framework of the computation of the steady-state solution of large Markov chains in which states are aggregated into macrostates to reduce the computational complexity of the solution (*nearly or completely decomposable systems*).

Practical experience shows that using decomposition techniques for the solution of non-product form networks made up of subsystems that, taken in isolation, satisfy the product form conditions often yields quite acceptable results [AMBC86].

A complementary approach to the approximation techniques for the analysis of queueing networks is the computation of *bounds*. Performance bounds are useful in the preliminary phases of the design of a system, in which many parameters are not known accurately. Several alternatives for those parameters should be quickly evaluated, and rejected those that are clearly bad. Exact (and even approximate) solutions would be computationally very expensive. Bounds become useful in these instances since they usually require much less computation effort.

A large number of bounding techniques have been proposed for the performance measures of queueing networks. The first family is that of *asymptotic bound analysis* [Kle76,DB78]. Asymptotic bounds are obtained by considering two extreme situations: (1) no queueing takes place at any node, and (2) at least one station is saturated. These bounds do not require the product form property to hold and their computation is very fast, but they are not accurate in general. The rest of bounds that have been introduced are tighter but do require the product form assumption. This is the case of *balanced job bounds* [ZSEG82,Kri84], which are based on the *mean value theorem* [RL80]. Finally, several schemes for the construction of *hierarchies of bounds* have been developed that guarantee any level of accuracy (including the exact solution), by investing the necessary computational effort: *performance bound hierarchies* [ES83,ES86], *succesively improving bounds* [Sri87], *generalized quick bounds* [Sur84]. All these techniques are derived from mean value theorem, thus they are valid only for product form networks.

1.5 An overview of performance bounds for stochastic Petri nets

Many works exist concerning the performance evaluation in the case of deterministically timed nets, mainly for strongly connected marked graphs [Ram74,Sif78,RH80,Mag84,Mur85]. We assume all these results, which can be identified as a particular case (in fact an "extreme" case) of the general stochastic timing, and we reformulate them in a general form which allows efficient computation methods. Extensions to non-ordinary nets have been presented in the case of deterministic timing [Hil88]. Our work considers also these nets in a unified formulation.

In the framework of stochastic Petri nets, only a few works exist related with the computation of performance bounds [Mol85,BG85,IA89], and all of them are valid just for restrictive assumptions on the nets.

M. Molloy [Mol85] noted that the average token flows in an ordinary Markovian network at steady-state are conserved. Therefore, a series of *flow balance equations* can be written. Token flows are conserved in places so the sum of all flows into a place equals the sum of all flows out of the place. On the other hand, all token flows on the input and output arcs of a transition are equal. These equations determine the average token flows in the cycles of the net to within a constant. This constant cannot be determined without Markovian analysis at the reachability graph level. However, limit flows when the number of tokens tends to infinity can be computed. In order to do that, bottleneck transitions must be first located. Then, the actual flow through a bottleneck transition is (under saturation conditions) equal to its potential firing rate.

It is well-known that the conservation of flows presented by M. Molloy is not only valid for Markovian nets. In fact, some of most important laws of queueing theory hold under very general assumptions. These general situations are considered in our work, and some fundamental laws taken from queueing theory (such as Little's formula) are applied to stochastic Petri net models.

S. Bruell and S. Ghanta [BG85] developed algorithms for computing upper and lower bounds for the throughput of a restricted subclass of generalized stochastic Petri nets (with immediate and exponentially timed transitions). The considered nets include *control tokens* to model a physical restriction, such as semaphores, which is not a design parameter. The rest of tokens of such nets, grouped in *classes*, correspond to the notion of a job or customer in a monoclass queueing network, and its number is treated as a parameter of the net. The upper and lower bounds on throughput are computed hierarchically estimating maximum and minimum time of the path followed by each class of jobs.

Unfortunately, the above cited article [BG85], which is considered by the authors as a "preliminary work", suffers from an excessive informal style that makes confusing both the characterization of the considered net subclasses and the computation algorithms. However, in those cases in which we have been able to applied the techniques presented in [BG85], the obtained results agree with the ones that we get using the algorithms we present in this work.

In the paper of S. Islam and H. Ammar [IA89], methods to compute upper and lower bounds for the steady-state token probabilities of a subclass of generalized stochastic Petri nets are presented. The considered nets are obliged to admit a *time scale decomposition*. This means that the transitions of the net are supposed to be divided into two classes: slow and fast transitions, with several orders of magnitude of difference in the duration of activities. Moreover, the subnets obtained after removing all slow transitions with their input and output arcs must be conservative and admit a reversible initial marking. The computation is based on *near-completely decomposability* of Markov chains.

Our approach is different, and complementary, from the one presented in [IA89]. One objective of this text is to present algorithms for the computation of bounds for stochastic Petri nets for arbitrary mean values of service times of transitions and, moreover, for arbitrary distribution functions of the timing. This main objective is attacked in an unified framework considering both qualitative and quantitative properties of stochastic Petri nets, and laying special emphasis on structure theory of nets. The computation of both the upper and lower bounds is based on an efficient calculation of the *visit ratios for transitions*, a concept taken from classical queueing theory. These visit ratios, together with the average service time of transitions, the net structure, and the initial marking, are used for the derivation of proper linear programming problems whose optimum solutions are the desired bounds. In the next chapter, we focus on the computability of visit ratios from different net parameters, such as the net structure and the stochastic interpretation. Net subclasses for which this computation is possible in polynomial time are especially considered (their characterization, inclusion relations. . .), making emphasis on those qualitative properties that are interesting from a performance point of view.

Chapter 2

Petri net subclasses and bases of qualitative theory

In the previous chapter, we recalled that the exact computation of performance measures for closed product form queueing networks can be computed from the knowledge of the average service demands of customers from stations. These parameters have been defined for each service center as the product of visit ratio by average service time. While service times are supposed to be always explicitly given as a part of the model specification, visit ratios for stations are derived, in the case of queueing networks, from the routing probabilitites among stations.

Concerning stochastic Petri nets, we assume also that the average service times of transitions are known. Then, in order to compute the average service demand of tokens from transitions, it is necessary to compute just the visit ratios or relative throughputs of transitions.

Unfortunately, as we remarked in previous chapter, the introduction of synchronization schemes can lead to the "pathological" behaviour of models reaching a total deadlock, thus with null visit ratios for all transitions, in the limit. In other words, for these models it makes no sense to speak about steady-state behaviour. Therefore, in the rest of this text we consider only deadlock-free Petri nets. Moreover, in most subclasses in which we are interested, deadlock-freeness implies liveness of the net, in other words, the existence of an infinite activity of all the transitions is assured.

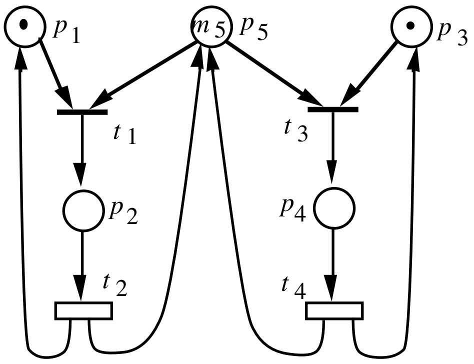


Figure 2.1: A net whose visit ratios depend on the structure, on the routing at conflicts, on the initial marking, and on the service times.

The counterpart of routing probabilities of queueing networks in stochastic Petri nets consists both on the *net structure* $\mathcal{N}$ (as a particular case of deterministic routing) and the *routing rates at conflicts* (let us denote as $\mathcal{R}$). Unfortunately, in the general Petri net case it is not possible to derive the visit ratios only from $\mathcal{N}$ and $\mathcal{R}$. Nets can be constructed such that the visit ratios for transitions do depend on the net structure, on the routing rates at conflicts, but also on the initial marking (distribution of customers), and on the average service time of transitions:

$$\vec{v}^{(1)} = \varphi(\mathcal{N}, \mathcal{R}, M_0, \vec{s}) \tag{2.1}$$

where $\vec{v}^{(1)}$ and $\vec{s}$ denote the vectors with components $v_i^{(1)}, i = 1, \ldots, m$, and $s_i, i = 1, \ldots, m$, respectively.

As an example, let us consider the net depicted in figure 2.1. Transitions t_1 and t_3 are *immediate* (i.e., they fire in zero time). The constants $r_1, r_3 \in \mathbb{N}^+$ define the conflict resolution policy, i.e., when t_1 and t_3 are simultaneously enabled, t_1 fires with relative frequency $r_1/(r_1 + r_3)$ and t_3 with $r_3/(r_1 + r_3)$. Let s_2 and s_4 be the average service times of t_2 and t_4, respectively. If $m_5 = 1$ (initial marking of p_5) then p_1 and p_3 are implicit (see section 1.2.2), hence they can be deleted (without affecting the behaviour!). Thus a closed queueing network topology is derived. A product form queueing network can be obtained and the visit ratios, normalized for transition t_1 can be computed: $\vec{v}^{(1)} = (1, 1, r_3/r_1, r_3/r_1)^T$. If $m_5 = 2$ (different initial marking for p_5) then p_5 is implicit, hence it can be deleted; two isolated closed tandem queueing networks are obtained and $\vec{v}^{(1)'} = (1, 1, s_2/s_4, s_2/s_4)^T$. Obviously $\vec{v}^{(1)} \neq \vec{v}^{(1)'}$, in general.

In this chapter we classify Petri nets attending to the computability of visit ratios as functions of the net structure (or deterministic routing), the routing rates at conflicts, the initial marking, and the average service time of transitions. This particular classification criterion is quite different from the usually applied in the theory of Petri nets. However, some of the main classical net subclasses (*marked graphs*, *state machines*, *free choice nets*...) will be recognized attending to the dependence of visit ratios on the different parameters defining the model. Some properties for the introduced net subclasses, interesting from both the *qualitative* and the *quantitative* points of view are recalled or derived here, like: the complexity of the characterization of different subclasses; characterizations of structural liveness, liveness, boundedness...; liveness monotonicity, reversibility, home state existence; computation of the visit ratios and its complexity, weak and strong ergodicities...

Firstly, in section 2.1, we consider those nets whose associated vector of visit ratios for transitions can be computed from the net structure and the routing rates at conflicts. Since the existence of strictly positive visit ratios requires liveness of the net and we are looking for an structural computation, we restrict ourselves to *structurally live* Petri nets. On the other hand, unless otherwise explicitly stated, we consider *structurally bounded* nets. Under these restrictions, we characterize the class of nets for which the vector of visit ratios can be computed *without any behavioural analysis* (i.e., from structure and routing rates at conflicts) in terms of a *rank condition* over the incidence matrix of the net. These nets are defined as having *freely related T-semiflows* and denoted, for short, as *FRT-nets*. This means that they can have several independent T-semiflows but the vector of visit ratios, which is always a linear combination of the minimal T-semiflows, is computed as an "average T-semiflow" from local free choices among transitions, governed by the routing rates.

A particular class of FRT-nets is that of *mono-T-semiflow nets*, which have a unique minimal T-semiflow. They include *structurally* *decision-free nets* and, in particular, the well-known net subclass of *marked graphs.*

FRT-nets include also *free choice nets.* This last class includes marked graphs, thus intersecting with mono-T-semiflow nets. *State machines* are also included in the class of free choice nets and constitute the Petri net counterpart of monoclass queueing networks.

Other well-known subclass of marked FRT-nets is that of *deterministic systems of sequential processes*, that is, 1–bounded state machines communicating through private buffers non-disturbing local decisions at state machines. In fact, *FRT-nets communicating through private buffers non-disturbing local decisions* are also FRT-nets. In this sense, FRT-nets can be recursively defined.

Finally, in section 2.2, net subclasses whose vector of visit ratios can be computed from the structure, routing rates, and initial marking, but independently of the service times, are considered. *Persistent nets* and *behaviourally extended free choice nets* ("réseaux á choix non imposé", in French) are included in that paragraph. These nets are behaviourally defined. Therefore, after the analysis of the reachability graph necessary for their characterization, the vector of visit ratios for transitions can be derived, and it is independent of the transitions service times.

2.1 FRT-nets and subclasses

In this section we define and give some interesting properties for the class of structurally live and structurally bounded nets whose vector of visit ratios for transitions can be computed just from the net structure and the routing rates at conflicts. Nets belonging to this subclass, presented in section 2.1.1, are said to have *freely related T-semiflows* and called *FRT-nets.*

Later, in section 2.1.2, *mono-T-semiflow nets* are shown to be a subclass of FRT-nets with the particular property of having the vector of visit ratios computable just from the net structure. *Structurally decision-free nets* are mono-T-semiflow nets, and *marked graphs*, a well-known Petri net class, is identified as a subclass of structurally decision-free nets whose vector of visit ratios is always the unity vector.

Free choice nets are also considered, in section 2.1.3, and identified

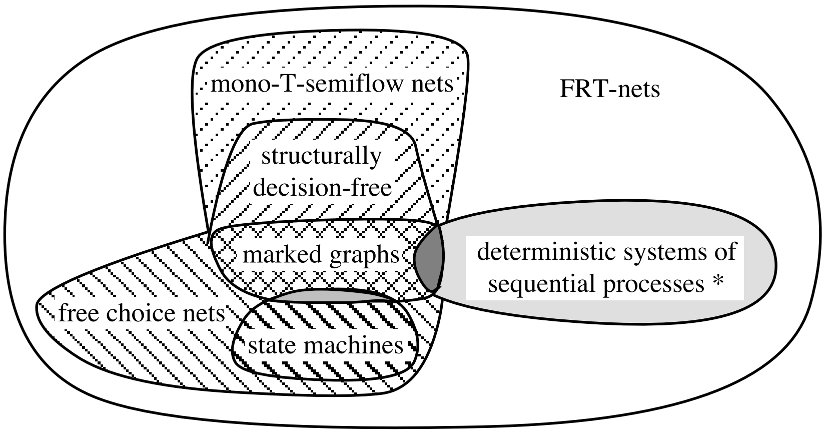

mono-T-semiflow nets
FRT-nets
structurally decision-free
marked graphs
deterministic systems of sequential processes *
free choice nets
state machines

Figure 2.2: Inclusion relations among FRT-net subclasses (* these are marked nets).

as a subclass of FRT-nets with some interesting additional properties.

Finally, in section 2.1.4, a modular technique of composition of FRT-nets for obtaining new FRT-nets is presented by means of the communication with *private buffers*. *Deterministic systems of sequential processes* constitute the particular case in which the communicated subnets are 1–bounded state machines.

Inclusion relations among above mentioned subclasses are depicted in figure 2.2.

2.1.1 FRT-nets

In this section, we consider the most general class of structurally live and structurally bounded nets whose vector of visit ratios can be computed from the structure and the routing rates at conflicts. Before giving a formal definition, let us remark that the vector of visit ratios for transitions of any net should verify the two following conditions:

- The vector of visit ratios $\vec{v}^{(j)}$ (normalized, for instance, for transition t_j) must be a non-negative right annuller of the incidence matrix (i.e., a T-semiflow):

$$C \cdot \vec{v}^{(j)} = 0 \tag{2.2}$$

- The visit ratios of two different transitions t_{i_1} and t_{i_2} in *structural free conflict* (i.e., having equal pre-incidence function) must be proportional to the corresponding routing rates r_{i_1} and r_{i_2} defining the conflict resolution: $r_{i_1} v_{i_2}^{(j)} = r_{i_2} v_{i_1}^{(j)}$. This condition can be also written in vector form as:

  $$R \cdot \vec{v}^{(j)} = 0 \tag{2.3}$$

  where R is a matrix with as many rows as pairs of transitions in structural free conflict.

The above remarked conditions together with the normalization $v_j^{(j)} = 1$ for a given transition t_j characterize a unique vector if and only if the number of independent rows of the matrix

$$\begin{pmatrix} C \\ R \end{pmatrix} \tag{2.4}$$

is $m - 1$, with $m = |T|$.

2.1.1.1 Definition

We introduce now the class of structurally live and structurally bounded nets verifying the previous rank condition. In order to do that, we define an equivalence relation on the set of T-semiflows of the net. After that, the class of FRT-nets will be defined as nets having only one equivalence class for this relation.

Definition 2.1.1 (Freely connected T-semiflows) *Let $\mathcal{N}$ be a Petri net and X_a, X_b two different T-semiflows of $\mathcal{N}$. X_a and X_b are said to be freely connected by places $P' \subset P$, denoted as $X_a \overset{P'}{\wedge} X_b$, iff $\exists t_a \in ||X_a||, t_b \in ||X_b||$ such that: $PRE[t_a] = PRE[t_b]$ and ${}^\bullet t_a = {}^\bullet t_b = P'$.*

Definition 2.1.2 (Freely related T-semiflows) *Let $\mathcal{N}$ be a Petri net and X_a, X_b two T-semiflows of $\mathcal{N}$. X_a and X_b are said to be freely related, denoted as $(X_a, X_b) \in FR$, iff one of the following conditions holds:*

1. $X_a = X_b$,

2. $\exists P' \subset P$ *such that* $X_a \overset{P'}{\wedge} X_b$, *or*

3. $\exists X_1, \ldots, X_k$ *T-semiflows of* $\mathcal{N}$ *and* $P_1, \ldots, P_{k+1} \subset P$, $k \geq 1$, *such that* $X_a \overset{P_1}{\wedge} X_1 \overset{P_2}{\wedge} \ldots \overset{P_k}{\wedge} X_k \overset{P_{k+1}}{\wedge} X_b$.

From the above definition the next property trivially follows:

Property 2.1.1 *FR is an equivalence relation on the set of T-semiflows of a net.*

The introduction of this equivalence relation on the set of T-semiflows induces a partition into equivalence classes. FRT-nets are defined as follows:

Definition 2.1.3 (FRT-nets) *We say that a Petri net $\mathcal{N}$ is a net with freely related T-semiflows (FRT-net, for short) iff the introduction of the freely relation on the set of its T-semiflows induces only one equivalence class.*

Note that FRT-nets are necessarily connected. Therefore, in what follows, unless otherwise explicitly stated, we consider only connected nets.

As an example, let us consider the net depicted in figure 2.3. It is a live and structurally bounded net. Its minimal T-semiflows are:

$$\begin{array}{rcl} X_1 & = & (1,0,0,0,0,0,1,0,0,0,1,0,1,0)^T \\ X_2 & = & (0,1,1,0,0,0,0,1,0,0,0,0,1,0)^T \\ X_3 & = & (0,0,0,1,1,0,0,0,1,0,0,0,0,1)^T \\ X_4 & = & (0,0,0,0,0,1,0,0,0,1,0,1,0,1)^T \end{array} \tag{2.5}$$

Then, the net is an FRT-net because:

$$X_1 \overset{\{p_1\}}{\wedge} X_2 \overset{\{p_2\}}{\wedge} X_3 \overset{\{p_3\}}{\wedge} X_4 \tag{2.6}$$

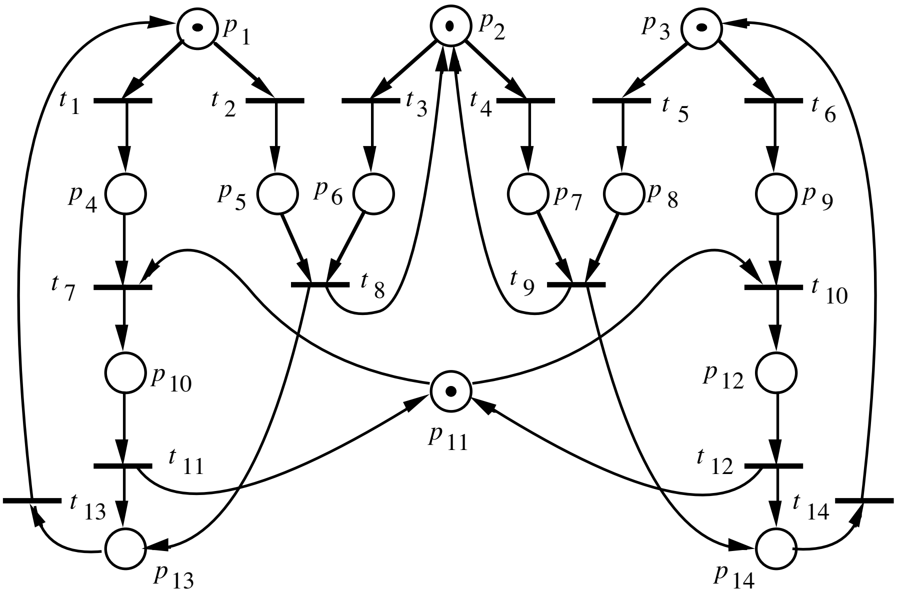


Figure 2.3: A live and structurally bounded FRT-net.

2.1.1.2 Algebraic characterization

From the definition of FRT-nets, it may appears that a direct checking of the pertenence of a given net to this net subclass is not a polynomial problem on the net size. This is because the number of T-semiflows of a net can grow exponentially with the number of places and transitions. However, if structural liveness and structural boundedness are assumed, a nice characterization of the FRT-nets subclass can be obtained and checked in polynomial time. Before the presentation of that result, we introduce a second equivalence relation, now on the set of transitions of the net.

Definition 2.1.4 (Equality conflict relation) [CCS90d] *Two transitions t_a and t_b are said to be in equality conflict relation, denoted by $(t_a, t_b) \in ECR$, iff $PRE[t_a] = PRE[t_b]$.*

Since the equality conflict relation is based on the equality of vectors, the next property follows:

Property 2.1.2 *ECR is an equivalence relation on the set of transitions.*

Each equivalence class will be called *equality conflict set*, and denoted as ECS. Let D be an ECS, the number $\delta_D = |D| - 1$ is called *number of non-redundant free conflicts of D.* The reason of the name lies on the fact that δ_D is exactly the number of independent relations among the throughput of transitions belonging to D that can be derived from the routing rates defining the resolution of the conflict. The *number of non-redundant free conflicts of a net*, denoted as δ, is the sum of all δ_D corresponding to the ECSs of the net: $\delta = \sum_{D \in T/ECR} \delta_D$.

Theorem 2.1.1 *Let $\mathcal{N}$ be a structurally live and structurally bounded net. Then $\mathcal{N}$ is an FRT-net if and only if $rank(C) = m - \delta - 1$, where C is the incidence matrix of $\mathcal{N}$, $m = |T|$, and δ is the number of non-redundant free conflicts of the net.*

Before giving the proof of the above theorem, let us state an important conclusion.

Corollary 2.1.1 *If $\mathcal{N}$ is structurally live and structurally bounded, deciding if $\mathcal{N}$ belongs to the class of FRT-nets is polynomial on the net size.*

Structural boundedness of a net can be always be checked in polynomial time (iff $\exists Y \geq \mathbb{1}$ such that $Y^T \cdot C \leq 0$ [Mur89]). Unfortunately, structural liveness of FRT-nets cannot be decided (so far) efficiently. Nevertheless, a *necessary condition* for a net to be structurally live structurally bounded and FRT-net can be checked in polynomial time, looking for the consistency, conservativeness, and rank condition over the incidence matrix, because structural liveness and structural boundedness implies consistency and conservativeness (see, e.g., [Sil85]).

In order to prove theorem 2.1.1 we previously present some lemmatas. The first one concerns a reduction of the non-determinism at equality conflicts, preserving the liveness property, by means of the merging of a special class of nets: *local schedulers*.

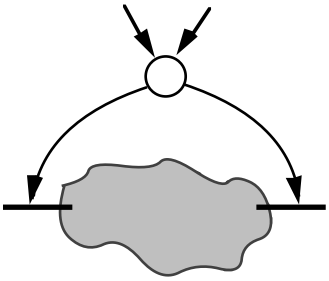


Figure 2.4: Introduction of a local scheduler at an equality conflict set.

Definition 2.1.5 (Local scheduler) [CCS90d] *Let $D = \{t_i \mid i = 1, \ldots, \delta_D + 1\}$ be an ECS of the net $\mathcal{N}$. A local scheduler for D is a net $\mathcal{LS}_D$ defined as (see figure 2.4): $\mathcal{LS}_D = \langle P_{\mathcal{LS}_D}, T_{\mathcal{LS}_D}, Pre_{\mathcal{LS}_D}, Post_{\mathcal{LS}_D} \rangle$, such that $T_{\mathcal{LS}_D} \cap T = D$, $T^\bullet_{\mathcal{LS}_D} \cup {}^\bullet T_{\mathcal{LS}_D} = P_{\mathcal{LS}_D}$, and $P_{\mathcal{LS}_D} \cap P = \emptyset$.*

Lemma 2.1.1 *Let $\mathcal{N}$ be a net and D an ECS of $\mathcal{N}$. Let $\mathcal{LS}_D$ be a local scheduler for D. If $\mathcal{N}$ and $\mathcal{LS}_D$ are structurally live in isolation, then the net $\mathcal{N}^{\mathcal{LS}_D}$ obtained by merging the common transitions of $\mathcal{N}$ and $\mathcal{LS}_D$ is structurally live.*

Proof. Let M_0 and $M_{0\mathcal{LS}_D}$ be initial markings making live the nets $\mathcal{N}$ and $\mathcal{LS}_D$, respectively. Let $M_0^{\mathcal{LS}_D}$ be an initial marking of $\mathcal{N}^{\mathcal{LS}_D}$ such that its projection on P is M_0 and its projection on $P_{\mathcal{LS}_D}$ is $M_{0\mathcal{LS}_D}$. Let $M^{\mathcal{LS}_D} \in R(\mathcal{N}^{\mathcal{LS}_D}, M_0^{\mathcal{LS}_D})$ and t be a transition of $\mathcal{N}$. We prove that there exists a firing sequence, $\sigma^{\mathcal{LS}_D}$, in $\langle \mathcal{N}^{\mathcal{LS}_D}, M^{\mathcal{LS}_D} \rangle$ that yields to a marking enabling t (i.e., the net $\mathcal{N}^{\mathcal{LS}_D}$ is live under $M_0^{\mathcal{LS}_D}$).

The projection of $M^{\mathcal{LS}_D}$ on P is a marking $M \in R(\mathcal{N}, M_0)$ from which there exists at least one $\sigma \in L(\mathcal{N}, M)$, yielding to a marking M' that enables t (because the net $\mathcal{N}$ is live). From this fact, three cases arise:

a) If σ does not contain any transition belonging to D then it is also firable in the net $\mathcal{N}^{\mathcal{LS}_D}$.

b) If σ contains one transition $t_a \in D$, that is $\sigma = \sigma_0 t_a \sigma_a$, then there exist $\delta_D + 1$ firable sequences from M of the form $\sigma_0 t_i \sigma_i$, $t_i \in D$,

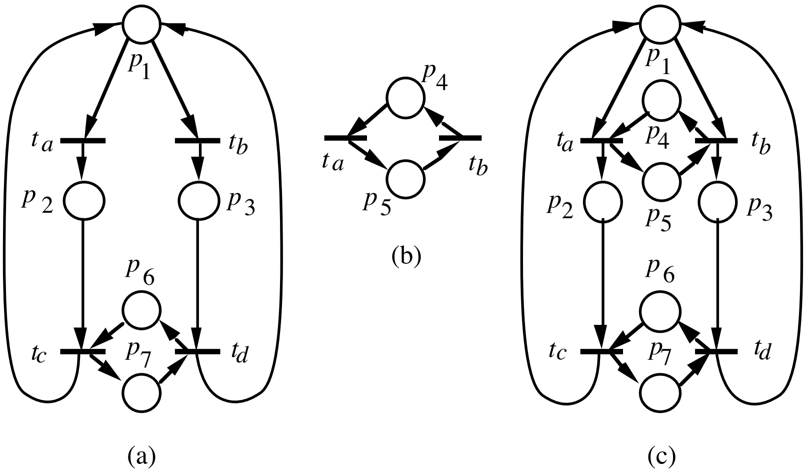


Figure 2.5: Counter-example to the converse of lemma 2.1.1.

$i = 1, \ldots, \delta_D+1$, that allow to reach a marking enabling t. This is because $\mathcal{N}$ is live and $M[\sigma_0\rangle M_D$; $\forall t_i \in D, M_D[t_i\rangle M_i \in R(\mathcal{N}, M_0)$ and $\forall i = 1, \ldots, \delta_D + 1$, $M_i[\sigma_i\rangle M_i'[t\rangle$. Therefore, at least one of the sequences $\sigma_0 t_i \sigma_i$ can be fired in $\mathcal{N}^{\mathcal{LS}_D}$: σ_0 and σ_i are firable according to the above case (a); at least one $t_i \in D$ is firable because $\mathcal{LS}_D$ is a live net (eventually, after the firing of some internal transitions of the local scheduler in order to enable t_i).

c) If σ contains more than one transition of D, we can find a firable sequence in $\mathcal{N}$ that is firable in $\mathcal{N}^{\mathcal{LS}_D}$. This can be done by applying repeatedly the above case (b).

Liveness of transitions belonging to $\mathcal{LS}_D$ can be proved with similar arguments. Therefore, the net $\mathcal{N}^{\mathcal{LS}_D}$ is live under $M_0^{\mathcal{LS}_D}$ and then structurally live. ■

Unfortunately, the converse of lemma 2.1.1 is not true. Let us consider, for instance, the structurally non-live net depicted in figure 2.5.a. The net of figure 2.5.b is a structurally live local scheduler for transitions a and b. The composition of the two nets is the net of figure 2.5.c that now is structurally live.

In the sequel, we consider a simple class of local schedulers called *regulation circuits*. These nets are used as a tool to prove the theorem 2.1.1. Nevertheless, they are not the unique local schedulers that can be used for that purpose.

Definition 2.1.6 (Regulation circuit) [CCS90d] *Let t_a and t_b be two transitions of $\mathcal{N}$ in equality conflict relation. A regulation circuit for t_a and t_b is a net $r_{ab} = \langle P_{r_{ab}}, T_{r_{ab}}, Pre_{r_{ab}}, Post_{r_{ab}} \rangle$, where $P_{r_{ab}} = \{p_{ab}, p_{ba}\}$, $T_{r_{ab}} = \{t_a, t_b\}$, ${}^\bullet p_{ab} = \{t_a\}$, $p_{ab}^\bullet = \{t_b\}$, $PRE_{r_{ab}}[p_{ab}, t_b] = POST_{r_{ab}}[p_{ab}, t_a] = 1$, ${}^\bullet p_{ba} = \{t_b\}$, $p_{ba}^\bullet = \{t_a\}$, and $PRE_{r_{ab}}[p_{ba}, t_a] = POST_{r_{ab}}[p_{ba}, t_b] = 1$.*

As an example, the local scheduler depicted in figure 2.5 is a regulation circuit for t_a and t_b. The composition of $\mathcal{N}$ and r_{ab} (by merging the common transitions t_a and t_b) will be denoted as $\mathcal{N}^{r_{ab}}$. The incidence matrix of $\mathcal{N}^{r_{ab}}$ will be denoted as $C^{r_{ab}}$.

Let $\mathcal{N}$ be a net and $D = \{t_i \mid i = 1, \ldots, \delta_D + 1\}$ be an ECS. The net obtained from $\mathcal{N}$ by adding a regulation circuit per each pair of transitions $t_k, t_{k+1} \in D, k = 1, \ldots, \delta_D$, will be denoted as $\mathcal{N}^{R_D}$, and its corresponding incidence matrix as C^{R_D}. The net obtained by adding regulation circuits for all ECSs as above will be denoted as $\mathcal{N}^R$, and its corresponding incidence matrix as C^R.

The following lemmatas present some properties of $\mathcal{N}^{R_D}$ derived from the corresponding properties of $\mathcal{N}$.

Lemma 2.1.2 *Let $\mathcal{N}$ be a net and $D = \{t_i \mid i = 1, \ldots, \delta_D + 1\}$ be an ECS. If $\mathcal{N}$ is structurally live and structurally bounded then $\mathcal{N}^{R_D}$ is structurally live and structurally bounded.*

Proof. The set of regulation circuits added to $\mathcal{N}$ is a local scheduler for D. This local scheduler is structurally live in isolation (this is obvious, putting enough tokens at each of the regulation circuits). The net $\mathcal{N}$ is also structurally live and then, by lemma 2.1.1, the net $\mathcal{N}^{R_D}$ is structurally live.

All places of $\mathcal{N}$ are structurally bounded. Taking into account the definitions of $p_{t_k t_{k+1}}$ and $p_{t_{k+1} t_k}$ it is easy to see that $C^{R_D}[p_{t_k t_{k+1}}] + C^{R_D}[p_{t_{k+1} t_k}] = 0$ (i.e., the sum of the rows in the incidence matrix corresponding to these places is zero). Therefore all new places added to

$\mathcal{N}$ are also structurally bounded and then $\mathcal{N}^{R_D}$ is structurally bounded. ■

Lemma 2.1.3 *Let $D = \{t_i \mid i = 1, \ldots, \delta_D + 1\}$ be an ECS. If $\mathcal{N}$ is structurally live and structurally bounded then*

$$\min\{m-1, n+2\delta_D-1\} \geq rank(C^{R_D}) = rank(C) + \delta_D \qquad (2.7)$$

Proof. $\mathcal{N}^{R_D}$ is structurally live and structurally bounded (lemma 2.1.3), thus conservative and consistent [Sil85]. Therefore, $rank(C^{R_D}) \leq \min\{m^{R_D}-1, n^{R_D}-1\}$, where $m^{R_D} = |T^{R_D}| = |T| = m$ and $n^{R_D} = |P^{R_D}| = |P| + |P_{r_{12}}| + \cdots + |P_{r_{kk+1}}| + \cdots + |P_{r_{\delta_D-1\delta_D}}| = n + 2\delta_D$. So, we obtain: $rank(C^{R_D}) \leq \min\{m-1, n+2\delta_D-1\}$.

Let $\mathcal{N}^{p_{t_2t_1}}$ be a net obtained from $\mathcal{N}$ by adding the place $p_{t_2t_1}$ belonging to the regulation circuit $r_{1,2}$. $\mathcal{N}^{p_{t_2t_1}}$ is non-conservative because for all marking that enables the transitions of D we can decide to fire always the transition t_2 (i.e., the place $p_{t_2t_1}$ is structurally unbounded). Then we can conclude that there is not a vector Y such that $Y^T \cdot C = C^{p_{t_2t_1}}[p_{t_2t_1}]$ (see proposition 2.8 in [CCS90d]) (i.e., the row vector $C^{p_{t_2t_1}}[p_{t_2t_1}]$ is linearly independent with respect to the row vectors of the incidence matrix of the net $\mathcal{N}$). Therefore, $rank(C^{p_{t_2t_1}}) = rank(C) + 1$. If we add the place $p_{t_1t_2}$ to the net $\mathcal{N}^{p_{t_2t_1}}$ we obtain the net $\mathcal{N}^{r_{1,2}}$. This last net has the same rank that the net $\mathcal{N}^{p_{t_2t_1}}$ because $C^{r_{1,2}}[p_{t_2t_1}] = -C^{r_{1,2}}[p_{t_1t_2}]$. Therefore, $rank(C^{r_{1,2}}) = rank(C) + 1$.

Let $\mathcal{N}^{R_{k-1}}$ be the net obtained from $\mathcal{N}$ by adding the regulation circuits $r_{1,2}, \ldots, r_{k-1,k}$. $\mathcal{N}^{R_{k-1}}$ verifies $rank(C^{R_{k-1}}) = rank(C) + k - 1$. We prove now that if we add the regulation circuit $r_{k,k+1}$ to the net $\mathcal{N}^{R_{k-1}}$ then $rank(C^{R_k}) = rank(C^{R_{k-1}}) + 1$.

We add the place $p_{t_{k+1}t_k}$ belonging to the regulation circuit $r_{k,k+1}$ to the net $\mathcal{N}^{R_{k-1}}$. This place is unbounded because for all marking that enables some transition of the set $\{t_1, \ldots, t_k\}$, t_{k+1} is also enabled at this marking and then we can decide to fire always the transition t_{k+1}. Then the row vector $C^{R_k}[p_{t_{k+1}t_k}]$ is linearly independent with respect to the row vectors of the incidence matrix of the net $\mathcal{N}^{R_{k-1}}$ (by proposition 2.8 in [CCS90d]). Therefore, $rank(C^{R_k}) = rank(C^{R_{k-1}}) + 1$ (because $C^{R_k}[p_{t_{k+1}t_k}] = -C^{R_k}[p_{t_kt_{k+1}}]$).

The number of added regulation circuits is δ_D, hence $rank(C^{R_D}) = rank(C) + \delta_D$. ■

Lemma 2.1.4 *Let $\mathcal{N}$ be a structurally live and structurally bounded net. Then $rank(C) \leq m - \delta - 1$, where C is the incidence matrix of $\mathcal{N}$, $m = |T|$, and δ is the number of non-redundant free conflicts of the net.*

Proof. $\mathcal{N}$ is conservative and consistent [Sil85]. If we add a local scheduler per each ECS of the net, we obtain a net denoted $\mathcal{N}^R$ that satisfies: $\min\{m-1, n+2\delta-1\} \geq rank(C^R) = rank(C) + \delta$. Then, $rank(C) \leq \min\{m-\delta-1, n+\delta-1\}$. Taking into account that the net $\mathcal{N}$ is conservative and consistent, we also have that $rank(C) \leq \min\{m-1, n-1\}$. Therefore, combining the two above upper bounds of $rank(C)$, we obtain: $rank(C) \leq \min\{m-\delta-1, n-1\}$. But, $\mathcal{N}$ being conservative, $rank(C) \leq n-1$ and the lemma follows. ■

Proof of theorem 2.1.1. The rank equality condition holds iff $\mathcal{N}^R$ has a unique minimal T-semiflow. So let us prove this condition.

The number of minimal T-semiflows of $\mathcal{N}^R$ is greater than or equal to 1 because this net is consistent.

We compute T-semiflows, $X \geq 0$ and $C \cdot X = 0$, applying the algorithm presented in [CS89b] to the net $\mathcal{N}^R$. To do so, we eliminate first the places $p_{t_i t_{i+1}}$ that connect two transitions in equality conflict relation (obviously, if we eliminate $p_{t_i t_{i+1}}$ we also eliminate $p_{t_{i+1} t_i}$ because $C^R[p_{t_i t_{i+1}}] = -C^R[p_{t_{i+1} t_i}]$). The elimination of $p_{t_i t_{i+1}}$ generates a unique new column that is a linear combination of the columns corresponding to t_i and t_{i+1}. In order to eliminate $p_{t_{i+1} t_{i+2}}$ we generate again a unique column that is a linear combination of the above added column and the column of t_{i+2}. If we repeat this procedure for all places $p_{t_i t_{i+1}}$ belonging to an ECS we obtain a unique new column in which all entries corresponding to places of the local scheduler are zero. The non-null entries of this row are ${}^\bullet ECS \cup ECS^\bullet$. Applying this procedure for all ECS of the net we obtain a matrix in which there is a new column per ECS and all columns in the original net corresponding to transitions that do not belong to any ECS. This matrix can be interpreted as the incidence matrix of a new net with at most one minimal T-semiflow iff the original net is an FRT-net. This is because, if the original net was an FRT-net, all its T-semiflows would be freely related (by definition of FRT-net), thus freely connected by pairs. And this occurs iff after the application of the above procedure (i.e., after the addition of the regulation circuits) each pair of originally freely connected T-semiflows constitute a unique T-semiflow.

Therefore, applying the rank formula of lemma 2.1.3 with $rank(C^R) = m - 1$ we obtain: $rank(C) = m - \delta - 1$. ▪

2.1.1.3 Qualitative properties

The next result gives a method for the computation of the vector of visit ratios for transitions of a structurally live and structurally bounded FRT-net (provided liveness), from the knowledge of the net structure and the routing rates at equality conflict sets.

Theorem 2.1.2 *Let $\mathcal{N}$ be a structurally live and structurally bounded FRT-net. Let C be the incidence matrix of $\mathcal{N}$, and R the matrix (with δ independent rows, where δ is the number of non-redundant free conflicts of $\mathcal{N}$) that defines the relative rates of transitions in equality conflict relation (i.e., the routing at the equality conflict sets). Then, the vector of visit ratios $\vec{v}^{(j)}$ normalized, for instance, for transition t_j can be computed from C and R solving the following linear system of equations:*

$$\begin{pmatrix} C \\ R \end{pmatrix} \cdot \vec{v}^{(j)} = 0, \quad v_j^{(j)} = 1 \tag{2.8}$$

(Note that this computation only makes sense when infinite behaviour is possible for the net from a given initial marking, in other words, when the net is deadlock-free.)

Proof. We only have to check that the above system has a unique solution. By theorem 2.1.1, the number of independent rows of matrix C is $m - \delta - 1$. Therefore, the $m - \delta - 1$ independent conditions given by $C \cdot \vec{v}^{(j)} = 0$ plus the δ independent conditions given by $R \cdot \vec{v}^{(j)} = 0$ plus the normalization condition $\vec{v}^{(j)}(t_j) = 1$ are enough to determine exactly the m components of the vector $\vec{v}^{(j)}$. ▪

From the above theorem, as we announced previously, for structurally live and structurally bounded FRT-nets we have:

$$\vec{v}^{(j)} = \varphi(\mathcal{N}, \mathcal{R}) \tag{2.9}$$

and the next complexity result follows:

Corollary 2.1.2 *The computation of the vector of visit ratios for transitions of a structurally live and structurally bounded FRT-net is polynomial on the net size.*

As an example, let us consider again the net depicted in figure 2.3. The vector of visit ratios must be a right annuller of the incidence matrix, hence a linear combination of a basis of T-semiflows:

$$\vec{v}^{(1)} = \sum_{i=1}^{4} \alpha_i X_i \tag{2.10}$$

where X_i, $i = 1, \ldots, 4$, are the minimal T-semiflows (2.5) of the net. If r_1, r_2 are the routing rates of t_1, t_2 in the conflict at p_1; r_3, r_4 the routing rates of t_3, t_4 in the conflict at p_2; and r_5, r_6 the routing rates of t_5, t_6 in the conflict at p_3, then $\vec{v}^{(1)}$ must satisfy:

$$\begin{aligned} r_2 v_1^{(1)} &= r_1 v_2^{(1)} \\ r_4 v_3^{(1)} &= r_3 v_4^{(1)} \\ r_6 v_5^{(1)} &= r_5 v_6^{(1)} \end{aligned} \tag{2.11}$$

And together with the normalization requirement:

$$v_1^{(1)} = 1 \tag{2.12}$$

the four parameters α_i, $i = 1, \ldots, 4$, can be determined.

Another interesting qualitative property follows from theorem 2.1.2, that does not hold for general nets:

Property 2.1.3 *Let $\mathcal{N}$ be a structurally live and structurally bounded FRT-net. Then $\mathcal{N}$ is live if and only if it is deadlock-free.*

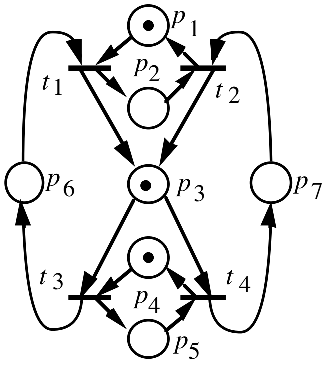


Figure 2.6: The addition of a token to p_5 kills the net (sequence $\sigma = t_4$ leads to a deadlock).

Proof. The "only if" direction is always true by definitions of liveness and deadlock-freeness. The other direction follows from theorem 2.1.2. The vector of visit ratios for transitions is univocally determined in that theorem and all its components are non-null. If an infinite behaviour of the net always occur (i.e., if the net is deadlock-free) the limit throughput of transitions must be proportional to the vector of visit ratios, which is strictly positive. Therefore, the net is live. ▪

From previous considerations and results, weak ergodicity of the firing process of live and structurally bounded FRT-nets follows:

Theorem 2.1.3 *Let $\langle \mathcal{N}, M_0 \rangle$ be a live and structurally bounded FRT-net. Then its firing process is weakly ergodic.*

For finishing this section, let us present some negative answers to several questions about qualitative properties of general FRT-nets. All of them are proven by giving the corresponding counter-example. The first one assures that liveness is not a monotonous property for the increasing of initial marking, and can be proven looking at the live and structurally bounded FRT-net depicted in figure 2.6. The addition of a token to p_5 kills the net.

Property 2.1.4 *Let $\langle \mathcal{N}, M_0 \rangle$ be a structurally bounded FRT-net and $M_0' \geq M_0$. Liveness of $\langle \mathcal{N}, M_0 \rangle$ does not imply liveness of $\langle \mathcal{N}, M_0' \rangle$.*

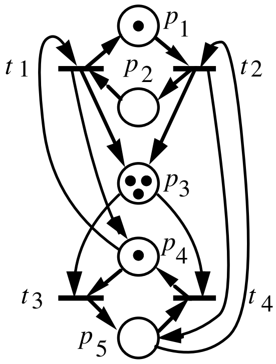


Figure 2.7: Live and bounded FRT-net which is not structurally bounded.

Other negative result is shown by the net depicted in figure 2.7. It is a live and bounded FRT-net. But the addition of a token to place p_5 makes that the sequence t_2t_1 can be fired *ad infinitum* leading to an unbounded marking at place p_3. In other words:

Property 2.1.5 *Let $\langle\mathcal{N}, M_0\rangle$ be a live FRT-net. The boundedness of $\langle\mathcal{N}, M_0\rangle$ does not imply the structural boundedness of $\mathcal{N}$.*

Given a place p of a marked net, the maximum number of tokens at this place over all reachable markings is called the *marking bound* of p, and denoted as $B(p)$. The structural counterpart of this concept can be defined in terms of a linear programming problem as follows:

Definition 2.1.7 (Structural marking bound) [SC88] *Let $\langle\mathcal{N}, M_0\rangle$ be a marked Petri net. The structural marking bound of a given place p of $\mathcal{N}$ is*

$$SB(p) \overset{\text{def}}{=} \begin{array}{ll} \textit{maximize} & M(p) \\ \textit{subject to} & M = M_0 + C \cdot \vec{\sigma} \\ & M, \vec{\sigma} \geq 0 \end{array} \qquad \text{(LPP2)}$$

Since $M_0[\sigma\rangle M$ implies $M = M_0 + C \cdot \vec{\sigma} \geq 0$ with $\vec{\sigma} \geq 0$, but the reverse is not true in general, $SB(p)$ is greater than or equal to $B(p)$.

Now, let us consider the net depicted in figure 2.8. It is a live and structurally bounded FRT-net, and it verifies: (1) The marking bound

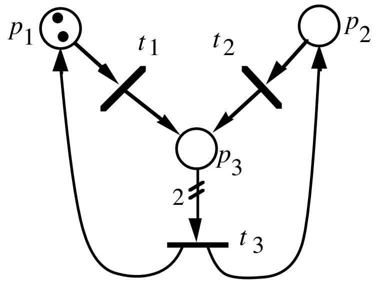


Figure 2.8: A non-reversible live and structurally bounded FRT-net.

of place p_2 is 1 and it is different from its structural marking bound, which is 2; (2) The enabling bound of transition t_2 is 1 and it is different from its structural enabling bound, which is 2; (3) It is not reversible (i.e., the initial marking is not a home state; (4) The liveness bound of transition t_1 is 1 and it is different from its enabling bound, which is 2. Therefore, the next result can be stated:

Property 2.1.6 *Let $\langle \mathcal{N}, M_0 \rangle$ be a live and structurally bounded FRT-net. Then,*

1. *Equality between the marking and the structural marking bound of a place cannot be assured.*
2. *Equality between the enabling and the structural enabling bound of a transition cannot be assured.*
3. *Reversibility of the net cannot be assured.*
4. *Equality between the liveness and the enabling bound of a transition cannot be assured.*

Item 3 in the above property assures that, in general, the initial marking of a live and structurally bounded FRT-net is not a home state. Moreover, the existency of home states is not guaranteed for live and structurally bounded FRT-nets, as can be seen for the net depicted in figure 2.9.

Property 2.1.7 *Let $\langle \mathcal{N}, M_0 \rangle$ be a live and structurally bounded FRT-net. Then, the existency of home state cannot be assured.*

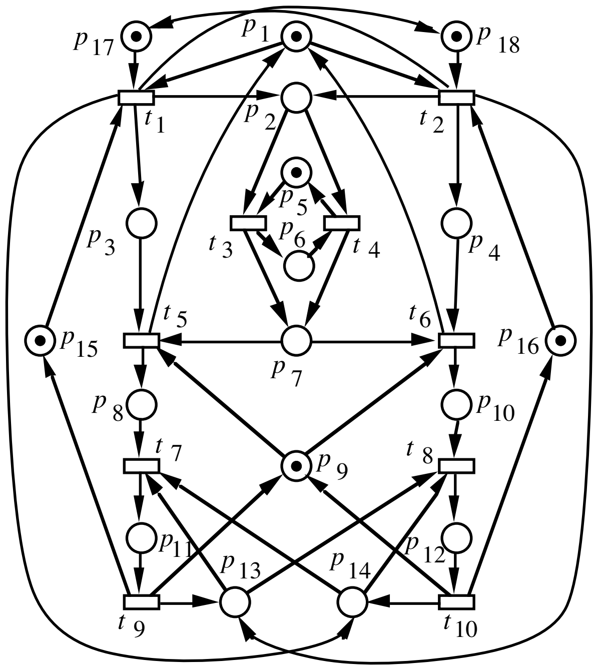


Figure 2.9: A live and structurally bounded FRT-net without home states.

Therefore, from the above property the next result follows (see section 1.2.3.3):

Property 2.1.8 *Let $\langle \mathcal{N}, M_0 \rangle$ be a live and structurally bounded FRT-net. Then, (even weakly) marking ergodicity cannot be assured.*

In the next sections we identify some subclasses of FRT-nets. Their interest arises from the fact that the previous negative statements for general FRT-nets turn into positive results for some particular subclasses.

2.1.2 Mono-T-semiflow, structurally decision-free nets, and marked graphs

In this section we define and give some interesting properties for a subclass of FRT-nets having only one minimal T-semiflow. The vector of visit ratios can be computed for them just from the incidence matrix of the net, since it is proportional to the unique minimal T-semiflow. Nets belonging to this subclass are called *mono-T-semiflow.* Later, *structurally decision-free* nets are shown to be a subclass of mono-T-semiflow nets with some particular properties. Finally, *marked graphs*, a well-known Petri net subclass, is identified as a subclass of structurally decision-free nets, for which the vector of visit ratios is always the unity vector.

2.1.2.1 Mono-T-semiflow nets

We introduce a class of structurally characterized nets, called *mono-T-semiflow.*

Definition 2.1.8 (Mono-T-semiflow nets) *A net $\mathcal{N}$ is called mono-T-semiflow iff it has a unique minimal T-semiflow.*

In a mono-T-semiflow net, conflicts may be reached, so that different behaviours can occur. However, from the steady-state performance point of view, these decisions yield a unique vector of visit ratios for transitions, provided that the net is live (all different behaviours let the same set of transitions, characterized by the only T-semiflow of

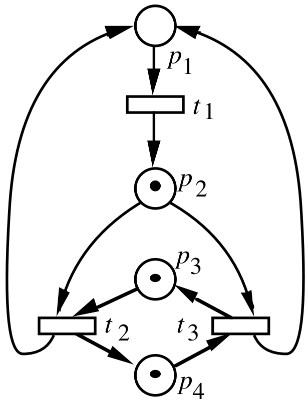


Figure 2.10: A live and structurally bounded mono-T-semiflow net.

the net, fire, perhaps in a different order). For example, $\sigma_a = t_2t_1t_3$ and $\sigma_b = t_3t_1t_2$ are possible sequences in the net of figure 2.10, both firable from M_0. Even if the performance can be equal for any conflict resolution policy (which is not always true), from the functional point of view the results can be different (imagine t_2 and t_3 be two non-commutative operations).

Note that mono-T-semiflow nets constitute (by definition) a subclass of FRT-nets, in which each pair of transitions in structural conflict (i.e., sharing an input place) belongs to the unique minimal T-semiflow:

Property 2.1.9 *Mono-T-semiflow nets are FRT-nets.*

For the particular case of consistent nets, the following characterization of mono-T-semiflow subclass can be stated:

Theorem 2.1.4 *Let $\mathcal{N}$ be a consistent net and C its incidence matrix. Then $\mathcal{N}$ is mono-T-semiflow if and only if $rank(C) = m - 1$.*

Proof. If a net is mono-T-semiflow then the dimension of the space of right annullers of C is greater than or equal to 1, thus $rank(C) \leq m-1$. We prove $rank(C) \geq m - 1$ by contradiction. Suppose X and X' are two linearly independent right annullers of C, where X is a T-semiflow such that $X \geq \mathbb{1}$ (it exists by consistency). Then, two independent and positive right annullers of C can be constructed: $X_1 = X$ and

$X_2 = X' + kX$, taking $k > 0$ large enough, and this is against the hypothesis of mono-T-semiflow. ∎

From previous theorem, the next statement follows:

Corollary 2.1.3 *If $\mathcal{N}$ is consistent, deciding if $\mathcal{N}$ belongs to the class of mono-T-semiflow nets is polynomial on the net size.*

Now, we present an efficient method for the computation of the vector of visit ratios for transitions of a structurally live and structurally bounded mono-T-semiflow net (provided liveness), from the net structure. It follows from theorems 2.1.2 and 2.1.4.

Theorem 2.1.5 *Let $\mathcal{N}$ be a structurally live and structurally bounded mono-T-semiflow net and C its incidence matrix. Then, the vector of visit ratios $\vec{v}^{(j)}$ normalized, for instance, for transition t_j can be computed from C solving the following linear system of equations:*

$$C \cdot \vec{v}^{(j)} = 0, \quad v_j^{(j)} = 1 \tag{2.13}$$

From the above theorem, for structurally live and structurally bounded mono-T-semiflow nets we have:

$$\vec{v}^{(j)} = \varphi(\mathcal{N}) \tag{2.14}$$

and the next complexity result follows:

Corollary 2.1.4 *The computation of the vector of visit ratios for transitions of a structurally live and structurally bounded mono-T-semiflow net is polynomial on the net size.*

Since mono-T-semiflow nets are FRT-nets, the “good” properties exhibited for these are inherited by those. Related with the “bad” results presented for general FRT-nets in properties 2.1.4, 2.1.5, 2.1.6, 2.1.7, and 2.1.8, the same can be stated for mono-T-semiflow nets. This can be seen looking at the FRT-nets depicted in figures 2.6, 2.7, 2.8, and 2.9, that were used as counter-examples. All of them are also mono-T-semiflow nets.

In the next section, we identify a subclass of mono-T-semiflow nets for which some of the previous negative results change.

2.1.2.2 Structurally decision-free nets

Let us introduce a class of structurally defined nets for which never exist conflicts, whichever it is the initial marking.

Definition 2.1.9 (Structurally decision-free nets) [CCS89] *A net $\mathcal{N}$ is said to be structurally decision-free iff for all place p: $|p^\bullet| \leq 1$.*

For example, the net depicted in figure 2.8 is a live structurally bounded and structurally decision-free net. Now, we prove that all structurally live structurally bounded and structurally decision-free nets are mono-T-semiflow:

Property 2.1.10 *Let $\mathcal{N}$ be a connected, consistent, and structurally decision-free net. Then $\mathcal{N}$ is mono-T-semiflow.*

Proof. Since the net is consistent, it has at least a T-semiflow. It has not more than one because if a transition belongs to a T-semiflow X, all output transitions of its output places must belong to X (because the net is structurally decision-free). Since the net is connected there exists at most one T-semiflow. ■

The reverse of property 2.1.10 is not true. For example the net of figure 2.10 is mono-T-semiflow but is not structurally decision-free. Thus, consistent and structurally decision-free nets constitute a proper subclass of mono-T-semiflow nets.

We have seen that, in general, live structurally bounded mono-T-semiflow nets have not home state. However the subclass of deadlock-free and bounded structurally decision-free nets have home state.

Property 2.1.11 *Let $\langle \mathcal{N}, M_0 \rangle$ be a deadlock-free and bounded structurally decision-free net. Then, it has a home state.*

Proof. Boundedness of the net guarantees a bounded number of reachable markings. In this case, the absence of decisions assures the existence of home state. ■

As a corollary, ergodicity of the marking process of such nets follows:

Corollary 2.1.5 *Let $\langle \mathcal{N}, M_0 \rangle$ be a bounded structurally decision-free net. Then its marking process is weakly ergodic. Moreover, if the net is Markovian, its marking process is strongly ergodic.*

2.1.2.3 Marked graphs

In this section, ordinary structurally decision-free nets without multiple attributions to places are considered: the well-known subclass of em marked graphs.

Marked graphs can be seen as a generalization of the classical PERT tool [MP70]. With PERT model, the relationship among the tasks of a project can be represented by a network of activities (arrows) and events (nodes). Timing interpretation can be added to activities for the purpose of evaluating the completion time of the project. The obtained network is an *acyclic* graph, i.e., repetitive systems cannot be modelled.

With marked graphs, *cyclic* behaviours can be modelled as well as many different classes of non shared resources for the realization of activities (tokens at places of the net).

Let us briefly recall what marked graphs are and some of their basic properties. Marked graphs allow to model *concurrency and synchronization but no decisions* because they are structurally decision-free nets.

Definition 2.1.10 (Marked graphs) [CHEP71] *Marked graphs are ordinary Petri nets (i.e., pre- and post-incidence functions taking values in* $\{0, 1\}$*) such that for all place p:* $|{}^\bullet p| = |p^\bullet| = 1$*.*

Property 2.1.12 *Let* $\mathcal{N}$ *be a marked graph.*

1. $\mathcal{N}$ *is structurally decision-free.*
2. $\mathcal{N}$ *is consistent and its unique minimal T-semiflow is* $X = \mathbb{1}$*.*
3. *The vector of visit ratios for transitions of* $\mathcal{N}$ *is* $\vec{v} = \mathbb{1}$ *(provided liveness), independently of the initial marking and of the average service times associated with transitions.*

The reverse of property 2.1.12.1 is not true. For example, the net depicted in figure 2.8 is structurally decision-free but it is not a marked graph.

Some interesting results from qualitative theory of marked graphs are recalled bellow. In particular, checking their liveness characterization is polynomial on the net size.

Theorem 2.1.6 [Mur89] *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked graph.*

1. *The elementary P-semiflows of $\mathcal{N}$ are exactly its directed circuits.*
2. *$\langle \mathcal{N}, M_0 \rangle$ is live iff all its directed circuits are marked.*

Putting an initial marking large enough, after marking all circuits the system will be live:

Corollary 2.1.6 *Marked graphs are structurally live.*

Corollary 2.1.7 *The liveness of a marked graph can be decided in polynomial time on its size, checking that there is no unmarked P-semiflow:*

$$\nexists Y \gneq 0,\ \ Y^T \cdot C = 0,\ \ Y^T \cdot M_0 = 0 \tag{2.15}$$

From the theorem 2.1.6.2, the following liveness monotonicity result follows:

Corollary 2.1.8 *If $\langle \mathcal{N}, M_0 \rangle$ is a live marked graph and $M_0' \geq M_0$ then $\langle \mathcal{N}, M_0' \rangle$ is live.*

For live marked graphs, boundedness and structural boundedness are equivalent properties:

Property 2.1.13 [Sil85] *Let $\mathcal{N}$ be a marked graph.*

1. *The following three statements are equivalent:*
   - *i) $\mathcal{N}$ is structurally bounded.*
   - *ii) $\mathcal{N}$ is strongly connected.*
   - *iii) $\mathcal{N}$ is conservative (i.e., $\exists Y \geq \mathbb{1}, Y^T \cdot C = 0$).*
2. *Let $\langle \mathcal{N}, M_0 \rangle$ be live. Then $\langle \mathcal{N}, M_0 \rangle$ is bounded iff $\mathcal{N}$ is structurally bounded.*

Hopefully, the reachability problem, i.e., the efficient characterization of reachable markings, has a satisfactory solution for live marked graphs:

Theorem 2.1.7 [Mur77] *Let $\langle \mathcal{N}, M_0 \rangle$ be a live marked graph. The three following statements are equivalent:*

i) *$M \in R(\mathcal{N}, M_0)$, i.e., M is reachable from M_0.*

ii) *$M = M_0 + C \cdot \vec{\sigma}$, with $M, \vec{\sigma} \geq 0$.*

iii) *$B_f \cdot M = B_f \cdot M_0$, with B_f the fundamental circuit matrix of the graph, and $M \geq 0$.*

According to the above theorem $M \in R(\mathcal{N}, M_0)$ if and only if $M_0 \in R(\mathcal{N}, M)$. In other words:

Corollary 2.1.9 *Live marked graphs are reversible.*

Weak ergodicity of the firing and the marking processes for live and strongly connected marked graphs follows (from corollary 2.1.5), since they are bounded and structurally decision-free nets:

Corollary 2.1.10 *The firing process of a live marked graph is weakly ergodic. If the net is strongly connected the marking process is also weakly ergodic. Moreover, if the net is Markovian, its marking process is strongly ergodic.*

Finally, the next interesting property of live marked graphs, can be deduced:

Property 2.1.14 *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked graph, and t a transition of $\mathcal{N}$. Then $E(t) = L(t) = SE(t)$.*

Proof. Marked graphs are reversible, by corollary 2.1.9. Then, by property 1.2.2, $E(t) = L(t)$ for all transitions t. Finally, $E(t) = SE(t)$, by theorem 2.1.7.i and ii. ■

This allows an efficient computation of enabling and liveness bounds based on the linear programming problem (LPP1) that characterizes the structural enabling bound of transitions.

2.1.3 Free choice nets

Another interesting subclass of FRT-nets is that of *free choice nets.* Free choice nets [Hac72] are a well-known subclass of ordinary Petri nets that hold a particularly restricted interplay between concurrency and decisions. They are rich enough to be non-trivial but restricted enough to allow a number of interesting results that do not hold in general and that constitute a quite elegant theory (see, e.g., [Hac72, TV84,Bes87,CCS90a,Esp90,ES90]).

Free choice nets allow both synchronization and conflict but in a restricted and disciplinated way. In a free choice net, if a place has a shared output transition then it is the only output transition of this place. And, equivalently, if a transition has a shared input place then it is the only input place of this transition.

Definition 2.1.11 (Free choice nets) [Hac72] *Free choice nets are ordinary Petri nets (i.e., pre- and post-incidence functions taking values in* $\{0,1\}$*) such that for all place* p*:* $|p^\bullet| > 1 \Rightarrow {}^\bullet(p^\bullet) = \{p\}$*.*

Since all decisions are free in a free choice net, all the T-semiflows are freely related and the following inclusion holds:

Property 2.1.15 *Free choice (connected) nets are FRT-nets.*

Let us remark also that marked graphs, presented in previous section, are free choice nets.

This section introduces a minimum of qualitative results from the large body of free choice nets theory. Additional qualitative results are derived from the quantitative/performance based approach introduced in this work. This approach clearly points out the interest of interleaving the qualitative and quantitative theories.

Let $\mathcal{N} = \langle P, T, Pre, Post\rangle$ be a Petri net and $P' \subseteq P$. $\mathcal{N}' = \langle P', T', Pre', Post'\rangle$ is called a *P-component* of $\mathcal{N}$ iff $\mathcal{N}'$ is the subnet of $\mathcal{N}$ generated by P' (i.e., $T' \subseteq T$ and $Pre', Post'$ are the restrictions of $Pre, Post$ to P' and T') and $\forall t \in T'$: $|{}^\bullet t \cap P'| \leq 1 \wedge |t^\bullet \cap P'| \leq 1$.

An important result in the structure theory of free choice nets assures that each minimal P-semiflow of a structurally live and structurally bounded free choice net generates a P-component, and that liveness can be assured when all the P-components are marked:

Theorem 2.1.8 *Let $\mathcal{N} = \langle P, T, Pre, Post \rangle$ be a structurally live and structurally bounded free choice net.*

1. [ES90] *$Y \geq 0$ is a minimal P-semiflow of $\mathcal{N}$ iff the two following conditions hold:*

   a) $\forall p \in P$: $Y(p) \in \{0, 1\}$

   b) $\exists\ \mathcal{N}' = \langle P', T', Pre', Post' \rangle$ *P-component of $\mathcal{N}$ and* $||Y|| = P'$

2. [Esp90] *If M_0 is a given initial marking for $\mathcal{N}$, $\langle \mathcal{N}, M_0 \rangle$ is live if and only if all its P-components are marked.*

Note that the above theorem is a generalization of theorem 2.1.6 (stated for marked graphs), for the case of structurally live and structurally bounded free choice nets. The characterization of liveness for such nets is the same than for marked graphs (corollary 2.1.7):

Corollary 2.1.11 *The liveness of a structurally live and structurally bounded free choice net can be decided in polynomial time on its size, checking that there is no unmarked P-component:*

$$\nexists Y \gneq 0,\ Y^T \cdot C = 0,\ Y^T \cdot M_0 = 0 \qquad (2.16)$$

From the previous characterization of liveness, a monotonicity result trivially follows for structurally bounded nets:

Corollary 2.1.12 *If $\langle \mathcal{N}, M_0 \rangle$ is a live structurally bounded free choice net and $M_0' \geq M_0$ then $\langle \mathcal{N}, M_0' \rangle$ is live.*

In fact, structural boundedness is not necessary in the previous property (since liveness monotonicity can be derived from a more general characterization of liveness for free choice nets [Hac72]).

Unfortunately, for general (non-structurally bounded) free choice nets, the following "bad" result has been proven:

Theorem 2.1.9 [JLL77] *Let $\langle \mathcal{N}, M_0 \rangle$ be a free choice net. The decision of non-liveness for $\langle \mathcal{N}, M_0 \rangle$ is NP-complete.*

As in the more general case of live and bounded FRT-nets, weak ergodicity of the firing process is assured (and strong ergodicity for Markovian nets), and the vector of visit ratios can be computed in polynomial time from the net structure and the routing rates at conflicts, solving the system 2.8 presented in section 2.1.1.

A particular version of the *rank theorem* of structurally live and structurally bounded FRT-nets (cfr. theorem 2.1.1) for free choice nets can be stated:

Theorem 2.1.10 [CCS90a] *Let $\mathcal{N}$ be a strongly connected structurally bounded free choice net. Then the net is structurally live iff $rank(C) = m-1-(a-n)$, where C is the incidence matrix of $\mathcal{N}$, $m = |T|$, $n = |P|$, and a is the number of input arcs to transitions.*

The importance of this statement for free choice nets lies on the fact that several key results of free choice theory appear as corollaries. For example, the characterization of simultaneous structural liveness and structural boundedness in free choice nets is of polynomial complexity, therefore, from theorem 2.1.8.2, the next result follows:

Corollary 2.1.13 *Let $\langle \mathcal{N}, M_0 \rangle$ be a structurally bounded free choice net. Then it can be decided in polynomial time on the number of arcs of $\mathcal{N}$ if the marked net is live, checking the rank characterization for structural liveness (theorem 2.1.10) and if all P-components are marked (with the algebraic characterization of corollary 2.1.11).*

The following duality result follows also from theorem 2.1.10:

Corollary 2.1.14 *Let $\mathcal{N} = \langle P, T, Pre, Post \rangle$ be a free choice net. $\mathcal{N}$ is structurally live and structurally bounded iff the reverse-dual of $\mathcal{N}$, $\mathcal{N}_{rd} = \langle T, P, Post, Pre \rangle$, is structurally live and structurally bounded.*

Proof. If $\mathcal{N}$ is connected structurally live and structurally bounded then it is strongly connected, consistent, and conservative [CCS90d]. Then $\mathcal{N}_{rd}$ is strongly connected, consistent, and conservative, thus structurally bounded.

Finally, since $rank(C) = rank(C_{rd})$, $m_{rd} = n$, $n_{rd} = m$, and $a_{rd} = a$, we have $m_{rd} - 1 - (a_{rd} - n_{rd}) = n - 1 - (a - m) = m - 1 - (a - n)$, i.e., if $\mathcal{N}$ is structurally live then $\mathcal{N}_{rd}$ is also structurally live. ∎

Based on [BV84], W. Vogler proved in 1989 that a live and bounded free choice net has at least one home state.

Theorem 2.1.11 [Vog89] *Let $\langle \mathcal{N}, M_0 \rangle$ be a live and bounded free choice net. Then $\langle \mathcal{N}, M_0 \rangle$ has a home state.*

The importance of the previous result from the performance evaluation point of view is stated in the next corollary (see section 1.2.3.3):

Corollary 2.1.15 *Let $\langle \mathcal{N}, M_0 \rangle$ be a stochastic live and bounded free choice net. Then its marking process is weakly ergodic. Moreover, if the net is Markovian, its marking process is strongly ergodic.*

As in the case of marked graphs, for live and bounded free choice nets, it is possible to show that $SB(p) = B(p)$.

Theorem 2.1.12 [Esp90] *Let $\langle \mathcal{N}, M_0 \rangle$ be a live and bounded free choice net, then for all place p of $\mathcal{N}$: $B(p) = SB(p)$.*

In other words, the structural marking bound is always reached in a live and bounded free choice net, and the next result follows:

Corollary 2.1.16 *A live free choice net is bounded iff it is structurally bounded.*

The importance of the above results lies on the fact that marking bounds can be efficiently computed (looking for the structural ones) and, in particular, that boundedness can be algebraically characterized ($\exists Y \geq \mathbb{1}$ such that $Y^T \cdot C \leq 0$ [Mur89]).

Using theorem 2.1.12, an interesting property of live and bounded free choice nets, that allows an efficient computation of liveness bound of transitions, can be derived:

Theorem 2.1.13 *Let $\langle \mathcal{N}, M_0 \rangle$ be a live and bounded free choice net. Then, for all transition t of $\mathcal{N}$: $E(t) = L(t) = SE(t)$.*

Proof. Let t_i be a given transition of $\mathcal{N}$. A new live and bounded free choice net $\langle \mathcal{N}', M_0' \rangle$ is obtained by splitting transition t_i into a transition t_{i_1}, an unmarked place p_i, and another transition t_{i_2}. Then, for t_i and p_i: $SE(t_i) = SB(p_i)$ and $E(t_i) = B(p_i)$. Since for live and bounded free choice nets $B(p_i) = SB(p_i)$ (cfr. theorem 2.1.12) then $E(t_i) = SE(t_i)$.

Live and bounded free choice nets are structurally bounded (corollary 2.1.16) and live. Since structurally bounded nets are conservative [Sil85], the structural marking bound coincides with the bound obtained from a basis of P-semiflows: $SB(p_i) = \max\{M(p_i) \mid B^T \cdot M = B^T \cdot M_0', M \geq 0\}$ [CS89c].

Let M_h be a home state of $\langle \mathcal{N}, M_0 \rangle$ (its existence is guaranteed by theorem 2.1.11). Because M_h is reachable from M_0', $B^T \cdot M_h = B^T \cdot M_0'$. Considering as a new starting time that in which M_h is reached for the first time: $SB(p_i) = \max\{M(p_i) \mid B^T \cdot M = B^T \cdot M_h, M \geq 0\}$. Thus $SB(p_i)$ is reached from a home state, and $E(t_i) = L(t_i)$. ■

Now, from the previous theorem and taking into account that for any transition t the computation of the structural enabling bound $SE(t)$ can be formulated in terms of the problem (LPP1), the following monotonicity property of the liveness bound of a transition with respect to the initial marking is obtained:

Corollary 2.1.17 *If $\langle \mathcal{N}, M_0 \rangle$ is a live and bounded free choice net and $M_0' \geq M_0$ then the liveness bound of t in $\langle \mathcal{N}, M_0' \rangle$ is greater than or equal to the liveness bound of t in $\langle \mathcal{N}, M_0 \rangle$.*

The previous result appears to be a generalization (stated for the particular case of bounded nets) of the classical liveness monotonicity property for free choice nets stated in corollary 2.1.12.

Finally, let us define *state machines*, a well-known subclass of free choice nets:

Definition 2.1.12 (State machines) *State machines are ordinary Petri nets (i.e., pre- and post-incidence functions taking values in $\{0,1\}$) such that for all transition t: $|{}^\bullet t| = |t^\bullet| = 1$.*

State machines allow the modelling of decisions (conflicts) and reentrancy (when $\sum_{p\in P} M_0(p) \geq 2$) but not synchronization. They are the Petri net counterpart of monoclass queueing networks topology. We recall some well-known results from qualitative theory of state machines in the following property.

Property 2.1.16 [Sil85] *Let $\langle \mathcal{N}, M_0\rangle$ be a marked state machine. Then:*

1. *$\mathcal{N}$ is structurally bounded.*
2. *$\mathcal{N}$ is structurally live iff it is strongly connected.*
3. *$\langle \mathcal{N}, M_0\rangle$ is live iff $\mathcal{N}$ is strongly connected and $\sum_{p\in P} M_0(p) \geq 1$.*
4. *If $\langle \mathcal{N}, M_0\rangle$ is live, then it is k–bounded iff $\sum_{p\in P} M_0(p) = k$.*

2.1.4 FRT-nets communicating through buffers

In this section, *deterministic systems of sequential processes* are identified as a subclass of marked FRT-nets. In fact, the more general class of *systems of FRT-nets* communicating through private buffers belongs also to the FRT-nets class, thus being recursively defined.

Firstly we define systems of FRT-nets and later we focus our attention on the particular case of deterministic systems of sequential processes.

Definition 2.1.13 (Systems of FRT-nets) *A Petri net $\mathcal{N} = \langle P_1 \cup \ldots \cup P_s \cup B, T_1 \cup \ldots \cup T_s, Pre, Post\rangle$ is a system of FRT-nets iff:*

i) $P_i \cap P_j = \emptyset$, $T_i \cap T_j = \emptyset$, $P_i \cap B = \emptyset$, $i, j = 1, \ldots, s; i \neq j$,

ii) $\mathcal{N}_i = \langle P_i, T_i, Pre|_i, Post|_i\rangle$, $i = 1, \ldots, s$, *are FRT-nets (where $Pre|_i$ and $Post|_i$ are the restrictions of Pre and $Post$ to P_i and T_i), and*

iii) the set of buffers B is such that $\forall b \in B$:

a) ${}^\bullet b \neq \emptyset$, $b^\bullet \neq \emptyset$,

b) $\exists i, j \in \{1, \ldots, s\}, i \neq j$, *such that* ${}^\bullet b \subset T_i$ *and* $b^\bullet \subset T_j$, *and*

c) $\forall p \in P_1 \cup \ldots \cup P_s : p^\bullet \cap b^\bullet = \emptyset \vee p^\bullet \subseteq b^\bullet$.

We remark that the above defined systems of FRT-nets are composed by FRT-nets communicating by means of *private buffers* (condition iii.b of previous definition).

From the strong conditions imposed to the connection of buffers (iii.b and iii.c), the inclusion of the above class in FRT-nets can be easily deduced:

Property 2.1.17 *Systems of FRT-nets are also FRT-nets.*

In fact, since a given FRT-net can be seen also as a (trivial) system of FRT-nets (with an empty set of buffers), both net subclasses are the same. In other words, we obtain the possibility of getting a decomposed view of FRT-nets. This fact is useful from both a modelling (it allows a modular design of systems) and analysis (it provides *divide and conquer* techniques) point of views.

Therefore, all the results presented in section 2.1.1 for general FRT-nets can be applied to systems of such nets communicating through buffers. In particular, for structurally live and structurally bounded systems, the vector of visit ratios for transitions can be computed in polynomial time on the net size from the structure and routing rates at conflicts, using theorem 2.1.2.

In the next section, a particular subclass of marked systems of FRT-nets for which some interesting qualitative results can be derived are considered.

2.1.4.1 Deterministic systems of sequential processes

Deterministic systems of sequential processes [SB88] are used for the modelling and analysis of distributed systems composed by sequential processes communicating through private buffers. Each *sequential process* is modelled by a binary (1–marked) state machine. The communication among them is described by *buffers* (places) which contain *products/messages* (tokens) of some processes that are resources for others. Each buffer is *private* for two state machines, in the sense that it is an output place of only one machine and input place of the other (possibly the same). From a queueing network perspective, 1–bounded state machines represent “complex servers” while buffers represent queues (see figure 2.11 where grey places are buffers).

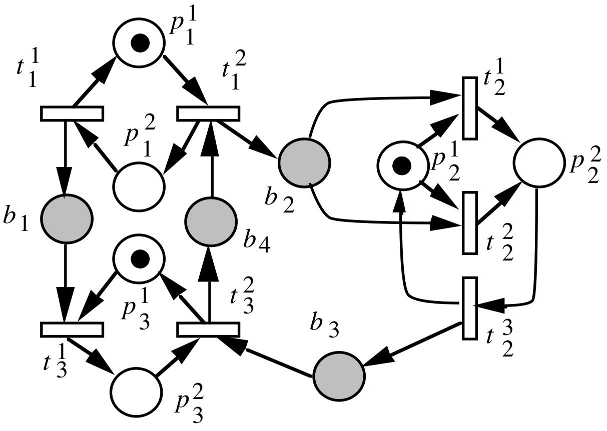


Figure 2.11: A deterministic system of sequential processes.

Definition 2.1.14 (Deterministic systems of sequential processes) [SB88] *A marked Petri net* $\langle \mathcal{N}, M_0 \rangle = \langle P_1 \cup \ldots \cup P_s \cup B, T_1 \cup \ldots \cup T_s, Pre, Post, M_0 \rangle$ *is a deterministic system of sequential processes iff:*

i) $P_i \cap P_j = \emptyset$, $T_i \cap T_j = \emptyset$, $P_i \cap B = \emptyset$, $i, j = 1, \ldots, s; i \neq j$,

ii) $\langle \mathcal{N}_i, M_0|_i \rangle = \langle P_i, T_i, Pre|_i, Post|_i, M_0|_i \rangle$, $i = 1, \ldots, s$, *are live and 1–bounded state machines (where* $Pre|_i$,$Post|_i$, *and* $M_0|_i$ *are the restrictions of* Pre, Post, *and* M_0 *to* P_i *and* T_i*), and*

iii) the set of buffers B *is such that* $\forall b \in B$:

a) ${}^\bullet b \neq \emptyset$, $b^\bullet \neq \emptyset$,

b) $\exists i, j \in \{1, \ldots, s\}, i \neq j$, *such that* ${}^\bullet b \subset T_i$ *and* $b^\bullet \subset T_j$, *and*

c) $\forall p \in P_1 \cup \ldots \cup P_s : p^\bullet \cap b^\bullet = \emptyset \vee p^\bullet \subseteq b^\bullet$.

Trivially, such systems are a subclass of marked systems of FRT-nets (hence marked FRT-nets), thus the vector of visit ratios can be computed by the application of theorem 2.1.2. However, we present here a *modular technique* for the computation of the visit ratios: the relative throughput among transitions of each state machine can be *computed separately*, and then conveniently *scaled* because of the influence of buffers.

Theorem 2.1.14 *Let* $\langle \mathcal{N}, M_0\rangle = \langle P_1 \cup \ldots \cup P_s \cup B, T_1 \cup \ldots \cup T_s, Pre, Post, M_0\rangle$ *be a live and bounded deterministic system of sequential processes. The visit ratios for transitions can be computed in polynomial time on the net size using the following algorithm:*

Step 1. *For each* $i = 1, \ldots, s$*, compute the visit ratios* $\vec{u}^{(i_1)}$ *for transitions of* T_i*, solving the system (2.8) of theorem 2.1.2 for the state machine* $\langle \mathcal{N}_i, M_0|_i\rangle = \langle P_i, T_i, Pre|_i, Post|_i, M_0|_i\rangle$*.*

Step 2. *For each* $i = 1, \ldots, s$*, substitute the state machine* $\langle \mathcal{N}_i, M_0|_i\rangle$ *of the system by a single transition* t_i*, with pre- and post-incidence functions given by:* $Pre(b, t_i) = \sum_{t\in T_i} u_t^{(i_1)} Pre(b, t)$ *and* $Post(b, t_i) = \sum_{t\in T_i} u_t^{(i_1)} Post(b, t)$*, for all* $b \in B$*.*

Step 3. *The net resulting from Step 2 is structurally decision-free, thus mono-T-semiflow. Compute the visit ratios* $\vec{w}^{(1)}$ *for transitions of this net, according to theorem 2.1.5.*

Step 4. *The visit ratios for transitions of the whole net are given by:* $v_t^{(1_1)} = w_i^{(1)} u_t^{(i_1)}$*, for each* $t \in T_i$*, for all* T_i*.*

Proof. In Step 1, the relative throughput of transitions of each state machine can be computed in isolation, and it is equal to the one computed in non-isolation, by requirement iii.c of definition 2.1.14 (i.e., buffers do not disturb the decisions taken by state machines).

In Step 2, the pre- and post-incidence functions of the new transitions which substitute the state machines must preserve the relative throughput among the whole state machines. This means that if a state machine $\langle \mathcal{N}_i, M_0|_i\rangle$ needs $\sum_{t\in T_i} u_t^{(i_1)} Pre(b, t)$ tokens from buffer b for the firing of a sequence with firing count vector equal to the visit ratio $\vec{u}^{(i_1)}$, then the pre-incidence function from buffer b to the new transition t_i must be $Pre(b, t_i) = \sum_{t\in T_i} u_t^{(i_1)} Pre(b, t)$. And the same for the post-incidence function.

The net resulting from Step 2 is structurally decision-free because the buffers of the original system are private (requirement iii.b of definition 2.1.14).

Finally, in step 4 the visit ratios of separate state machines are scaled because of the influence of buffers. ■

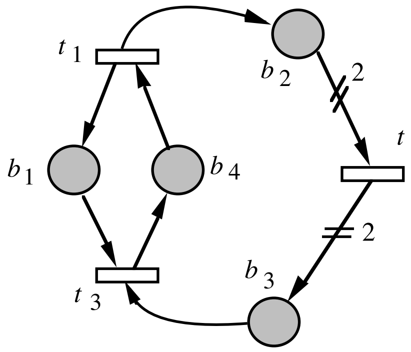


Figure 2.12: Substitution of state machines by transitions in the net of figure 2.11.

We remark that the previous *divide and conquer* technique can be applied for general systems of structurally live and structurally bounded FRT-nets communicating through buffers.

As an example, let us consider the net depicted in figure 2.11. For transitions t_1^1 and t_1^2 the visit ratios in the isolated state machine are:

$$\vec{u}^{(1_1)} = (1, 1)^T \tag{2.17}$$

If the conflict at place p_2^1 is solved with equal rate in favour of t_2^1 and t_2^2 (i.e., with probabilities 1/2 and 1/2), the visit ratios for transitions of the isolated second state machine are:

$$\vec{u}^{(2_1)} = (1, 1, 2)^T \tag{2.18}$$

And for the third state machine in isolation:

$$\vec{u}^{(3_1)} = (1, 1)^T \tag{2.19}$$

The next step (according to theorem 2.1.14) consists of substituting the state machines for transitions, as depicted in figure 2.12. Now, the visit ratios for transitions t_1, t_2, t_3 of the net in figure 2.12, which are given by its T-semiflow, are:

$$\vec{w}^{(2)} = (2, 1, 2) \tag{2.20}$$

Finally, the vector of visit ratios for the transitions in the whole net can be easily derived from (2.17), (2.18), (2.19), and (2.20):

$$\vec{v}^{(2_1)} = (2, 2, 1, 1, 2, 2, 2)^T \tag{2.21}$$

In [SB88] a characterization for structural liveness of deterministic systems of sequential processes is derived. We recall this result after the definition of the *small ring* concept.

Definition 2.1.15 [SB88] *Let $\langle \mathcal{N}, M_0 \rangle$ be a deterministic system of sequential processes. The string $Z = b_0\mathcal{M}_0 b_1\mathcal{M}_1 \ldots b_{k-1}\mathcal{M}_{k-1}$ is called a small ring of $\mathcal{N}$ iff:*

a) $\forall \mathcal{M}_i, \mathcal{M}_j$, $i, j = 0, \ldots, k-1$ *state machines of* $\mathcal{N}$*:* $i \neq j \Rightarrow \mathcal{M}_i \neq \mathcal{M}_j$*.*

b) $b_i \in {}^\bullet T_i$ *and* $b_{i+1} \in T_i^\bullet$, $i = 0, \ldots, k-1$ *($i+1$ is modulo k).*

c) $\displaystyle\prod_{i=1}^{k} \min \left\{ \frac{\sum_{t \in Y} Post(b_i, t)}{\sum_{t \in Y} Pre(b_{i+1}, t)} \mid Y \text{ simple circuit of } \mathcal{M}_i \right\} < 1.$

Theorem 2.1.15 [SB88] *Let $\langle \mathcal{N}, M_0 \rangle$ be a deterministic system of sequential processes. Then $\mathcal{N}$ is structurally live iff it does not contain a small ring.*

The following monotonicity property can be found also in the work of Y. Souissi et N. Beldiceanu:

Property 2.1.18 [SB88] *Let $\langle \mathcal{N}, M_0 \rangle$ be a live deterministic system of sequential processes. If the initial marking of buffers increases, the obtained marked net is also live.*

2.1.4.2 Totally open deterministic systems of sequential processes

The special subclass of *totally open* deterministic systems of sequential processes is introduced now. Its interest lies on the fact that some qualitative and quantitative properties can be derived, that do not hold for general (non-totally open) systems. In particular, necessary and sufficient conditions for the existency of an exponential timing making

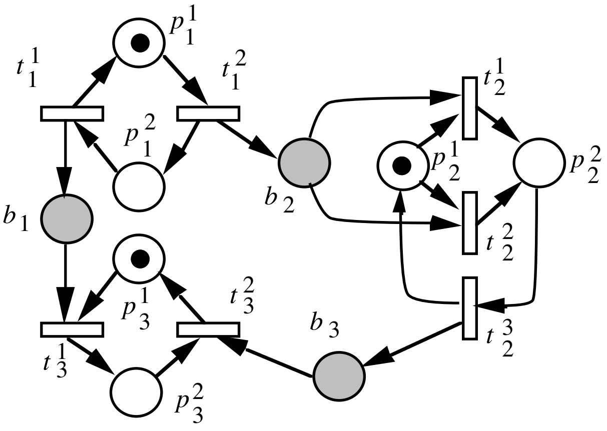


Figure 2.13: A totally open deterministic system of sequential processes.

ergodic such systems are derived. Moreover, in chapter 5 we prove that the ergodicity characterization and the exact computation of steady-state performance measures is possible in polynomial time on the net size for these nets (assuming exponential timing).

Definition 2.1.16 (Totally open deterministic systems of sequential processes) [CS89a] *A deterministic system of sequential processes is called totally open iff the underlying net has not any circuit containing buffers.*

An example of totally open deterministic system of sequential processes is depicted in figure 2.13.

Some interesting qualitative results can be derived from the structure of these nets. Liveness of totally open deterministic systems of sequential processes and unboundedness of the buffers are presented in theorem 2.1.16. In theorem 2.1.17, consistency (necessary condition for marking ergodicity of live Markovian nets, cfr. theorem 1.2.3) is shown to collapse with existence of home state for this subclass of nets.

Theorem 2.1.16 *Let $\langle \mathcal{N}, M_0 \rangle$ be a totally open deterministic system of sequential processes. Then $\langle \mathcal{N}, M_0 \rangle$ is live and all buffers are unbounded.*

Proof. Let be $\langle \mathcal{N}, M_0 \rangle = \langle P_1 \cup \ldots \cup P_s \cup B, T_1 \cup \ldots \cup T_s, Pre, Post, M_0 \rangle$. All $\langle \mathcal{N}_i, M_0|_i \rangle = \langle P_i, T_i, Pre|_i, Post|_i, M_0|_i \rangle$ are live in isolation (by property 2.1.16, because M_0 marks all the state machines by definition 2.1.14.ii). All transitions of those state machines without input buffers can be fired an infinite number of times, independently of the rest, so all the output buffers of these machines do not restrict the firing of the other machines. This argument can be repeated for all the system because of the absence of circuits containing buffers. Thus, the net is live.

From the liveness of the system and from the fact that buffers are not contained in any circuit, the input transitions of buffers can be fired an infinite number of times without firing their output transitions. Thus, all buffers are unbounded. ■

An interesting property of live marked graphs, presented in theorem 2.1.7, that states a bridge between its behavioural and structural analysis is that all potentially reachable markings are reachable. It is also true for totally open deterministic systems of sequential processes:

Property 2.1.19 *Let $\langle \mathcal{N}, M_0 \rangle$ be a totally open deterministic system of sequential processes. Then $M \in R(\mathcal{N}, M_0)$ iff $M \in PR(\mathcal{N}, M_0)$. In other words, each vector $\vec{\sigma} \in \mathbb{N}^m$ such that $M_0 + C \cdot \vec{\sigma} \geq 0$ corresponds at least to one firable sequence in $\mathcal{N}$ from M_0.*

Proof. Let $\vec{\sigma} \in \mathbb{N}^m$ be such that $M_0 + C \cdot \vec{\sigma} \geq 0$. All transitions represented in $\vec{\sigma}$ belonging to state machines without input buffers $(\mathcal{N}_{i_1}, \ldots, \mathcal{N}_{i_r})$ are firable at first. Then, transitions belonging to state machines whose input buffers are output of $\mathcal{N}_{i_1}, \ldots, \mathcal{N}_{i_r}$ can be fired. This procedure can be repeated for all state machines since no circuits containing buffers exist. ■

The following theorem relates, for totally open deterministic systems of sequential processes, a behavioural property (existence of home state) with a structural one (consistency).

Theorem 2.1.17 *Let $\langle \mathcal{N}, M_0 \rangle$ be a totally open deterministic system of sequential processes. Then $\mathcal{N}$ is consistent iff M_0 is a home state.*

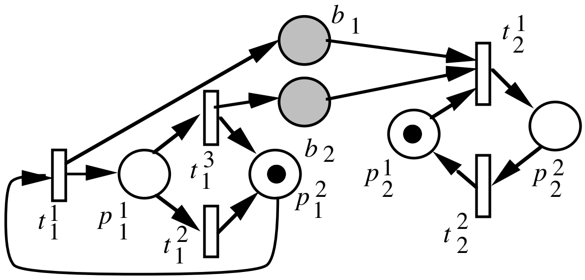
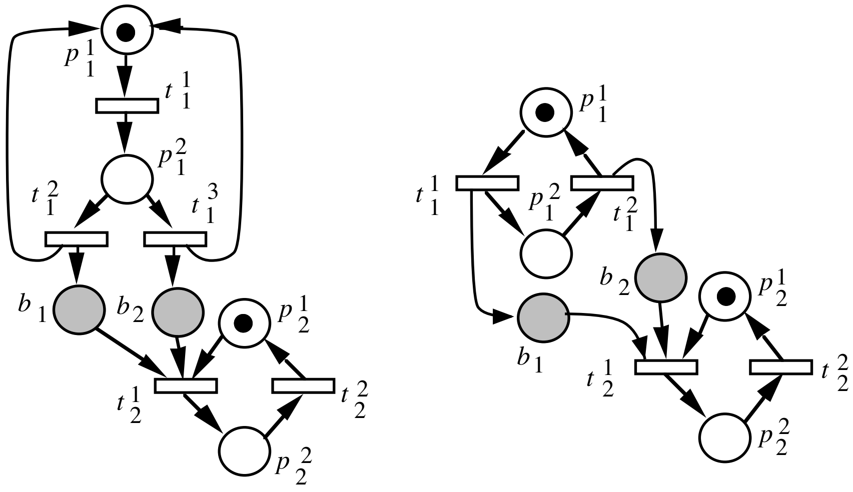


Figure 2.14: A non-consistent totally open deterministic system of sequential processes.

Proof. Let us suppose that there exists $X \in (\mathbb{N}^+)^m$ such that $C \cdot X = 0$ (i.e., the net is consistent). Let $M \in R(\mathcal{N}, M_0)$ and σ such that $M_0[\sigma\rangle M$. Let $k \in \mathbb{N}$ be such that $kX \geq \vec{\sigma}$. Then, $\vec{\delta} = kX - \vec{\sigma} \geq 0$, $M_0[\sigma\rangle M[\delta\rangle M_0$ (fireability of δ is deduced from property 2.1.19) and M_0 is a home state.

Let us suppose that M_0 is a home state. Since the net is live (cfr. theorem 2.1.16), there exist a marking M_1 and a firing sequence σ_1 including all transitions such that $M_0[\sigma_1\rangle M_1$. Since M_0 is a home state, there exists a firing sequence σ_2 such that $M_1[\sigma_2\rangle M_0$. Then the vector $\vec{\sigma_1} + \vec{\sigma_2} \in (\mathbb{N}^+)^m$ is such that $C \cdot (\vec{\sigma_1} + \vec{\sigma_2}) = 0$, where C is the incidence matrix of the net. Therefore, $\mathcal{N}$ is consistent. ∎

In theorem 1.2.3, a necessary condition for the marking ergodicity of a live Markovian Petri net is shown. Now, let us remark that there exist non-consistent totally open deterministic systems of sequential processes (see figure 2.14: $M(b_1) - M(b_2) = \vec{\sigma}(t_1^1) - \vec{\sigma}(t_1^3) = \vec{\sigma}(t_1^1) - [\vec{\sigma}(t_1^1) - \vec{\sigma}(t_1^2) - M(p_1^1)] = M(p_1^1) + \vec{\sigma}(t_1^2)$. Since the net is live, $\vec{\sigma}(t_1^2) \rightarrow \infty \Rightarrow M(b_1) - M(b_2) \rightarrow \infty \Rightarrow$ structurally marking non-ergodic net). Then, in practice, it is convenient to check consistency (a polynomial time computation) of the underlying net before computing marking ergodicity conditions for a given Markovian interpretation of the totally open deterministic system of sequential processes. Taking into account the above remark and theorem 1.2.3, the following result with practical interest can be stated:



(a) Structurally non-ergodic: $({}^\bullet b_1, {}^\bullet b_2) \notin$ SDR; $(b_1{}^\bullet, b_2{}^\bullet) \in$ SDR.

(b) Potentially ergodic: $({}^\bullet b_1, {}^\bullet b_2) \in$ SDR; $(b_1{}^\bullet, b_2{}^\bullet) \in$ SDR.

Figure 2.15: Consistent totally open deterministic systems of sequential processes with two state machines and two buffers.

Corollary 2.1.18 *There exist totally open deterministic systems of sequential processes that are marking non-ergodic for all timing interpretation. In particular, non-consistent systems are always marking non-ergodic.*

Unfortunately, it cannot be stated that if $\langle \mathcal{N}, M_0 \rangle$ is a consistent totally open deterministic system of sequential processes, there exists a Markovian interpretation such that the stochastic net is marking ergodic. The net in figure 2.15.a is consistent but there does not exist any Markovian interpretation making it marking ergodic: the case of (exponential) distribution rates $\lambda_1^2 = \lambda_1^3$ (of course, only possible in theory!) leads to a null recurrent Markov process and so non-ergodic, because the marking process at buffers b_1 and b_2 can be shown to be isomorphic to a symmetrical random walk [Rev84].

The rest of this section is devoted to the study of necessary and sufficient conditions for the "potential marking ergodicity" of systems. We say that a net is potentially marking ergodic iff there exists a Markovian interpretation (i.e., an assignment of exponential random timing) that can lead to marking ergodic systems.

For characterizing the possible existence of a Markovian interpretation making marking ergodic a given totally open deterministic system of sequential processes, let us give local rules that will be composed step by step for a large system.

As a first step, a necessary and sufficient condition for a deterministic system of two sequential processes to be potentially marking ergodic in terms of consistency of the net and of some *synchronic distance relations* among transitions is presented.

After that, a "transitivity rule" for systems composed by three state machines is presented. It gives a necessary and sufficient condition for such systems to be potentially marking ergodic.

An iterative application of the presented rules leads to the derivation of necessary and sufficient conditions for a general totally open deterministic system of sequential processes to be potentially marking ergodic.

Let us now recall the concept of *global synchronic distance relation.* If two subsets of transitions are in global synchronic distance relation then it is not possible to fire an infinite number of times some transition of the first subset without firing any transition of the second subset, and vice versa. Even more, if two subsets of transitions are in global synchronic distance relation they behave like if they were included in a regulation circuit (see definition 2.1.6). Global synchronic distance relation is used below for finding necessary and sufficient conditions for the existence of a Markovian interpretation that makes marking ergodic a totally open deterministic system of sequential processes.

Definition 2.1.17 (Global synchronic distance relation) [Sil87] *Let $\langle \mathcal{N}, M_0\rangle$ be a Petri net and T_1, T_2 subsets of transitions. T_1 and T_2 are in global synchronic distance relation, denoted as $(T_1, T_2) \in SDR$, iff $\exists W_1, W_2 \in \mathbb{N}^m$ vectors which express the weights associated with the transitions of the subsets T_1 and T_2 (i.e., $||W_1|| = T_1$ and $||W_2|| = T_2$), and $\exists k \in \mathbb{N}$ such that*

$$\sup_{\substack{\sigma\in L(\mathcal{N},M)\\ M\in R(\mathcal{N},M_0)}} |(W_1 - W_2)^T \cdot \vec{\sigma}| \leq k \tag{2.22}$$

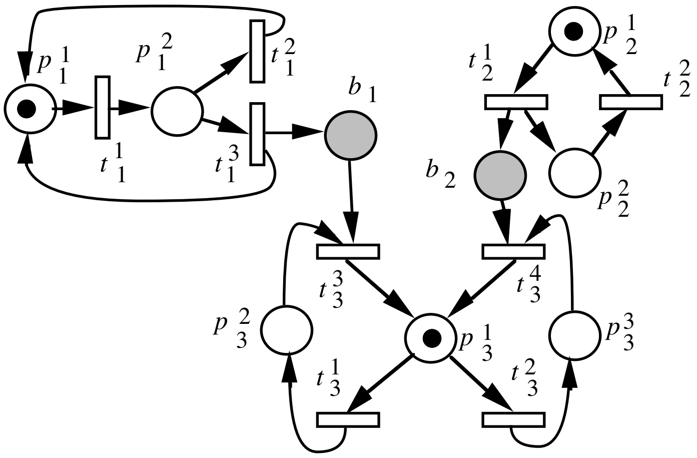


Figure 2.16: Structurally marking non-ergodic system with three state machines.

The first result is a negative one. If a given state machine receives tokens from two different state machines, one of them without input buffers, the system cannot be marking ergodic (see figure 2.16).

Theorem 2.1.18 *Let* $\langle \mathcal{N}, M_0\rangle = \langle P_1 \cup \ldots \cup P_s \cup B, T_1 \cup \ldots \cup T_s, Pre, Post, M_0\rangle$ *be a totally open deterministic system of sequential processes such that for one of their communicating state machines* $\langle \mathcal{N}_i, M_0|_i\rangle = \langle P_i, T_i, Pre|_i, Post|_i, M_0|_i\rangle$*:*

a) $\exists b_1$ *such that* $b_1^\bullet \subseteq T_i$ *(i.e., it is an input buffer of the machine* $\langle \mathcal{N}_i, M_0|_i\rangle$*) and* ${}^\bullet b_1 \subseteq T_j$*, where* T_j *is the set of transitions of another state machine* $\langle \mathcal{N}_j, M_0|_j\rangle$ *such that* $\nexists b \in B$ *satisfying* $b^\bullet \subseteq T_j$ *(i.e., the input state machine of buffer* b_1 *has not input buffers), and*

b) $\exists b_2$ *such that* $b_2^\bullet \subseteq T_i$ *(i.e., another input buffer of the machine* $\langle \mathcal{N}_i, M_0|_i\rangle$*) and* ${}^\bullet b_2 \not\subseteq T_j$ *(i.e., the input state machine of buffer* b_2 *is not* $\langle \mathcal{N}_j, M_0|_j\rangle$*).*

Then, there is no Markovian interpretation such that the corresponding stochastic net is marking ergodic.

Proof. The arrival processes of tokens to buffers b_1 and b_2 are Poisson-like independent stochastic processes [Ros83] joint by a state machine.

Then, the underlying Markov chain is transient (in the case in which the marking of one buffer tends to infinity with time) or null recurrent (case os stochastic equilibrium, equivalent to a symmetrical random walk) but never positive recurrent. ■

Now, let us give necessary and sufficient conditions for the existence of a Markovian timing interpretation that makes marking ergodic a system composed by two state machines (see figures 2.14 and 2.15). Basically, the net must be consistent and for each pair of buffers between both state machines, the input (output) transitions of one buffer cannot fire an infinite number of times without firing the input (output) transitions of the other buffer. In this way, null recurrency of the associated Markov process is discarded.

Theorem 2.1.19 *Let $\langle \mathcal{N}, M_0\rangle = \langle P_1 \cup P_2 \cup B, T_1 \cup T_2, Pre, Post, M_0\rangle$ be a totally open deterministic system of sequential processes composed by two state machines and a set of buffers B such that $\forall b \in B : {}^\bullet b \subseteq T_1, b^\bullet \subseteq T_2$. Then, there exists a Markovian interpretation making marking ergodic the system if and only if:*

i) $\mathcal{N}$ is consistent and

ii) $\forall b_i, b_j \in B$: $({}^\bullet b_i, {}^\bullet b_j) \in SDR$ and $(b_i^\bullet, b_j^\bullet) \in SDR$.

Proof. Let us suppose that there exists a Markovian timing making marking ergodic the system. Then the net is consistent by theorem 1.2.3. If $({}^\bullet b_1, {}^\bullet b_2) \notin SDR$ or $(b_1^\bullet, b_2^\bullet) \notin SDR$ then the marking of b_1 and b_2 cannot be linearly expressed the one in function of the other. These buffers have two non-equal arrival rates, joint by a unique server (the state machine). Thus, the underlying Markov chain is transient or null recurrent, but never positive recurrent. Therefore, it is marking non-ergodic.

Now, let us suppose that (i) and (ii) hold. Let us consider $b_1, b_2 \in B$. Let us denote ${}^\bullet b_1 = T_{11}$, ${}^\bullet b_2 = T_{12}$, $b_1^\bullet = T_{21}$, and $b_2^\bullet = T_{22}$. Since $(T_{11}, T_{12}) \in SDR$ and $(T_{21}, T_{22}) \in SDR$, there exist vectors $W_{11}, W_{12}, W_{21}, W_{22} \in \mathbb{N}^m$ with $||W_{ij}|| = T_{ij}, i, j = 1, 2$ (see definition 2.1.17) such that two regulation circuits can be added without changing the behaviour of the net, as follows (see figure 2.17):

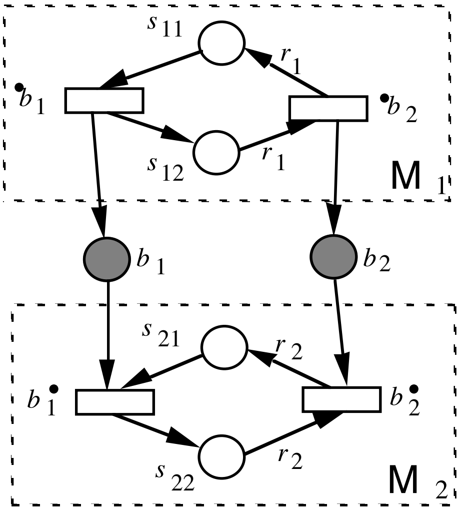


Figure 2.17: Regulation circuits between transitions in global synchronic distance relation.

$P_1' = P_1 \cup \{s_{11}, s_{12}\}$ with $s_{11}^\bullet = {}^\bullet s_{12} = T_{11}$, ${}^\bullet s_{11} = s_{12}^\bullet = T_{12}$, and $Pre(s_{11}, t) = Post(s_{12}, t) = W_{11}(t), \forall t \in T_{11}$, $Pre(s_{12}, t) = Post(s_{11}, t) = W_{12}(t), \forall t \in T_{12}$.

$P_2' = P_2 \cup \{s_{21}, s_{22}\}$ with $s_{21}^\bullet = {}^\bullet s_{22} = T_{21}$, ${}^\bullet s_{21} = s_{22}^\bullet = T_{22}$, and $Pre(s_{21}, t) = Post(s_{22}, t) = W_{21}(t), \forall t \in T_{21}$, $Pre(s_{22}, t) = Post(s_{21}, t) = W_{22}(t), \forall t \in T_{22}$.

Now, from consistency of the net: $\exists X \geq \mathbb{1}$ such that $C \cdot X = 0$. Then, the column vectors of the incidence matrix (of the modified net) corresponding with transitions T_{11}, T_{12}, T_{21}, and T_{22} must be linearly independent, or equivalently: $W_{11} = W_{21}$ and $W_{12} = W_{22}$. This implies that the markings of both buffers are linearly independent. The argument above can be applied to all pair of buffers of the net. Then, the marking of all of them can be expressed in terms of the marking of one buffer and the marking of the state machines. Then, a Markov timing can be associated such that the interarrival times of tokens to the buffers are greater than the "service times" (mean cycle times of the output state machines, in isolation). ■

Note that in the case of totally open deterministic systems of sequential processes composed by two state machines, if (i) and (ii) of theorem 2.1.19 hold then the marking of all the buffers can be always computed from the marking of one buffer and the marking of the state machines. With the object of computing ergodicity conditions for a larger system including $\mathcal{N}$ as a subsystem, if (i) and (ii) hold, from the performance point of view, we can suppose without loss of generality that the two state machines are communicating with at most one buffer.

Let us now give the "transitivity rule" for three state machines communicating with buffers like in figure 2.13. This rule completes the stating of necessary and sufficient conditions for the existence of a Markovian timing that makes marking ergodic a given totally open deterministic system of sequential processes.

Theorem 2.1.20 *Let* $\langle \mathcal{N}, M_0 \rangle = \langle P_1 \cup P_2 \cup P_3 \cup \{b_1, b_2, b_3\}, T_1 \cup T_2 \cup T_3, Pre, Post, M_0 \rangle$ *be a totally open deterministic system of sequential processes composed by three state machines and three buffers such that* ${}^\bullet b_1 \subseteq T_1, b_1^\bullet \subseteq T_3$, ${}^\bullet b_2 \subseteq T_1, b_2^\bullet \subseteq T_2$, ${}^\bullet b_3 \subseteq T_2$, *and* $b_3^\bullet \subseteq T_3$. *Then, there exists a Markovian interpretation making marking ergodic the system iff:*

i) $\mathcal{N}$ *is consistent and*

ii) $({}^\bullet b_1, {}^\bullet b_2) \in SDR$, $(b_2^\bullet, {}^\bullet b_3) \in SDR$, $(b_1^\bullet, b_3^\bullet) \in SDR$.

Proof. If $({}^\bullet b_1, {}^\bullet b_2) \notin SDR$ or $(b_2^\bullet, {}^\bullet b_3) \notin SDR$ or $(b_1^\bullet, b_3^\bullet) \notin SDR$ then the marking of b_1 and b_3 cannot be linearly expressed the one in function of the other. Then, these buffers have non-equal arrival rates, joint by a unique server (the state machine). Thus, the underlying Markov chain is transient or null recurrent, but never positive recurrent. Therefore, it is marking non-ergodic.

Now, let us suppose that (i) and (ii) hold. $(b_2^\bullet, {}^\bullet b_3) \in SDR$ implies that b_2, $\langle \mathcal{N}_2, M_0|_2 \rangle = \langle P_2, T_2, Pre|_2, Post|_2, M_0|_2 \rangle$, and b_3 can be substituted by a unique buffer without changing the behaviour of $\langle \mathcal{N}_1, M_0|_1 \rangle = \langle P_1, T_1, Pre|_1, Post|_1, M_0|_1 \rangle$ and $\langle \mathcal{N}_3, M_0|_3 \rangle = \langle P_3, T_3, Pre|_3, Post|_3, M_0|_3 \rangle$. Then, if the net is consistent, $({}^\bullet b_1, {}^\bullet b_2) \in SDR$ and $(b_1^\bullet, b_3^\bullet) \in SDR$, and theorem 2.1.19 can be applied. ■

If (i) and (ii) of theorem 2.1.20 hold, then the marking of b_3 can be always computed from the marking of b_1, b_2 and the marking of the state machines. As an example, let as consider the system depicted in figure 2.13. It verifies conditions (i) and (ii) of theorem 2.1.20. And it can be easily checked that: $M(b_3) = M(b_1) + M(p_1^2) + M(p_3^2) - M(b_2) - M(p_2^1)$, for all marking M, reachable from the initial marking.

With the object of computing conditions for a larger system including $\mathcal{N}$ as a subsystem, if (i) and (ii) hold, the state machine $\mathcal{M}_2$ and the buffers b_2, b_3 can be substituted by a unique buffer.

Theorems 2.1.19 and 2.1.20 provide rules for an iterative reduction of buffers of a totally open deterministic system of sequential processes. These rules preserve the possibility of existence of a Markovian timing that makes the system marking ergodic if the necessary and sufficient conditions (stated in the mentioned theorems) hold.

Therefore the existence of a Markovian timing that makes marking ergodic a totally open deterministic system of sequential processes is characterized in terms of pure structural conditions that can be checked in polynomial time: consistency and some global synchronic distance relations.

2.2 Persistent nets and behaviourally extended free choice nets

Persistent nets [LR78] and *behaviourally extended free choice nets* (or “réseaux à choix non-imposé” [Bra83]) are recalled in this section as behaviourally defined net subclasses for which some reachability analysis is needed for the computation of the vector of visit ratios for transitions. Therefore, visit ratios do depend not only on the structure and routing but also on the initial marking.

2.2.1 Persistent nets

Persistent nets [LR78] constitute a behaviourally characterized subclass of Petri nets which has a common property with live and bounded mono-T-semiflow nets: all their consistent firing count vectors are proportional to a unique vector, which is the unique minimal T-semiflow

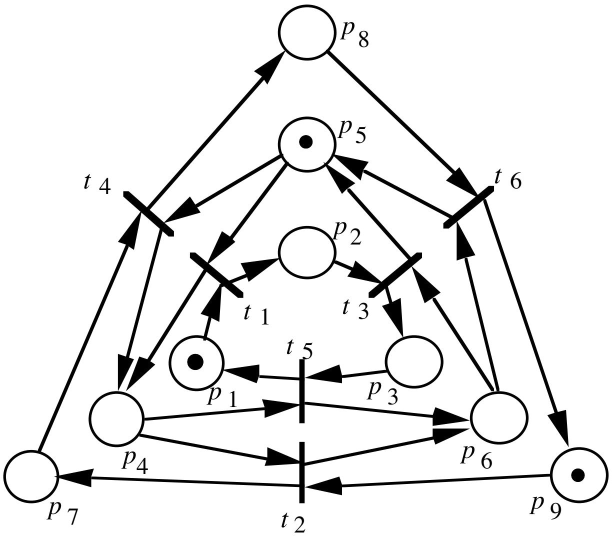


Figure 2.18: Persistent net.

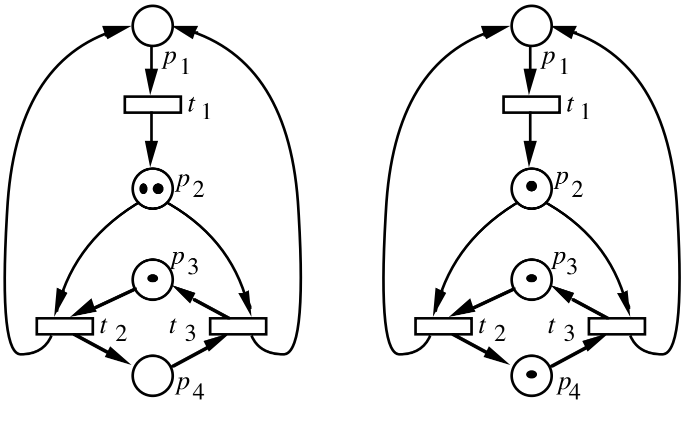
in the case of mono-T-semiflow nets [CCS89,CCS91].

Definition 2.2.1 (Persistent nets) [LR78] *A marked net $\langle \mathcal{N}, M_0 \rangle$ is said to be persistent iff for all reachable marking M and for all different transitions, t_1 and t_2, enabled in M, the sequences t_1t_2 and t_2t_1 are firable from M.*

As an example look at the net in figure 2.18. This net has structural conflicts (e.g., p_4 has two output transitions, t_2 and t_5) but for the initial marking $M_0 = (1, 0, 0, 0, 1, 0, 0, 0, 1)^T$ no state can be reached in which a decision must be taken. Persistency is a behavioural property. However, we have not found in the literature any result about theoretical complexity of deciding persistency.

The same net structure with a different initial marking can give non-persistent behaviour. For example, the net in figure 2.19.a is persistent, but that in figure 2.19.b (with the same structure) is not.

Definition 2.2.2 (Nets with unique consistent firing count vector) [CCS89,CCS91] *A marked net $\langle \mathcal{N}, M_0 \rangle$ has a unique consistent firing count vector $\vec{\sigma}_R$ associated with all marking repetitive sequences*



(a) Persistent net. (b) Non-persistent net.

Figure 2.19: Persistent and non-persistent nets with the same structure.

iff for all markings M reachable from M_0 such that $M[\sigma\rangle M$, there exists $k \in \mathbb{N}$ such that $\vec{\sigma} = k\vec{\sigma}_R$.

Note that the vector $\vec{\sigma}$ in the above definition is a (possibly non-elementary) T-semiflow because $M = M + C \cdot \vec{\sigma}$, thus $C \cdot \vec{\sigma} = 0$.

Obviously, live and bounded mono-T-semiflow nets have a unique consistent firing count vector, the minimal T-semiflow, which can be structurally characterized.

Let us recall a property and two results that will lead to the conclusions that persistent nets have a unique consistent firing count vector and that the firing process associated with a bounded persistent net is weakly ergodic. The property is that of *directedness*:

Definition 2.2.3 (Directedness property) [Bra83] *A marked net $\langle \mathcal{N}, M_0 \rangle$ possesses the directedness property iff $\forall M, M' \in R(\mathcal{N}, M_0) : R(\mathcal{N}, M) \cap R(\mathcal{N}, M') \neq \emptyset$.*

This means that if a net possesses the directedness property, any two reachable markings have at least one common successor marking.

Lemma 2.2.1 [Bra83] *All persistent nets have the directedness property.*

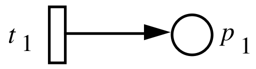


Figure 2.20: An unbounded live persistent net having the directedness property but without home states.

Lemma 2.2.2 [BV84] *For bounded marked nets, directedness and the existence of a home state are equivalent properties.*

Corollary 2.2.1 *Bounded persistent nets have home states.*

Figure 2.20 illustrates an example that shows that the above lemma does not hold for unbounded nets: the net depicted there is unbounded, it has the directedness property, and has not home state.

Using the previous lemmatas, the following statement can be derived:

Theorem 2.2.1 *Live bounded persistent connected nets without implicit places have a unique consistent firing count vector.*

Proof. Consider a live bounded persistent connected net $\langle \mathcal{N}, M_0 \rangle$. Bounded persistent nets have home states (corollary2.2.1), so that let M be a home state of the net. Since the net is live, there exist at least one firing sequence σ such that $M[\sigma\rangle M$ and $\vec{\sigma} \geq \mathbb{1}$. Now assume that there exist two different firing count vectors $\vec{\sigma_1}$ and $\vec{\sigma_2}$ such that $M[\sigma_1\rangle M$, $M[\sigma_2\rangle M$, and $\vec{\sigma_1}, \vec{\sigma_2} \geq \mathbb{1}$. Then, there must exist three transitions t_i, t_j, and t_k such that $t_i \in ||\vec{\sigma_1}||$, $t_i \notin ||\vec{\sigma_2}||$, $t_j \in ||\vec{\sigma_2}||$, $t_j \notin ||\vec{\sigma_1}||$, $t_k \in ||\vec{\sigma_1}||$, and $t_k \in ||\vec{\sigma_2}||$. Moreover, since the net is connected and each firing count vector is a consistent component, there must be a structural conflict between the two transitions t_i and t_j, i.e., $\exists p \in t_k^\bullet$ such that $p \in {}^\bullet t_i \cap {}^\bullet t_j$. Since the net is persistent, the structural conflict between t_i and t_j cannot be effective, and the two sequences σ_1 and σ_2 are firable independently one of the other, so that the shared place p must be implicit. ■

Since live bounded persistent connected nets have a unique consistent firing count vector and home states, the following result can be obtained:

Theorem 2.2.2 *Let $\langle \mathcal{N}, M_0 \rangle$ be a live bounded connected marked net without implicit places. If $\langle \mathcal{N}, M_0 \rangle$ is persistent then:*

1. *Both the marking and the firing processes are weakly ergodic.*
2. *The vector of visit ratios for transitions is proportional to the unique consistent firing count vector.*
3. *If $\langle \mathcal{N}, M_0 \rangle$ is Markovian, its marking and firing processes are also strongly ergodic.*

Let us now introduce a subclass of persistent nets such that the persistency is inherent to the structure.

Definition 2.2.4 (Structurally persistent nets) [LR78] *A net $\mathcal{N}$ is said to be structurally persistent iff $\langle \mathcal{N}, M_0 \rangle$ is persistent for all finite initial marking M_0.*

Structurally decision-free nets (definition 2.1.9) are trivially included in with structurally persistent nets:

Property 2.2.1 *If a net is structurally decision-free then it is structurally persistent.*

The converse of the above property is not true. As an example, the net depicted in figure 2.21 is structurally persistent but it is not structurally decision-free. Nevertheless, if self-loops are discarded (*pure* nets), structurally persistent nets are also structurally decision-free:

Property 2.2.2 [LR78] *If a net is structurally persistent, then either it is structurally decision-free or for each place p with more than one output transition, such transitions are on self-loop with p.*

2.2.2 Behaviourally extended free choice nets

In a similar way than persistent nets (behaviourally characterized) have a common property with live mono-T-semiflow nets (structurally characterized), a behaviourally counterpart of extended free choice nets can be defined. These nets are called *behaviourally extended free choice* or "réseaux à choix non imposé", in French [Bra83].

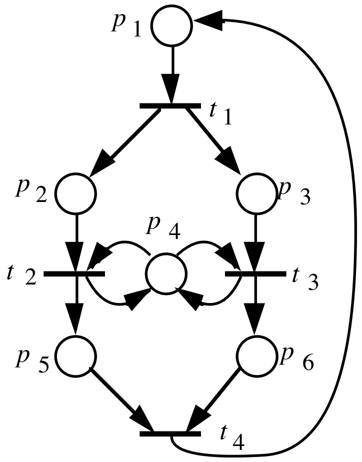


Figure 2.21: Structurally persistent but non-structurally decision-free net.

For these nets structural asymmetric and/or structural non-transitive conflicts [Bes87] can exist, violating the extended free choice assumption, but a marking allowing this kind of non-free choice is never reached from the initial one.

Definition 2.2.5 (Behaviourally extended free choice nets) *A marked Petri net $\langle \mathcal{N}, M_0 \rangle$ is said to be behaviourally extended free choice iff $\forall t, t' \in T$ such that ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$, $\forall M \in R(\mathcal{N}, M_0)$: $M[t\rangle \Longleftrightarrow M[t'\rangle$.*

For live and bounded behaviourally extended free choice nets, the vector of visit ratios can be computed in an analogous way than for live and bounded extended free choice nets, from the net structure and from the routing rates at (effectively free) conflicts. Nevertheless, a previous behavioural analysis based on the reachability graph is needed for concluding about the “choix non imposé” assumption (the same that occurs for persistent nets).

2.3 Conclusions

In this chapter we have recalled or introduced those net subclasses whose vector of visit ratios for transitions can be computed independently of the average service times of transitions. The first section was dedicated to structurally defined nets. FRT-nets have been characterized as nets whose visit ratios can be efficiently derived from net structure and routing rates at conflicts (if structural liveness and structural boundedness is assumed), solving a linear system of equations. An alternative recursive definition has been presented with interesting consequences: modularity of modelling and analysis, by means of communication through private buffers of FRT-subnets. Some qualitative properties interesting per se or from a performance point of view have been derived.

Well-known Petri net subclasses have been recalled and identified as particular classes of FRT-nets (marked graphs, structurally decision-free nets, state machines, free choice nets, deterministic systems of sequential processes), and other subclasses with interesting properties have been introduced, such as mono-T-semiflow nets.

Finally, some behavioural extensions have been considered. Persistent nets and behaviourally extended free choice nets are net subclasses whose vector of visit ratios for transitions can be computed independently of the average service times, after a previous behavioural analysis, if liveness and boundedness are assumed.

Chapter 3

Bounds for strongly connected marked graphs

In this chapter, we obtain *upper and lower bounds* on the steady-state performance of marked graphs [CCCS89,CCCS90], a well-known subclass of Petri nets (see definition 2.1.10) that allow only concurrency and synchronization but no choice. In particular we derive bounds for the throughput of transitions (see definition 1.2.5), defined as the average number of firings per time unit (or its inverse, that we call the *mean cycle time* of transitions). From this quantity, applying Little's formula [Lit61] it is possible to derive other average performance estimates of the model. Under these restrictions we will show results that can be computed in polynomial time on the size of the net model, and that depend only on the mean values and not on the higher moments of the probability distribution functions of the random variables that describe the timing of the system. The independence of the probability distribution can be viewed as a useful generalization of the performance results, since higher moments of the service delays are usually unknown for real cases, and difficult to estimate and assess. Moreover we show that both upper and lower bounds, computed by means of proper linear programming problems, are tight, in the sense that for any marked graph model it is possible to define families of stochastic timings such that the steady-state performances of the timed Petri net models are arbitrarily close to either bound.

Figure 3.1 depicts an example of a live and 1–bounded marked

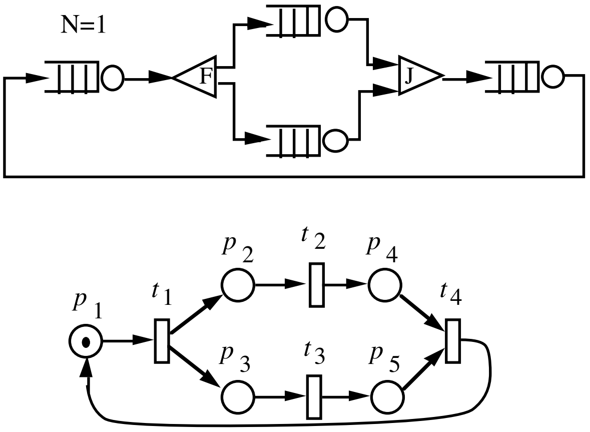


Figure 3.1: Example of a 1–bounded marked graph and its synchronized queueing network counterpart.

graph. In the same figure the equivalent representation in terms of queueing network with synchronization primitives [SMK82] is also depicted. According to figure 3.1, Petri net places correspond with queues, while net's transitions represent servers and synchronization constraints. It is easily seen that only sum and "max" operators are needed to compute the performance: indeed the actual cycle time in this example is the random variable $\gamma = \tau_1 + \max\{\tau_2, \tau_3\} + \tau_4$ (where τ_i denotes the enabling time of transition t_i, or its service time, with queueing networks terminology), therefore the mean cycle time is

$$\Gamma = E[\gamma] = E[\tau_1] + E[\max\{\tau_2, \tau_3\}] + E[\tau_4] = s_1 + E[\max\{\tau_2, \tau_3\}] + s_4 \tag{3.1}$$

where s_i denotes the average enabling time of transition t_i, i.e., its average service time. Cohen et al. developed a special algebra to formalize the properties of this kind of models in the deterministic case [CMQV89]. F. Baccelli et al. extended this approach to the stochastic case [BM89,BBW89].

Our idea is that of computing fast bounds for the throughput of transitions based only on the knowledge of the first moments of probability distribution functions. This can be intuitively explained as follows. The sum is independent of the probability distribution (for linearity); since for non-negative variables $x_i \leq \max_i\{x_i\} \leq \sum_i x_i$, $E[\max_i\{x_i\}]$ can be bounded by $\max_i\{E[x_i]\} \leq E[\max_i\{x_i\}] \leq \sum_i E[x_i]$. Therefore for the net in figure 3.1 we can write:

$$s_1 + \max\{s_2, s_3\} + s_4 \leq \Gamma \leq s_1 + s_2 + s_3 + s_4 \tag{3.2}$$

In this chapter, we show how linear programming problems based on the incidence matrix of the underlying Petri net structure can be solved to compute this kind of bounds for marked graphs. In section 3.1, we focus our attention on throughput upper bounds for strongly connected marked graphs. Applying Little's formula [Lit61] to each place of the net and using structural information taken from P-semiflows, a linear programming problem is derived whose optimum solution (which can be computed in polynomial time) is a lower bound for the mean cycle time of transitions (inverse of the average throughput). Moreover, this bound is shown to be reachable for arbitrary net structure, initial marking, and mean and variance for transition service times. From the linear programming form of the computed bound, some interesting results are derived.

A tight lower bound for the steady-state throughput (upper bound for the mean cycle time) is obtained in polynomial time in section 3.2, from the knowledge of the given average service times and the liveness bounds of transitions, which are computed by solving proper linear programming problems. This bound cannot be improved unless more information from the service times of transitions than their mean values are used.

The case of non-strongly connected (i.e., unbounded) marked graphs is considered in section 3.3. For these nets, the exact throughput of transitions can be derived from the knowledge of the exact throughput of the isolated strongly connected components. Since we are able to compute bounds for the throughput of the isolated strongly connected components, bounds for the whole net can be obtained. Finally, in section 3.4, some concluding remarks are presented.

3.1 Upper bound for the steady-state throughput

In this section, upper bounds on throughput for strongly connected (and thus structurally bounded, by property 2.1.13) marked graphs are presented. We remark that strong connectivity of a graph is a well-known problem of polynomial time complexity.

3.1.1 Little's law and P-semiflows

Three of the most significant performance measures for a closed region r of a network in the analysis of queueing systems are related by Little's formula [Lit61], which holds under very general (i.e., weak) conditions:

$$Q_r = X_r R_r \tag{3.3}$$

Q_r is the average number of customers in the region, X_r is the output rate (throughput) from the region (which is equal to the input rate), and R_r is the average time spent by a customer within the region.

Now, Little's result is applied to each place of a weakly ergodic net. Denoting as $\overline{M}(p_i)$ the limit average number of tokens at place p_i, $\overline{X}$ the limit vector of transition throughputs (see definition 1.2.5), and $\overline{R}(p_i)$ the average time spent by a token within the place p_i (average response time at place p_i), the above mentioned relationship is stated as follows (see [FN85a]):

$$\overline{M}(p_i) = (PRE[p_i] \cdot \overline{X})\overline{R}(p_i) \tag{3.4}$$

where $PRE[p_i]$ is the i^{th} row of the pre-incidence matrix of the underlying Petri net, thus $PRE[p_i] \cdot \overline{X}$ is the output rate of place p_i.

In the study of computer systems, Little's law is frequently used when two of the related quantities are known and the third one is needed. This is not exactly the case here. Now, $\overline{R}(p_i)$ and $\overline{M}(p_i)$ are unknown. On the other hand, the vector of visit ratios

$$\vec{v}^{(j)} = \frac{1}{\overline{X}(t_j)}\overline{X} = \Gamma_{(j)}\overline{X} \tag{3.5}$$

normalized for having the j^{th} component equal 1, can be easily computed for important net subclasses (see chapter 2) and, in particular, for live marked graphs. $\Gamma_{(j)}$ is called *mean cycle time* of transition t_j (inverse of its average throughput).

The average response times at places $\overline{R}(p_i)$ are unknown. In fact, they can be expressed as sums of the average *waiting* times due to the synchronization schemes and the average *service* times associated with transitions, and only average service times are known: s_i, $i = 1, \ldots, m$. Thus the average response times can be *lowerly bounded* from the knowledge of the average service times, and the following system of inequalities can be derived from (3.4):

$$\Gamma_{(j)}\overline{M} \geq PRE \cdot \vec{D}^{(j)} \tag{3.6}$$

where $\vec{D}^{(j)}$ is the vector with components $D_i^{(j)} = v_i^{(j)} s_i$, average service demand (or *loading*) for each transition t_i of the net, that is the average total service that a token demands from transition t_i in all its visits to it. The superscript "$^{(j)}$" indicates that the vector is normalized for having the j^{th} component $D_j^{(j)}$ equal to s_j (i.e., $v_i^{(j)} = 1$).

Since marked graphs are consistent nets and their unique minimal T-semiflow is $\mathbb{1}$, we have $\vec{v}^{(j)} = \mathbb{1} = \vec{v}$ for all transition t_j (cfr. property 2.1.12), thus for all $j = 1, \ldots, m$, $\Gamma_{(j)} = \Gamma$, $\vec{D}^{(j)} = \vec{D} = \vec{s}$ (where $\vec{s}$ denotes the vector with components s_i, $i = 1, \ldots, m$), and

$$\Gamma\ \overline{M} \geq PRE \cdot \vec{s} \tag{3.7}$$

From this inequality, a lower bound Γ^{min} for the mean cycle time of transitions can be derived. We take into account that Γ^{min} must be such that inequality (3.7) holds and for some place p_i the equality is reached:

$$\Gamma^{min} = \frac{PRE[p_i] \cdot \vec{s}}{\overline{M}(p_i)} \tag{3.8}$$

Since the vector $\overline{M}$ is unknown, (3.8) cannot be solved. However, the following structural marking invariant can be written using a P-semiflow Y:

$$Y^T \cdot M_0 = Y^T \cdot M = Y^T \cdot \overline{M},\ \forall M_0 \in \mathbb{N}^n,\ \forall M \in R(\mathcal{N}, M_0) \tag{3.9}$$

Now, from (3.7) and (3.9):

$$\Gamma(Y^T \cdot M_0) \geq Y^T \cdot PRE \cdot \vec{s} \tag{3.10}$$

And a lower bound for the mean cycle time in steady-state is:

$$\Gamma^{min} = \max_{Y \in \{P-semiflow\}} \frac{Y^T \cdot PRE \cdot \vec{s}}{Y^T \cdot M_0} \tag{3.11}$$

Of course, an upper bound for the throughput of transitions is $1/\Gamma^{min}$.

Let us formulate the previous lower bound for the mean cycle time in terms of a particular class of optimization problems called *fractional programming problems* [Mur83]:

$$\begin{aligned} \Gamma^{min} = \quad & \text{maximize} \quad \frac{Y^T \cdot PRE \cdot \vec{s}}{Y^T \cdot M_0} \\ & \text{subject to} \quad Y^T \cdot C = 0 \\ & \qquad\qquad\quad \mathbb{1}^T \cdot Y > 0 \\ & \qquad\qquad\quad Y \geq 0 \end{aligned} \tag{3.12}$$

The above problem can be rewritten as follows:

$$\begin{aligned} \Gamma^{min} = \quad & \text{maximize} \quad \frac{Y^T \cdot PRE \cdot \vec{s}}{q} \\ & \text{subject to} \quad Y^T \cdot C = 0 \\ & \qquad\qquad\quad \mathbb{1}^T \cdot Y > 0 \\ & \qquad\qquad\quad Y^T \cdot M_0 = q \\ & \qquad\qquad\quad Y \geq 0 \end{aligned} \tag{3.13}$$

Then, because $Y^T \cdot M_0 > 0$ (guaranteed for live marked graphs, by corollary 2.1.7), we can change $\frac{Y}{q}$ by Y and obtain the linear programming formulation stated in the next theorem (in which $\mathbb{1}^T \cdot Y > 0$ is removed because $Y^T \cdot M_0 = 1$ implies $\mathbb{1}^T \cdot Y > 0$):

Theorem 3.1.1 *A lower bound for the mean cycle time for live strongly connected marked graphs can be obtained by solving the following linear programming problem:*

$$\begin{array}{rll} \Gamma^{min} = & \textit{maximize} & Y^T \cdot PRE \cdot \vec{s} \\ & \textit{subject to} & Y^T \cdot C = 0 \\ & & Y^T \cdot M_0 = 1 \\ & & Y \geq 0 \end{array} \qquad \text{(LPP3)}$$

The following theorem concerns a special class of optimum solutions of (LPP3) that will be used later in the interpretation of this linear programming problem: the *minimal* P-semiflows. Firstly, we present a lemma that will be used in the proof of the theorem.

Lemma 3.1.1 [MS82] *Let $\mathcal{N}$ be a Petri net and C its incidence matrix. A P-semiflow Y of $\mathcal{N}$ is minimal iff the cardinal of its support is one unit higher than the rank of the submatrix made up of the rows l_i of C such that $Y(i)$ is not zero.*

In order to prove the theorem, we use the concept of *basic feasible solution* from linear programming [Mur83], and the problem (LPP3) rewritten in the following way:

$$\begin{array}{rll} \Gamma^{min} = & \text{maximize} & Y^T \cdot PRE \cdot \vec{s} \\ & \text{subject to} & Y^T \cdot [C|M_0] = (0|1) \\ & & Y \geq 0 \end{array} \qquad \text{(LPP4)}$$

Let $\mathcal{Y}$ be the set of feasible solutions of (LPP4). If $Y \in \mathcal{Y}$, the set of row vectors of $A = [C|M_0]$ that Y uses is $\{A[j] \mid j \text{ is such that } Y[j] > 0\}$. The feasible solution $Y \in \mathcal{Y}$ is said to be a basic feasible solution for (LPP4) iff the set of row vectors of A that Y uses is a linearly independent set.

Theorem 3.1.2 *Under the conditions of theorem 3.1.1, if (LPP3) has an optimum solution, then it has an optimum solution which is a minimal P-semiflow.*

Proof. Taking into account [Mur83, theorem 3.3], if (LPP4) has an optimum feasible solution, then it has a basic feasible solution Y that is optimum. Therefore, the set of rows that are used by Y is linearly independent (i.e., full rank). Considering that $Y^T \cdot C = 0$, the number of non-null entries of vector Y (i.e., the number of rows used by Y) is equal to the rank of rows of C used by Y plus one. This last statement is precisely the characterization of a minimal P-semiflow, presented in lemma 3.1.1. ■

It is well-known that the *simplex* method for the solution of linear programming problems gives good results in practice, even if it has exponential worst case complexity. In any case, an algorithm of polynomial worst case complexity can be found in [Kar84].

Theorem 3.1.1 shows that the problem of finding an upper bound for the steady-state throughput (lower bound for the mean cycle time) in a strongly connected stochastic marked graph can be solved looking at the mean cycle time associated with each P-semiflow (circuits for marked graphs, see theorem 2.1.6) of the net, considered in isolation. These cycle times can be computed making the summation of the average enabling times of all the transitions involved in the P-semiflow (service time of the whole circuit), and dividing by the number of tokens present in it (customers in the circuit).

3.1.2 Reachability of the upper bound

The above bound that holds for any stochastic interpretation, happens to be the same that has been obtained for strongly connected deterministically timed marked graphs by other authors (see for example [Ram74,RH80]), but here it is considered in a practical linear programming form. For deterministically timed nets, the reachability of this bound has been shown [Ram74,RH80]. Since deterministic timing is just a particular case of stochastic timing, the reachability of the bound is assured for our purposes as well. Even more, the next result shows that the previous bound cannot be improved only on the base of the knowledge of the coefficients of variation for the transition service times.

Theorem 3.1.3 *For live strongly connected marked graphs with arbitrary values of mean and variance for transition service times, the* *lower bound for the mean cycle time obtained from (LPP3) cannot be improved.*

Proof. We know from [Ram74] that for deterministic timing the bound is reached. Only "max" and sum operators are needed to compute the cycle time. Therefore we must construct a family of random variables with arbitrary means and variances behaving in the limit like deterministic timing for both operators (max and sum).

This is the case for the following family of random variables, for varying values of the parameter $\alpha \in [0, 1)$:

$$X_{s_i,\sigma_i}(\alpha) = \begin{cases} s_i\alpha & \text{with probability } 1-\epsilon_i \\ s_i(\alpha + \frac{1-\alpha}{\epsilon_i}) & \text{with probability } \epsilon_i \end{cases} \tag{3.14}$$

where

$$\epsilon_i = \frac{s_i^2(1-\alpha)^2}{s_i^2(1-\alpha)^2 + \sigma_i^2} \tag{3.15}$$

These variables are such that $E[X_{s_i,\sigma_i}(\alpha)] = s_i$, $Var[X_{s_i,\sigma_i}(\alpha)] = \sigma_i^2$, and they verify:

$$\lim_{\alpha\to 1} E[\max\{X_{s_i,\sigma_i}(\alpha), X_{s_j,\sigma_j}(\alpha)\}] = \max\{s_i, s_j\} \tag{3.16}$$

and, of course, for all α such that $0 \le \alpha < 1$: $E[X_{s_i,\sigma_i}(\alpha) + X_{s_j,\sigma_j}(\alpha)] = s_i + s_j$.

Then, if random variables $X_{s_i,\sigma_i}(\alpha)$ are associated with transitions t_i, $i = 1, \ldots, m$, taking α closer to 1, the mean cycle time tends to the bound given by (LPP3). ■

A polynomial computation of the minimal cycle time for deterministically timed strongly connected marked graphs was proposed in [Mag84], solving the following linear programming problem:

$$\begin{array}{rll} \Gamma^{min} = & \text{minimize} & \gamma \\ & \text{subject to} & -C \cdot z + \gamma M_0 \ge POST \cdot \vec{s} \\ & & \gamma,\ z \ge 0 \end{array} \tag{LPP5}$$

To investigate the relationship between (LPP3) and (LPP5) let us consider the *dual* problem [Mur83] of (LPP5):

$$\Gamma^{min} = \begin{array}{ll} \text{maximize} & Y^T \cdot POST \cdot \vec{s} \\ \text{subject to} & Y^T \cdot C \leq 0 \\ & Y^T \cdot M_0 \leq 1 \\ & Y \geq 0 \end{array} \tag{LPP6}$$

Since strongly connected marked graphs are conservative (property 2.1.13), there does not exist $Y \geq 0$ such that $Y^T \cdot C \not\leq 0$ and then the constraint $Y^T \cdot C \leq 0$ of (LPP6) becomes $Y^T \cdot C = 0$ (i.e., the constraint of (LPP3)). For all Y such that $Y^T \cdot C = 0$: $Y^T \cdot POST = Y^T \cdot PRE$. For live marked graphs, $\forall Y \in \mathbb{N}^n$, $Y \neq 0$ such that $Y^T \cdot C = 0$ then $Y^T \cdot M_0 \geq 1$ (corollary 2.1.7). Thus the constraint $Y^T \cdot M_0 \leq 1$ of (LPP6) becomes $Y^T \cdot M_0 = 1$ for live nets (i.e., the constraint of (LPP3)).

Hence for live strongly connected marked graphs, the problem (LPP3) is equivalent to (LPP5) formulated in [Mag84] for deterministic systems.

3.1.3 Interpretation and derived results

Linear programming problems give an easy way to derive results and interpret them. Just looking at the objective function of the problem (LPP3) the following monotonicity property is obtained: the lower bound for the mean cycle time does not increase if $\vec{s}$ decreases or if M_0 increases.

Property 3.1.1 *Let $\langle \mathcal{N}, M_0 \rangle$ be a live strongly connected marked graph and $\vec{s}$ the vector of average service times.*

1. *For a fixed $\vec{s}$, if $M_0' \geq M_0$ (i.e., increasing the number of initial resources) then the lower bound for the mean cycle time of $\langle \mathcal{N}, M_0', \vec{s} \rangle$ is less than or equal to the one of $\langle \mathcal{N}, M_0, \vec{s} \rangle$ (i.e., $\Gamma^{min'} \leq \Gamma^{min}$).*
2. *For a fixed M_0, if $\vec{s'} \leq \vec{s}$ (i.e., for faster resources) then the lower bound for the mean cycle time of $\langle \mathcal{N}, M_0, \vec{s'} \rangle$ is less than or equal to the one of $\langle \mathcal{N}, M_0, \vec{s} \rangle$ (i.e., $\Gamma^{min'} \leq \Gamma^{min}$).*

The next property is strongly related to the reversibility of live marked graphs.

Property 3.1.2 *For any live strongly connected marked graph $\langle \mathcal{N}, M_0 \rangle$, the bound obtained with the problem (LPP3) does not change for any marking reachable from M_0.*

Proof. Let us consider the lower bound for the mean cycle time for a marking $M \in R(\mathcal{N}, M_0) \Leftrightarrow M = M_0 + C \cdot \vec{\sigma} \geq 0$ for some $\vec{\sigma} \geq 0$ in terms of a linear programming problem:

$$\begin{array}{rll} \Gamma^{min} = & \text{maximize} & Y^T \cdot PRE \cdot \vec{s} \\ & \text{subject to} & Y^T \cdot C = 0 \\ & & Y^T \cdot M = 1 \\ & & M = M_0 + C \cdot \vec{\sigma} \\ & & Y,\ M,\ \vec{\sigma} \geq 0 \end{array} \tag{LPP7}$$

Since $Y^T \cdot M = Y^T \cdot M_0$, this problem is equivalent to:

$$\begin{array}{rll} \Gamma^{min} = & \text{maximize} & Y^T \cdot PRE \cdot \vec{s} \\ & \text{subject to} & Y^T \cdot C = 0 \\ & & Y^T \cdot M_0 = 1 \\ & & M = M_0 + C \cdot \vec{\sigma} \\ & & Y,\ M,\ \vec{\sigma} \geq 0 \end{array} \tag{LPP8}$$

Because the restrictions $M = M_0 + C \cdot \vec{\sigma}$, $M \geq 0$ and $\vec{\sigma} \geq 0$ do not affect the solution, they can be removed without changing the optimum of this problem with respect to (LPP3). ■

Since the upper bound on throughput is computed based on the total mean service time of elementary cycles and on the marking contained in them, it is easy to prove that the *reverse net* of $\mathcal{N} = \langle P, T, Pre, Post \rangle$ defined as $\mathcal{N}^{-1} = \langle P, T, Post, Pre \rangle$ yields the same bound in case of strongly connected marked graphs.

Property 3.1.3 *Let $\mathcal{N}$ be a strongly connected marked graph and $\mathcal{N}^{-1}$ its reverse net. Then, the upper bounds on throughput obtained for both nets with the problem (LPP3) are the same.*

In particular, if deterministic timing is considered (since the bound gives the exact throughput in this case), a reversibility property for the exact throughput follows. An analogous result under non-deterministic assumption is presented in [DLT90].

The next is a characterization of liveness for marked graphs in terms of the finiteness of the mean cycle time.

Theorem 3.1.4 *Let $\langle \mathcal{N}, M_0 \rangle$ be a strongly connected marked graph. $\langle \mathcal{N}, M_0 \rangle$ is live iff the value Γ^{min} given by theorem 3.1.1 is finite.*

Proof. For strongly connected marked graphs, the optimum value of (LPP3) is a lower bound for the mean cycle time. If this optimum value is infinite the mean cycle time is unbounded, and the net is dead. If the optimum value of (LPP3) is finite, since it is reachable for deterministic [Ram74] as well as for some stochastic (cfr. theorem 3.1.3) timing, the net must be deadlock-free. We know that for strongly connected marked graphs, liveness and deadlock-freeness are equivalent. Thus the finiteness of the optimum value of (LPP3) is sufficient to establish the liveness of a strongly connected marked graph. ■

The reader can interpret the previous result by noticing that the only way to obtain an infinite optimum solution for the problem (3.12) is for a solution of the system $Y \gneq 0, Y^T \cdot C = 0, Y^T \cdot M_0 = 0$, and the existence of such solution is a characterization of non-liveness (cfr. corollary 2.1.7).

3.2 Lower bound for the steady-state throughput

In this section, we present the computation of lower bounds on throughput for strongly connected marked graphs. We start by presenting a reachable lower bound for 1–live marked graphs (i.e., marked graphs with liveness bound for all transitions equal 1), and then we extend the result to bounded marked graphs. Finally we propose a polynomial complexity computation based on linear programming.

3.2.1 Basic result for 1–live marked graphs

A trivial lower bound for the steady-state performance of a live marked graph is, of course, given by the inverse of the sum of the service times of all the transitions. Since the net is live all transitions must be firable, and the sum of all service times corresponds to any complete sequentialization of all the activities represented in the model. This lower bound is always reached in a marked graph consisting of a single loop of transitions and containing a single token in one of the places, independently of the higher moments of the probability distribution functions (this observation can be trivially confirmed by the computation of the throughput upper bound, which in this case gives the same value).

To improve this trivial lower bound let us first consider the case of 1–live marked graphs (i.e., strongly connected marked graphs in which $L(t) = 1$ for all transition t, see definition 1.2.2). Of course live and 1–bounded marked graphs are guaranteed to be 1–live, but the result that we are going to present apply to more general cases. If we specify only the mean values of the transition service times and not the higher moments, we may always find stochastic models whose steady-state throughput is arbitrarily close to the trivial lower bound, independently of the topology of the marked graph (only provided that it is 1–live). Let us give a formal proof of this (somewhat counter-intuitive) result.

We define the family of random variables:

$$x^i_\mu(\epsilon) = \begin{cases} 0 & \text{with probability } 1 - \epsilon^i \\ \dfrac{\mu}{\epsilon^i} & \text{with probability } \epsilon^i \end{cases} \tag{3.17}$$

for $\mu \geq 0$; $0 < \epsilon \leq 1$; $i \in \mathbb{N}$.

It is straightforward to see that $E[x^i_\mu(\epsilon)] = \mu$, and $E[(x^i_\mu(\epsilon))^2] = \mu^2/\epsilon^i$. This implies that the coefficient of variation is 0 for $\epsilon = 1$, and that it tends to ∞ as $\epsilon \to 0$ provided that $i > 0$.

Theorem 3.2.1 *For any live and 1–bounded marked graph with a specification of the average service time s_j for each transition t_j it is possible to assign probability distribution functions to the transition service times such that the mean cycle time is $\Gamma = \sum_j s_j - O(\epsilon)$, $\forall\ 0 < \epsilon \leq 1$, independently of the topology of the net (and thus independently of the* *potential maximum degree of parallelism intrinsic in the marked graph). (We use here the notation* $O(f(x))$ *to indicate any function* $g(x)$ *such that* $\lim_{x\to 0}\frac{g(x)}{f(x)} \le k \in \mathbb{R}$.*)*

Proof. By construction, we will show that the association of the family of random variables $x^{j-1}_{s_j}(\epsilon)$ with each transition $t_j \in T$ yields exactly the mean cycle time Γ claimed by the theorem. To give the proof we will consider a sequence of models ordered by the index of transitions, in which the q^{th} model of the sequence has transitions $t_1, t_2, \ldots, t_q$ timed with the random variables $x^{j-1}_{s_j}(\epsilon)$, and all other transitions immediate (firing in zero time); the $|T|^{th}$ model in the sequence represents an example of reachability of the lower bound on throughput, independent of the net topology. Now we will prove by induction that the q^{th} model in the sequence has a mean cycle time

$$\Gamma_q = \sum_{j=1}^{q} s_j - O(\epsilon) \tag{3.18}$$

Base: $q = 1$: trivial since the repetitive cycle that constitute the steady-state behaviour of the marked graph contains only one (single-server) deterministic transition with average service time $\Gamma_1 = s_1$.

Induction step: $q > 1$: taking the limit $\epsilon \to 0$, the newly timed transition t_q will fire most of the times with time zero, thus normally not contributing to the computation of the mean cycle time, that will be just

$$\Gamma_{q-1} = \sum_{j=1}^{q-1} s_j - O(\epsilon) \tag{3.19}$$

(as in the case of model $q-1$) with probability $1-\epsilon^{q-1}$. On the other hand, the newly timed transition has a (very small) probability ϵ^{q-1} of delaying its firing of a time s_q/ϵ^{q-1}, which is at least order of $1/\epsilon$ bigger than any other firing time in the cycle, so that in this case all other transitions will wait for the firing of t_q after having completed their possible current firings in a time which is $O(\epsilon)$ lower than the firing time of t_q itself (i.e., $s_q/\epsilon^{q-1} = \Gamma_{q-1}/O(\epsilon)$). Therefore we obtain that

$$\Gamma_q = (1-\epsilon^{q-1})\Gamma_{q-1} + \epsilon^{q-1}\left(\frac{s_q}{\epsilon^{q-1}} - O(\epsilon)\right) = \sum_{j=1}^{q} s_j - O(\epsilon). \;\blacksquare \tag{3.20}$$

3.2.2 Extension to bounded marked graphs

Until now we have shown that the trivial sum of the average service times of all transitions in the net constitutes a tight (reachable) lower bound for the performance of a live and 1–bounded marked graph (or more generally of a 1–live strongly connected marked graph, but otherwise independently of the topology) in which only the mean values and neither the probability distribution functions nor the higher moments are specified for the transition service times. Let us now extend this result to the more general case of k–live strongly connected marked graphs.

Let us remember that for marked graphs $E(t) = L(t) = SE(t)$ (property 2.1.14); thus, the liveness bound of transitions can be efficiently computed using (LPP1). Now, an intuitive idea is to derive a lower bound on throughput for marked graphs containing transitions with liveness bound $k \geq 1$ by taking the algorithm used for the computation of the throughput upper bound in the case of k–bounded marked graphs, and substitute in it the “max” operator with the sum of the service times of all transitions involved. After some manipulation to avoid counting more than once the contribution of the same transition, one can arrive at the formulation of the following value for the maximum cycle time:

$$\Gamma^{max} = \sum_{j=1}^{m} \frac{s_j}{L(t_j)} \tag{3.21}$$

The proof of this result requires the following lemma.

Lemma 3.2.1 *Any strongly connected marked graph with arbitrary initial marking can be constraint to contain a main cycle including all transitions, without changing their liveness bound. This main cycle (which is not unique) contains a number of tokens equal to the maximum of the liveness bounds among all transitions. In addition there are other minor cycles that preserve the liveness bounds for transitions with bound lower than the maximum.*

The idea behind this constraint is to introduce a structural sequentialization among all transitions, thus potentially reducing the degree of concurrency among the activities modelled by the transitions. In other words, from the partial order given by the initial marked graph structure we try to derive a total order without changing the liveness bounds.

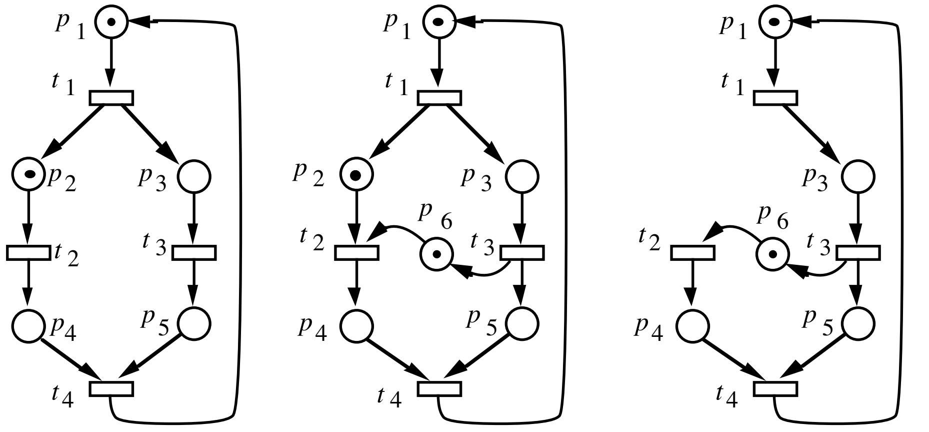
Proof of lemma 3.2.1. To construct a marked graph of the desired form we can apply the following iterative procedure that interleaves two non-disjoint cycles into a single one. Since the marked graph is strongly connected, each node belongs to at least one cycle; moreover, since the original marked graph is finite and each cycle cannot contain the same node more than once, this cycle interleaving procedure must terminate after a finite number of iterations. To reduce the number of cycles, implicit places created after each iteration can be removed. The iteration step is the following:

Step 1. Take two arbitrary non-disjoint cycles (unless the marked graph already contains a main cycle including all nodes, there always exists such a pair of cycles because the marked graph is strongly connected).

Step 2. Combine them in a single cycle in such a way that the partial order among transitions given by the two original cycles is substituted by a compatible but otherwise arbitrary total order. This combination can be obtained by adding new places that are connected as input for a transition of one cycle and output for a transition of the other cycle that we decide must follow in the sequence determined by the new cycle we are creating.

Step 3. Mark the new places added in such a way that the new cycle contains the same number of tokens as the maximum of the number of tokens in the two original cycles.

The above procedure is applied iteratively until all transitions are constrained into a single main cycle. At this point we can identify and eliminate the implicit places that have been created during the cycles interleaving procedure. We obtain then a marked graph composed by one main cycle containing $N_M = \max_{t \in T} L(t)$ tokens that connects all transitions, and a certain number of minor cycles containing less tokens than N_M that maintain the liveness bound of the other transitions. ■



a) Original net with t_2 and t_3 concurrent.

b) Transformed net: t_2 and t_3 sequentialized and p_2 made implicit.

c) Elimination of implicit places. Main loop: p_1, p_3, p_6, p_4. Minor cycle: p_1, p_3, p_5.

Figure 3.2: Example of structural sequentialization.

An example of application of the lemma follows, in order to clarify the procedure. Consider the net depicted in figure 3.2.a. This net contains only two cycles, namely t_1, t_2, t_4, and t_1, t_3, t_4; we can then add either the cycle t_1, t_2, t_3, t_4 or t_1, t_3, t_2, t_4; figure 3.2.b depicts the resulting net in case we choose to add the second cycle. In this case only place p_6 (from t_3 to t_2) needs to be added to obtain the longer cycle, and it should be marked with one token, so that the new cycle comprising places p_1, p_3, p_6, p_4 contains two tokens, as the original cycle p_1, p_2, p_4 (while the other original cycle p_1, p_3, p_5 contained only one). In our example, we need not to iterate the procedure since we already have obtained a cycle containing all transitions of the marked graph. At this point we can identify and eliminate the implicit places that have been created during the cycles interleaving procedure. In the present example, we can easily see that place p_2 becomes implicit in figure 3.2.b, so that it can be removed, finally leading ourselves to the marked graph depicted in figure 3.2.c.

It should be evident that the marked graph transformed by applying the above lemma has a mean cycle time which is greater than or equal to the mean cycle time of the original one, since some additional constraints have been added to the enabling of transitions: hence the mean cycle time of the transformed marked graph is a lower bound for the performance of the original one. Now if $N_M = \max_{t \in T} L(t) = 1$ in the above lemma, we re-find the lower bound of theorem 3.2.1. In the case of $N_M > 1$ we can show that the mean cycle time of the transformed net cannot exceed Γ^{max} of equation 3.21 as follows.

Theorem 3.2.2 *For any live and bounded marked graph with a specification of the average service time s_j for each transition t_j it is not possible to assign probability distribution functions to the transition service times such that the mean cycle time is greater than*

$$\Gamma^{max} = \sum_{j=1}^{m} \frac{s_j}{L(t_j)} \tag{3.22}$$

independently of the topology of the net (and thus independently of the potential maximum degree of parallelism intrinsic in the marked graph).

Proof. Without loss of generality, assume that transitions in the net resulting from the application of lemma 3.2.1 are partitioned in two classes $\mathcal{C}_2$ and $\mathcal{C}_1$, with liveness bounds $K_2 = N_M > 1$ and $K_1 < N_M$, respectively (the proof is easily extended to the case of more than two classes). Construct a new model containing only K_1 tokens in the main cycle; at this point all transitions behave as K_1–servers, so that the mean cycle time is given by the sum of the firing times of all transitions, divided by the total number of customers in the main loop K_1; moreover the delay time for the transitions belonging to class $\mathcal{C}_1$ is simply given by $S_1 = \sum_{t_j \in \mathcal{C}_1} s_j$. Now if we increase the number of tokens in the main loop from K_1 to K_2, the delay time of $\mathcal{C}_1$ cannot increase, so that the contribution of $\mathcal{C}_1$ to the mean cycle time cannot exceed S_1 for each of the first K_1 tokens. Under the hypothesis that the throughput of the system is given by the inverse of Γ^{max} (i.e., assuming $X = 1/\Gamma^{max}$), the average number of tokens of the main loop computed using Little's formula cannot exceed $N_1 = XS_1$, therefore the average number of tokens available to fire transitions in $\mathcal{C}_2$ cannot be lower than

$$N_2 = K_2 - N_1 = K_2 \frac{\frac{K_2 - K_1}{K_1} \sum_{t_j \in \mathcal{C}_1} s_j + \sum_{t_j \in \mathcal{C}_2} s_j}{\sum_{t_j \in \mathcal{C}_2} s_j + \frac{K_2}{K_1} \sum_{t_j \in \mathcal{C}_1} s_j} \tag{3.23}$$

On the other hand, we need only

$$N_2 = XS_2 = K_2 \frac{S_2}{\sum_{t_j \in \mathcal{C}_2} s_j + \frac{K_2}{K_1} \sum_{t_j \in \mathcal{C}_1} s_j} \tag{3.24}$$

tokens to sustain throughput X in subnet $\mathcal{C}_2$, so that we are assuming a delay in $\mathcal{C}_2$

$$S_2 \leq \frac{K_2 - K_1}{K_1} \sum_{t_j \in \mathcal{C}_1} s_j + \sum_{t_j \in \mathcal{C}_2} s_j \tag{3.25}$$

Now we claim that this is the actual maximum delay because the first K_1 tokens can proceed at the maximum speed in the whole net, thus experiencing only delay $\sum_{t_j \in \mathcal{C}_2} s_j$ in subnet $\mathcal{C}_2$, while the remaining $K_2 - K_1$ tokens can also queue up for travelling through $\mathcal{C}_1$, thus experiencing an additional delay of $\frac{1}{K_1} \sum_{t_j \in \mathcal{C}_1} s_j$ each. ■

Now, taking into account that the liveness bound of a transition of a net $\mathcal{N}$ does not change in the reverse net $\mathcal{N}^{-1}$, an analogous result to property 3.1.3 for the lower bound on throughput can be derived.

Property 3.2.1 *Let $\mathcal{N}$ be a strongly connected marked graph and $\mathcal{N}^{-1}$ its reverse net. Then, the lower bounds on throughput obtained for both nets as in theorem 3.2.2 are the same.*

3.2.3 Reachability of the lower bound

The lower bound in performance given by the computation of $1/\Gamma^{max}$ as defined in theorem 3.2.2 can be shown to be reachable for any marked graph topology and for some assignement of probability distribution functions to the service time of transitions, exploiting the reachability of the trivial bound shown in theorem 3.2.1 for 1–live marked graphs.

Theorem 3.2.3 *For any strongly connected marked graph with a specification of the average service time s_j for each transition t_j, and for all $0 < \epsilon \leq 1$, it is possible to assign probability distribution functions to the transition service times such that the mean cycle time is:*

$$\Gamma^{max} = \sum_{j=1}^{m} \frac{s_j}{L(t_j)} - O(\epsilon) \tag{3.26}$$

independently of the topology of the net (and thus independently of the potential maximum degree of parallelism intrinsic in the marked graph).

Proof. By construction, in a very similar way than in the case of theorem 3.2.1. The only technical difference is that now, without any loss of generality, we assume first of all to enumerate transitions in non-increasing order of liveness bound, i.e., rename the transitions in such a way that $\forall t_i, t_j \in T$, $i > j \implies L(t_i) \leq L(t_j)$. Then, as in the case of theorem 3.2.1, we can show that the association of the family of random variables $x_{s_j}^{j-1}(\epsilon)$ with each transition $t_j \in T$ yields exactly the mean cycle time Γ^{max} claimed by the theorem. To give the proof we consider a sequence of models ordered by the index of transitions, in which the q^{th} model of the sequence has transitions $t_1, t_2, \ldots, t_q$ timed with the random variables $x_{s_j}^{j-1}(\epsilon)$, and all other transitions immediate (firing in zero time); the $|T|^{th}$ model in the sequence represents the resulting model that is expected to provide the example of reachability of the lower bound. By induction we prove that the q^{th} model in the sequence has a mean cycle time

$$\Gamma_q = \sum_{j=1}^{q} \frac{s_j}{L(t_j)} - O(\epsilon) \tag{3.27}$$

Base: $q = 1$: Trivial since the repetitive cycle that constitute the steady-state behaviour of the marked graph contains only one ($L(t_1)$–server) deterministic transition with average firing time $\Gamma_1 = s_1/L(t_1)$.

Induction step: $q > 1$: Taking the limit $\epsilon \rightarrow 0$, each server of the newly timed transition t_q will fire most of the times with time zero, thus normally not contributing to the computation of the mean cycle time, that will be just

$$\Gamma_{q-1} = \sum_{j=1}^{q-1} \frac{s_j}{L(t_j)} - O(\epsilon) \tag{3.28}$$

(as in the case of model $q-1$) with probability $1 - \epsilon^{q-1}$. On the other hand, each of the servers of the newly timed transition has a (very small) probability ϵ^{q-1} of delaying its firing of a time s_q/ϵ^{q-1}, which is at least order of $1/\epsilon$ bigger than any other firing time in the cycle. Now if $L(t_q) = 1$, then the proof is completed, since also $\forall j > q$,

$L(t_j) = 1$ by hypothesis, and we reduce to the induction step of the proof of theorem 3.2.1. Instead if $L(t_q) > 1$ then we can consider $L(t_q)$ consecutive firings of t_q, and compute the average firing time as the total time to fire $L(t_q)$ times the transition, divided by $L(t_q)$. Now if we consider m consecutive firings of instances of transition t_q we obtain an average delay:

$$\sum_{j=0}^{m-1} (1 - \epsilon^{q-1})^j \epsilon^{(q-1)(m-j)} \frac{(m-j)s_q}{\epsilon^{(q-1)}} = s_q(1 + O(\epsilon)) \tag{3.29}$$

Therefore the mean cycle time of the q^{th} model will be

$$\Gamma_q = (1 - O(\epsilon^{q-1}))\Gamma_{q-1} + \frac{s_q}{L(t_q)}(1 + O(\epsilon)) = \sum_{j=1}^{q} \frac{s_j}{L(t_j)} - O(\epsilon). \ \blacksquare \tag{3.30}$$

3.2.4 A polynomial algorithm to compute the lower bound

First of all we recall (cfr. property 2.1.14) that in the case of live marked graphs the liveness bound equals the enabling and the structural enabling bounds for each transition; thus we present a characterization of the problem of the determination of the structural enabling bound in terms of a linear programming problem, which is known to be solvable in polynomial time.

For any transition $t \in T$, the computation of the structural enabling bound $SE(t)$ is formulated in definition 1.2.3, in terms of problem (LPP1). In that problem we can observe that the vector M is redundant in the system of linear inequalities, so that we can remove it, obtaining:

$$\begin{array}{rll} SE(t) = & \text{maximize} & k \\ & \text{subject to} & M_0 + C \cdot \vec{\sigma} \geq kPRE[t] \\ & & M_0 + C \cdot \vec{\sigma} \geq 0, \ \vec{\sigma} \geq 0 \end{array} \tag{LPP9}$$

Alternatively, we can switch to the dual linear programming problem:

$$SE(t) = \begin{array}{ll} \text{minimize} & Y^T \cdot M_0 \\ \text{subject to} & Y^T \cdot C \leq 0 \\ & Y^T \cdot PRE[t] = 1 \\ & Y \geq 0 \end{array} \tag{LPP10}$$

Marked graphs are consistent nets with a single minimal T-semiflow which is the vector $\mathbb{1}$ (property 2.1.12), so that the constraint $\vec{\sigma} \geq 0$ can be relaxed in the primal problem. The effect on the dual problem of this relaxation is the transformation of the first constraint into $Y^T \cdot C = 0$. In other words, the dual problem for the computation of $SE(t)$ can be rewritten as follows:

$$SE(t) = \begin{array}{ll} \text{minimize} & Y^T \cdot M_0 \\ \text{subject to} & Y^T \cdot C = 0 \\ & Y^T \cdot PRE[t] = 1 \\ & Y \geq 0 \end{array} \tag{LPP11}$$

This linear programming problem is less complex to solve with the simplex algorithm than the original dual problem because it involves the introduction of fewer slack variables.

For all strongly connected marked graph there exists an elementary P-semiflow for which the optimum of the objective function is achieved, as shown in theorem 3.1.2. In case of marked graphs, these elementary P-semiflows can only be elementary cycles, so that we can give the following interpretation of the linear programming problem (LPP11) in net terms: the liveness bound for a transition t of a strongly connected marked graph is given by the minimum number of tokens contained in any cycle of places containing transition t. In a non-strongly connected marked graph there can be no such cycle, so that this number can be infinite.

As final remarks we can state the following:

Property 3.2.2 *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked graph.*

1. *Liveness for $\langle \mathcal{N}, M_0 \rangle$ can be a byproduct of a more general (polynomial complexity) computation: $\langle \mathcal{N}, M_0 \rangle$ is a live marked graph if and only if for all transition t, $SE(t) > 0$.*

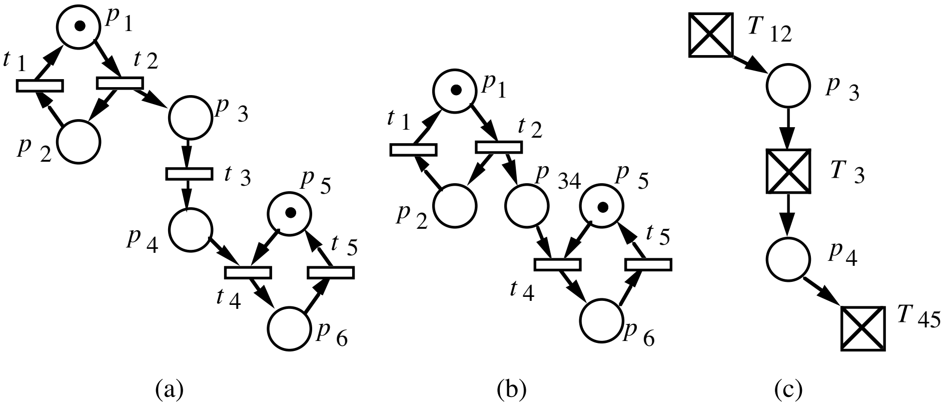


Figure 3.3: Non-strongly connected marked graphs.

2. If $\langle\mathcal{N}, M_0\rangle$ is live and $\exists t \in T$ such that $SE(t) = 1$, then $\forall t' \in T$ belonging to the same cycle denoted by Y in (LPP11), $SE(t') = 1$.

Note that the application of property 3.2.2.2 reduces the computational complexity of the structural enabling bound of all transitions.

3.3 Extending results to unbounded marked graphs

In the literature on deterministically timed marked graph models the case of non-strongly connected nets is usually considered a trivial extension to be left to the imagination of the reader [RH80,Mag84]. In this section we argue that the question is less trivial than one can perceive at first glance, and in fact we shall derive some examples that show that "direct" extensions of the results obtained in the case of strongly connected marked graphs, in general, make no sense. In fact, for the upper bound on throughput, we obtain a result similar to that proposed by F. Baccelli et al. [BBW89], even though their work is situated in a quite different framework.

Example 1. Let us first consider as an example the non-strongly connected marked graph in figure 3.3.a. First of all we can see that transition t_3 has an infinite liveness bound, so that in steady-state it *should not contribute* to the computation of the mean cycle time. Indeed, suppose that t_3 has a deterministic service time of 1000 time units, while transitions t_1 and t_2 have a deterministic service time of 1 time unit; thus the cycle t_1–t_2 starts generating tokens at a rate of one token every 2 time units, so that initially tokens accumulate in place p_3. At time 1001 eventually the first instance of transition t_3 fires, and at that point we reach the steady-state condition in which 499 instances of firing of t_3 are concurrently enabled, with a remaining enabling time shifted of two time units between each pair of subsequent firing instances. As we can see, the actual firing rate in steady-state for transition t_3 is 1/2 firings per second, i.e., it is determined by the mean cycle time of transitions t_1–t_2 completely independent of the service time of t_3 itself. Therefore, from the steady-state performance point of view, transition t_3 behaves as if it were an immediate transition, and it can be reduced by fusing places p_3 and p_4 into a single place p_{34}, as shown in figure 3.3.b.

Now let us consider the behaviour of the other two transitions t_4 and t_5. Their actual firing rate is determined both by their own service times and the rate with which the cycle t_1–t_2 is able to produce the tokens that are consumed by t_4 from place p_{34}. Thus the mean cycle time in steady-state condition for transitions t_4–t_5 is given by the maximum between the mean cycle time of t_1–t_2 and the sum of the service times of t_4 and t_5 (this sum would be the mean cycle time of the subnet generated by t_4 and t_5 if it were considered in isolation, i.e., the *potential mean cycle time* of t_4–t_5). In the case in which the mean cycle time of t_1–t_2 were greater than the one of t_4–t_5, the number of tokens at place p_{34} would remain bounded and the firing rate of t_4–t_5 would be the inverse of the mean cycle time of t_1–t_2. On the other hand, in the case in which the mean cycle time of t_1–t_2 were less than the one of t_4–t_5, place p_{34} would accumulate tokens and marking process of this place would not be (even weakly) ergodic. However, firing rate of transitions t_4–t_5 would be, in that case, equal to the inverse of their potential mean cycle time. In the case of equality between mean cycle time of t_1–t_2 and t_4–t_5, marking ergodicity at place p_{34} depends on the probability distribution of service time of transitions. In the particular case of deterministic timing, the marking process is weakly ergodic, while in the case of exponentially

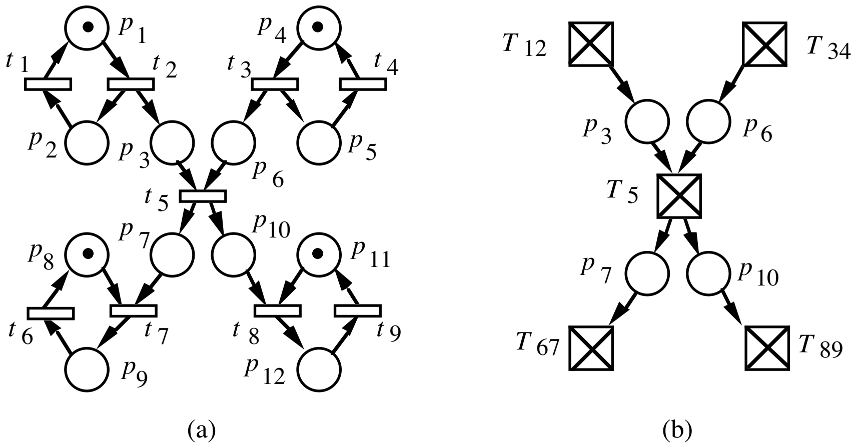


Figure 3.4: A more general non-strongly connected marked graph.

distributed service times the marking process is non-ergodic (because the embedded Markov process is *null-recurrent*).

Example 2. Let us consider the more general example shown in figure 3.4.a. Also in this case it is easy to understand that transition t_5 gives no contribution to the steady-state cycle time because it has an infinite liveness bound (it behaves as an immediate transition). However in this case we cannot just delete it because of the synchronization constraint that is due to its multiple input places (p_3 and p_6). On the other hand, it is clear that the two subnets composed of t_1–t_2 and t_3–t_4 behave completely independently of each other and of the rest of the net. If the mean cycle times of these two subnets are not exactly equal (let us assume without loss of generality that the mean cycle time of t_1–t_2 is greater than that of t_3–t_4), then one of the input places of t_5 (p_6 with our assumption) accumulates an infinite number of tokens in steady-state (in other words, the marking process at this place is not ergodic); thus it becomes redundant (in steady-state) since it cannot constraint the enabling condition of t_5, and it can be deleted without altering the behaviour of the net. In the case of exactly equal mean cycle times of the two subnets (t_1–t_2 and t_3–t_4), marking ergodicity depends on the distribution functions associated with transitions. For instance, for deterministic timing the marking process at p_3 and

p_6 remains bounded (i.e., it is weakly ergodic). On the other hand, for exponential timing, the marking of both places is a null-recurrent Markov process, thus non-ergodic. Deleting all the places that become unbounded in steady-state due to the average transition firing times, we obtain that the net is partitioned in disconnected subnets that can be studied independently of one another. Of course, not only the input but also the output places of t_5 (p_7 and/or p_{10}) may accumulate an infinite number of tokens in steady-state, provided that the potential mean cycle times of their output transitions (respectively, t_7 and t_8) are greater than the actual firing time of t_5. In this case, also the output places become redundant and can be deleted, and we may study the steady-state behaviours of the four disconnected subnets in isolation.

From the analysis of the above examples we can draw two considerations.

First: Marking ergodicity is not assured in the case of non-strongly connected marked graphs. Places having non-ergodic marking process can be found among structurally unbounded places (places do not belonging to any strongly connected component) in two cases: (1) after the comparison between the actual input firing rate and the potential firing rate of the output strongly connected component (example 1), or (2) after the comparison among the actual firing rate of all strongly connected components being synchronized by a given transition (example 2). In other words, strongly connected components of the marked graph can be seen as *producers* of *parts* (or *data*) for other components and *consumers* of parts that are produced by other components. Connections among these producers/consumers are modelled by means of places (or *buffers*). A place is marking ergodic if the throughput of the corresponding producer is less than the service rate of the consumer.

Second: There exists a *partial order relation* "$\succ$" among subsets of transitions defined as $T_i \succ T_j$ iff the firing delay of transitions in T_i can affect the actual firing rate of transitions in T_j but not vice versa. This partial order relation can be computed by applying a standard algorithm for the derivation of a *condensation* of the original net, as we explain below.

The previous considerations suggest that the first step that must be taken in order to check marking ergodicity and to compute actual throughput of transitions is the construction of the *condensation* of the net. The condensation of a given directed graph [Deo74] represents the interconnections among the strongly connected components of the original graph. Therefore, the vertices v_i of the condensation correspond with the strongly connected components C_i of the original one. There is an arc from one vertex v_1 to a different vertex v_2 in the condensation iff there is an arc in the original graph from some vertex in the component C_1 to some vertex in the component C_2.

Definition 3.3.1 (Condensation of a marked graph) [Deo74] *Let $\mathcal{N}$ be a marked graph. The marked graph resulting from $\mathcal{N}$ after the substitution of each strongly connected component by a single transition is called condensation of $\mathcal{N}$, and denoted $\mathcal{N}_c$. There is a place p_{ij} connecting two transitions T_i, T_j in the condensation of a marked graph ($p_{ij} \in T_i^\bullet \cap {}^\bullet T_j$) iff p_{ij} connects, in the original net, at least one transition of the strongly connected component associated with T_i with another one of the component associated with T_j ($p_{ij} \in t_{i_1}^\bullet \cap {}^\bullet t_{j_1}$, with $t_{i_1} \in T_i$ and $t_{j_1} \in T_j$).*

The condensation of a directed graph is always a *directed acyclic graph*, because if there were a cycle in it, then all the components in the cycle would really correspond to one strongly connected component in the original graph. An efficient algorithm for the computation of strongly connected components and the condensation of a directed graph can be found, for instance, in [MB86].

Now, let us remark that two kinds of transitions can be found in the condensation of a given non-strongly connected marked graph: those with infinite liveness bound (corresponding with trivial strongly connected components having only one transition) and those with finite liveness bound (obtained from the substitution of a non-trivial strongly connected component, i.e., having more than one transition). The first ones have null potential mean cycle time (i.e., infinite throughput if they are considered in isolation), while the potential mean cycle times of the second are always finite.

Figure 3.3.c represents the condensation of the marked graph depicted in figure 3.3.a. Its transitions can be considered as *complex* *servers* in a producers/consumers system, from a queueing theory point of view. Transitions T_{12} and T_{45} have finite liveness bound while transition T_3 has infinite liveness bound. Considering the net of figure 3.4.a, its condensation is depicted in figure 3.4.b, where transitions T_{12}, T_{34}, T_{67}, and T_{89} have finite liveness bound, while the one of T_5 is infinite.

The condensation of a given marked graph defines a *partial order relation* on the set of its strongly connected components:

Definition 3.3.2 (Partial order relation) *Let $\mathcal{N}$ be a marked graph and $\mathcal{N}_c$ its condensation. We denote "$\succ$" the binary relation among transitions of $\mathcal{N}_c$ defined as follows: $T_i \succ T_j$ iff there is a directed path of length one or more from T_i to T_j in $\mathcal{N}_c$.*

From previous definition and from the fact that a condensation of a directed graph is always a directed acyclic graph, the next property follows:

Property 3.3.1 *Relation "$\succ$" is a partial order on the set of transitions of the condensation $\mathcal{N}_c$ of a marked graph, because it is irreflexive and transitive.*

The method for the computation of steady-state throughput of transitions of a non-strongly connected marked graph that we present now is based on the previous considerations, using the above defined partial order relation, and considers the liveness bounds of transitions and their potential mean cycle time (i.e., their mean cycle time if they were in isolation). Before the presentation of the computation method we recall the concept of *maximal* element for a partial order relation:

Definition 3.3.3 (Maximal element) *Let $\mathcal{C}$ be a set and "$\succ$" be a partial order relation defined on $\mathcal{C}$. Then, $c \in \mathcal{C}$ is a maximal element of $\mathcal{C}$ for the relation "$\succ$" iff $\nexists c' \in \mathcal{C}$ such that $c' \succ c$.*

For the previously introduced partial order on the set of strongly connected components of a marked graph, maximal elements are the *source* transitions of the condensation of the graph.

Theorem 3.3.1 *Let $\langle \mathcal{N}, M_0 \rangle$ be a non-strongly connected marked graph with some given average service times associated with transitions. Let $\mathcal{N}_c$ be the condensation of $\mathcal{N}$. Let T_i, $i = 1, \ldots, K$, be a transition of $\mathcal{N}_c$ and $\Gamma_{(i)}^{pot}$ its potential mean cyle time (mean cycle time of the strongly connected component associated with T_i, considered in isolation). The actual mean cycle time $\Gamma_{(i)}$ of T_i, is*

i) If T_i is a maximal element for "$\succ$" then $\Gamma_{(i)} = \Gamma_{(i)}^{pot}$.

ii) If T_i is not a maximal element for "$\succ$", let $\gamma_i = \max\{\Gamma_{(i_1)}^{pot}, \ldots, \Gamma_{(i_r)}^{pot}\}$ where $T_{i_j}, j = 1, \ldots, r$ are such that $T_{i_j} \succ T_i, j = 1, \ldots, r$ and $T_{i_j}^{\bullet} \subseteq {}^{\bullet}T_i, j = 1, \ldots, r$ (i.e., there is a path of length one from T_{i_j} to T_i, $j = 1, \ldots, r$). Then $\Gamma_{(i)} = \max\{\Gamma_{(i)}^{pot}, \gamma_i\}$.

We remark that transitions T_i with infinite liveness bound have null potential mean cycle time ($\Gamma_{(i)}^{pot} = 0$). The exact mean cyle time of transitions can be computed according to the above theorem, starting from the maximal strongly connected components, which are independent of the others, and then iteratively using the results to solve the subsequent components.

Note that, in practice, the potential mean cycle time of strongly connected components ($\Gamma_{(i)}^{pot}, i = 1, \ldots, K$) are not known. Moreover, their computation is not possible, so far, in polynomial time on the net size from the transition service times. However, the bounds for the mean cycle time of strongly connected marked graphs derived in previous sections could be applied for deriving upper and lower bounds for the mean cycle time of transitions in the whole net, *substituting in theorem 3.3.1 the exact values $\Gamma_{(i)}^{pot}$ of the mean cycle time of isolated components by their upper and lower bounds, respectively.*

Finally, we remark that, as a by-product of theorem 3.3.1, *necessary and sufficient conditions for the marking ergodicity at places can be deduced.* If we define the input flow at a given place in the condensation of a marked graph as the actual throughput of its input transition, in the case of a transition with several input places, only places with minimum input flow has ergodic marking process (the rest of places accumulate infinite tokens in the limit). In the case of a transition with infinite liveness bound and only one input place, this place has always ergodic marking process. On the other hand, if the transition has finite liveness bound and only one input place, two cases arise: if the input flow to the place is less than the potential service rate of the transition (i.e., the inverse of its potential mean cyle time), the marking of the place is ergodic; but if the input flow is greater than the potential service rate of the transition, the marking of the place is unbounded, thus non-ergodic. The case of equality between the input flow and the service rate is not very well known, so far. It depends on the probability distribution functions associated with service time of transitions wether the marking process is ergodic or not. For instance, in the case of deterministic timing, equality between input and service rates assures weak ergodicity, while in the case of exponential distributions, such equality assures non-ergodicity (null-recurrent embedded Markov process).

3.4 Conclusions

The computation of the throughput of strongly connected marked graphs has been considered by many authors in the case of deterministically timed transitions [Ram74,Sif78,RH80,Mag84,Mur85]. In this chapter, we have shown that deterministic case represents an upper bound in performance independently of the probability distribution also in the framework of stochastic Petri nets. The computation of this bound has been reformulated in terms of a linear programming problem. Moreover, we have shown how the upper bound is reached not only in the deterministic case but also by stochastic models, with arbitrary values of coefficients of variation.

Concerning the trivial lower bound in performance consisting of the inverse of the sum of the average service times of all transitions, it has been shown to be reachable in the case of 1–bounded marked graphs (in fact, for marked graphs with 1–live transitions). The improvement for the case of bounded marked graphs, obtained dividing by the liveness bounds of transitions is new, and has been shown to be reachable for some service probability distributions when the coefficient of variation increases.

The extension to the case of non-strongly connected marked graphs, which has been considered in the literature of deterministically timed nets as straightforward, is less trivial than one can perceive at first glance. We have derived an algorithm for the computation of exact measures for the performance of non-strongly connected marked graphs, from the knowledge of the throughput of their isolated strongly connected components.

All the algorithms that we present have polynomial complexity on the net size, since they are mainly based on the solution of linear programming problems, which are known to be solvable in polynomial time [Kar84].

Some interesting behavioural properties have been derived from the performance/quantitative approach. We remark the monotonicity of performance bounds with the increasing of initial resources or with their faster processing. A reversibility property, concerning the equality between bounds for a given strongly connected marked graph and its reverse net, has been also obtained, analogous to that presented in [DLT90]. Finally, a polynomial complexity liveness characterization of strongly connected marked graphs in terms of the finiteness of the mean cycle time (which can be applied also to non-strongly connected marked graphs, looking for the liveness of isolated strongly connected components) is a good example of the interest of interleaving the qualitative and quantitative theories.

Chapter 4

Bounds for live and bounded free choice nets

The results presented in this chapter (that include part of those in [CCS90a] and [CCS90b]) are an extension to live and bounded free choice nets (see definition 2.1.11) of the performance bounds for strongly connected marked graphs developed in the previous chapter. The idea is that several consistent firing count vectors can be reproduced in steady-state, but decisions, freely done at certain places, are completely governed by the stochastic interpretation (in particular, by the routing rates) of the net, and the vector of visit ratios for transitions can be defined independently of the marking and the service times (see section 2.1).

In section 4.1 we focus our attention on throughput upper bounds for live and bounded free choice nets. Using Little's law like in previous chapter and structural linear marking relations, linear programming problems are derived whose optimum solutions are lower bounds for the mean cycle time of transitions. These problems include structural information of the net by means of the (pre-, post-) incidence matrices. All parameters defining stochastic interpretation are summarized in the vector of average service demands for transitions (products of visit ratios by average service times), which can be efficiently computed for live and bounded free choice nets (see section 2.1).

Lower bounds for the steady-state throughput are considered in section 4.2. These bounds are computed from the liveness bounds of transitions (obtained from linear programming problems in the case of live and bounded free choice nets) and from the average service demands for transitions.

The throughput upper bound is shown to be reachable for 1–bounded nets for some distribution functions of service times with arbitrary mean values and for some conflict resolution policy, with arbitrary long run rates. The lower bound on throughput is reachable for 1–bounded nets.

4.1 Upper bounds for the steady-state throughput

The computation of upper bounds for the throughput of transitions, defined as the average number of firings per time unit, is consider in this section, for live and bounded free choice nets.

In section 4.1.1, Little's law and structural linear marking relations are applied for the derivation of linear programming problems, analogous to that presented in section 3.1. The bounds obtained using P-semiflows (section 4.1.1.2) can be improved taking into account other marking invariants derived from the concept of *trap* (section 4.1.1.3), or after the addition of some *implicit places* to the net (section 4.1.2).

The bounds derived in sections 4.1.1 and 4.1.2 are non-reachable, in general. A reachable throughput upper bound for the case of 1–bounded nets is obtained in section 4.1.3. The idea is the following: a reachable bound for strongly connected marked graphs was computed in previous chapter using the circuits of the net. Such circuits can be interpreted in algebraic terms for marked graphs as elementary P-semiflows (see theorem 2.1.6.1). This is the reason why we try to derive bounds for free choice nets from P-semiflows in section 4.1.1.2. Other natural extension of circuits of marked graphs for the case of free choice nets can be found in the framework of graph theory: *multisets of circuits*. From this approach, a reachable throughput upper bound can be derived for 1–bounded nets.

4.1.1 Little's law and linear marking relations

Let us recall the system of inequalities (3.6) presented in chapter 3:

$$\Gamma_{(j)}\overline{M} \geq PRE \cdot \vec{D}^{(j)} \tag{4.1}$$

where $\Gamma_{(j)}$ is the mean cycle time of transition t_j (i.e., the inverse of its throughput), $\overline{M}$ is the vector of limit average markings, PRE is the pre-incidence matrix of the net, and $\vec{D}^{(j)}$ is the vector of average service demands for transitions, with components $\vec{D}_i^{(j)} = v_i^{(j)} s_i$, $i = 1, \ldots, m$.

We remark that vector $\vec{D}^{(j)}$ can be efficiently computed for live and bounded free choice nets, if average service times s_i are given, because the vector of visit ratios $\vec{v}^{(j)}$ can be derived for such nets by solving a linear system of equations (free choice nets are FRT-nets; therefore, theorem 2.1.2 can be used).

A goal of this section is the computation of lower bounds for the mean cycle time of transitions, based on the inequality (4.1). Since the limit average marking $\overline{M}$ is unknown, linear marking relations derived from the underlying net will be considered to achieve this goal:

$$Z^T \cdot M \leq k, \quad \forall M \in R(\mathcal{N}, M_0), \text{ with } Z \gneq 0 \tag{4.2}$$

Linearity is required in the above relation because, taking into account the definition of the limit average marking, a similar inequality can be derived for $\overline{M}$:

$$Z^T \cdot \overline{M} = \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau Z^T \cdot M_u \, du \leq \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau k \, du = k \tag{4.3}$$

In this case, the (unknown) vector $\overline{M}$ can be substituted in (4.1), premultiplied by Z, obtaining:

$$\Gamma_{(j)} \geq \frac{Z^T \cdot PRE \cdot \vec{D}^{(j)}}{k} \tag{4.4}$$

and so a lower bound for the mean cycle time of t_j.

An additional advantage can be taken of the use of linear relations, since this linearity will lead, in most cases, to *polynomial complexity* calculations, based on *linear algebra* and *linear programming* techniques.

4.1.1.1 Structural linear marking relations

Since the limit average marking $\overline{M}$ is unknown, we can use the approximation given by linear relations verified by all reachable markings. A first family of linear marking relations is obtained considering those being *structurally characterized*. These are stronger conditions than those expressed by inequality (4.2) (behaviourally defined), but they provide easier and more efficient techniques for their manipulation. Structural linear marking relations can be expressed using the incidence matrix C of the net:

$$Y^T \cdot C = 0, Y \geq 0 \Longrightarrow Y^T \cdot M = Y^T \cdot M_0, \forall M \in R(\mathcal{N}, M_0), \forall M_0 \tag{4.5}$$

$$Y^T \cdot C \lneq 0, Y \geq 0 \Longrightarrow Y^T \cdot M \leq Y^T \cdot M_0, \forall M \in R(\mathcal{N}, M_0), \forall M_0 \tag{4.6}$$

$$Y^T \cdot C \gneq 0, Y \geq 0 \Longrightarrow Y^T \cdot M \geq Y^T \cdot M_0, \forall M \in R(\mathcal{N}, M_0), \forall M_0 \tag{4.7}$$

Let us consider firstly the case of equality relation given by equation (4.5). Vectors $Y \geq 0$ verifying this equation are often called conservative components or *P-semiflows* (see section 1.2.2), and they have been used for the computation of throughput upper bounds for strongly connected marked graphs, in section 3.1.1, by premultiplying the inequality (4.1). The obtained results using P-semiflows as well as their limitations for the computation of reachable (i.e., tight) bounds for live and bounded free choice nets are summarized in the next section.

Regarding structural linear inequality relations for the reachable markings of a marked Petri net, vectors $Y \geq 0$ verifying (4.6) could be considered. Premultiplying the linear state equation of the net by such vectors, the following sequence of inequalities is obtained for each sequence of successor markings, and for all initial marking M_0:

$$Y^T \cdot M_0 \geq \cdots \geq Y^T \cdot M_{i-1} \geq Y^T \cdot M_i \geq Y^T \cdot M_{i+1} \geq \cdots \tag{4.8}$$

Moreover, $Y^T \cdot C \neq 0$ implies that there exists (at least) a transition t_j such that $Y^T \cdot C[t_j] < 0$, and if $M_i[t_j\rangle M_{i+1}$ then $Y^T \cdot M_i > Y^T \cdot M_{i+1}$ in the sequence of inequalities (4.8) (i.e., strict inequality). But in this case the net cannot be live (because if it was live then transition t_j could be fired an infinite number of times, an infinite number of strict inequalities would appear in (4.8), and this is impossible if the initial marking is finite). Thus, linear inequalities of the form $Y^T \cdot C \lneq 0$ are not usefull for us.

Non-negative vectors satisfying the inequality (4.7): $Z^T \cdot C \gneq 0$ cannot be used directly for the substitution of $\overline{M}$ in (4.1) (because they give inequalities in the opposite direction). P-semiflows Y could be considered such that $Y - Z \geq 0$, thus:

$$(Y - Z)^T \cdot C = \underbrace{Y^T \cdot C}_{0} - Z^T \cdot C \lneq 0 \tag{4.9}$$

But the existence of such vectors $Y - Z \geq 0$, $(Y - Z)^T \cdot C \lneq 0$, is not possible for conservative nets (and structurally live structurally bounded nets are conservative; see, e.g., [Sil85]).

Alternatively, other linear marking inequalities of the form $Y_\Theta^T \cdot M \geq 1$, for all (non-transient) marking M can be derived considering vectors $Y_\Theta \geq 0$ having a *trap* Θ as support. Traps are sets of places which remain marked once they have gained at least one token. This structural concept can be used to improve the throughput upper bound computed by means of Little's law and P-semiflows, and will be explained later.

4.1.1.2 Little's law and P-semiflows

P-semiflows Y are non-negative left annullers of the incidence matrix C (i.e., $Y^T \cdot C = 0$, thus $Y^T \cdot M = Y^T \cdot M_0$ for all reachable marking M). Now, using relation (4.4), the following lower bound for the mean cycle time of a given transition t_j can be derived:

$$\Gamma_{(j)} \geq \max_{Y \in \{P-semiflow\}} \frac{Y^T \cdot PRE \cdot \vec{D}^{(j)}}{Y^T \cdot M_0} \tag{4.10}$$

The previous lower bound can be formulated in terms of a fractional programming problem and later, after some considerations (see section 3.1.1), transformed into a linear programming problem:

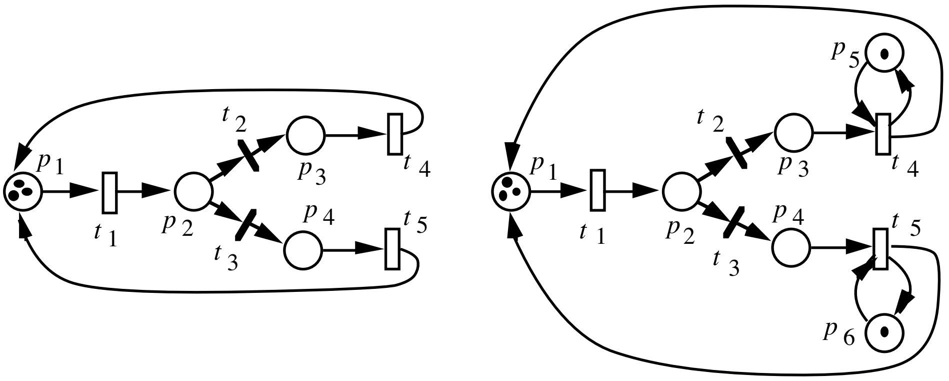
Theorem 4.1.1 *For any net, a lower bound for the mean cycle time of transition t_j can be computed by the following linear programming* *problem:*

$$\Gamma_{(j)} \geq \Gamma^{PS}_{(j)} = \begin{array}{rl} \textit{maximize} & Y^T \cdot PRE \cdot \vec{D}^{(j)} \\ \textit{subject to} & Y^T \cdot C = 0 \\ & Y^T \cdot M_0 = 1 \\ & Y \geq 0 \end{array} \qquad \text{(LPP12)}$$

If the solution of the problem (LPP12) is unbounded, since it is a lower bound for the mean cycle time of transition t_j, the non-liveness can be assured (infinite cycle time). If the visit ratios for all transitions are non-null, then $\vec{D}^{(j)} > 0$, and the unboundedness of the above problem implies that a total deadlock is reached by the net. This result has the following interpretation: if the problem (LPP12) is unbounded then there exists an unmarked P-semiflow, and the net is non-live (recall corollary 2.1.11).

Corollary 4.1.1 *The problem (LPP12) has unbounded solution iff $\exists Y \gneq 0$ such that $Y^T \cdot M_0 = 0$ and $Y^T \cdot C = 0$. Moreover, if this occurs, the net is non-live.*

In order to interpret the result presented in theorem 4.1.1, let us consider the particular case of the state machine (see definition 2.1.12) depicted in figure 4.1.a. Assume that $s_1 = 1$, $s_2 = s_3 = 0$, $s_4 = 1$, and $s_5 = 2$ are the average service times of t_1, t_2, t_3, t_4, and t_5, respectively, and that routing rates solving the conflict at place p_2 are $r_2 = r_3 = 1/2$ for the firing of transitions t_2 and t_3. In this case, the vector of visit ratios for transitions is $\vec{v}^{(1)} = (1, 1/2, 1/2, 1/2, 1/2)^T$, thus the vector of average service demands is $\vec{D}^{(1)} = (1, 0, 0, 1/2, 1)^T$. The unique elementary P-semiflow is $Y_1 = (1, 1, 1, 1)^T$, and it is such that $Y_1^T \cdot M_0 = 3$. Therefore, the application of theorem 4.1.1 gives the value $\Gamma^{PS}_{(1)} = (1 + 1/2 + 1)/3 = 0.8333$. In fact, in this case the obtained value is the exact mean cycle time of transition t_1, independently of the probability distribution of service times. As we remarked in chapter 1, state machines are the Petri net counterpart of classical queueing networks. Since we assume infinite server semantics for transitions, the net of figure 4.1.a is isomorphic to a queueing network with delay stations, and in this case the cycle time is easily obtained as the sum of all the



(a) (b)

Figure 4.1: Queueing networks with: (a) only delay nodes and (b) delay and single-server nodes, represented by means of (a) a state machine and (b) a free choice net.

average service demands divided by the number of customers, because no queueing takes place at any node.

Now, let us consider the net of figure 4.1.b, in which stations represented by transitions t_4 and t_5 are single-servers instead of delay nodes (with queueing terminology) or, in other words, have their liveness bounds limited to one (with our notation). In this case, the elementary P-semiflows are $Y_1 = (1,1,1,1,0,0)^T$, $Y_2 = (0,0,0,0,1,0)^T$, and $Y_3 = (0,0,0,0,0,1)^T$, with $Y_1^T \cdot M_0 = 3$, $Y_2^T \cdot M_0 = 1$, and $Y_3^T \cdot M_0 = 1$. Therefore, the problem (LPP12) gives the value $\Gamma_{(1)}^{PS} = \max\{(1+1/2+1)/3, 1/2, 1\} = \max\{0.8333, 0.5, 1\} = 1$, and its inverse, which is the throughput upper bound, is also 1. In a queueing theory framework, the obtained bound is known as the *asymptotic throughput upper bound* [Kle76,DB78] and is obtained as the minimum between (a) the bound computed assuming that no queueing takes place at any node (0.8333, in this case), and (b) the maximum throughput of the bottleneck station (transition t_5 in the figure) which cannot have an *utilization rate* greater than 1.

In the general free choice nets case, the bound presented in theorem 4.1.1 can be interpreted as the maximum among the asymptotic bounds obtained for the isolated subnets generated by all the elemen-

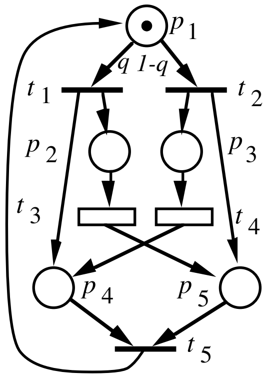


Figure 4.2: The throughput upper bound given by (LPP12) is non-reachable.

tary P-semiflows of the net.

Linear programming problems give an easy way to derive results and interpret them. Just looking at the problem (LPP12) the following monotonicity property is obtained, analogous to that obtained for marked graphs (property 3.1.1).

Corollary 4.1.2 *Let $\langle\mathcal{N}, M_0\rangle$ be a live and bounded free choice net and $\vec{s}$ the vector of average service times.*

i) For a fixed $\vec{s}$, if $M_0' \geq M_0$ (i.e., increasing the number of initial resources) then the lower bound for the mean cycle time of $\langle\mathcal{N}, M_0', \vec{s}\rangle$ is less than or equal to the one of $\langle\mathcal{N}, M_0, \vec{s}\rangle$ (i.e., $\Gamma_{(j)}^{PS'} \leq \Gamma_{(j)}^{PS}$).

ii) For a fixed M_0, if $\vec{s'} \leq \vec{s}$ (i.e., for faster resources) then the lower bound for the mean cycle time of $\langle\mathcal{N}, M_0, \vec{s'}\rangle$ is less than or equal to the one of $\langle\mathcal{N}, M_0, \vec{s}\rangle$ (i.e., $\Gamma_{(j)}^{PS'} \leq \Gamma_{(j)}^{PS}$).

Performance monotonicity does not hold for non-free choice nets increasing the number of initial resources, as was shown with the live net in figure 2.6 (for which the addition of one token makes it non-live).

For strongly connected marked graphs, the bound derived from theorem 4.1.1 has been shown to be reachable for arbitrary mean values and coefficients of variation associated with transition service times (theorem 3.1.3). Unfortunately, this is not the case for live and bounded free choice nets. Let us consider, for instance, the live and 1–bounded free choice net depicted in figure 4.2. Let s_3 and s_4 be the average service times associated with t_3 and t_4, respectively. Let t_1, t_2, and t_5 be *immediate* transitions (i.e., they fire in zero time). Let $q, 1-q \in (0,1)$ be the routing probabilities defining the resolution of conflict at place p_1. The vector of visit ratios normalized for t_5 is

$$\vec{v}^{(5)} = (q, 1-q, q, 1-q, 1)^T \tag{4.11}$$

The elementary P-semiflows are

$$\begin{array}{rcl} Y_1 & = & (1,1,0,0,1)^T \\ Y_2 & = & (1,0,1,1,0)^T \end{array} \tag{4.12}$$

Then, applying the problem (LPP12) to this net, the following lower bound for the mean cycle time of transition t_5 is obtained:

$$\Gamma_{(5)} \geq \max\{qs_3, (1-q)s_4\} \tag{4.13}$$

while the actual cycle time for this transition is

$$\Gamma_{(5)} = qs_3 + (1-q)s_4 \tag{4.14}$$

independently of the higher moments of the probability distribution functions associated with transitions t_3 and t_4. Therefore, the bound given by theorem 4.1.1 is non-reachable for the net in figure 4.2.

In the next section, we consider other linear marking relations, derived from the structural concept of trap, that can be used to improve the bound of theorem 4.1.1.

4.1.1.3 Little's law and traps

A trap in a Petri net $\mathcal{N}$ is a subset of places $\Theta \subseteq P$ such that $\Theta^\bullet \subseteq {}^\bullet\Theta$. A well-known property of these structural elements is recalled below.

Theorem 4.1.2 [Hac72] *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked Petri net and $\Theta \in P$ a trap. If Θ is initially marked, then Θ is marked throughout the net's evolution.*

This property can be expressed in algebraic terms considering the vector Y_Θ associated with a given trap Θ, and defined as $Y_\Theta(p) = \chi_\Theta(p)$, for all place p (we denote χ_Θ the *characteristic function of the set* Θ, i.e., $\chi_\Theta(p) = 1$ if $p \in \Theta$, and $\chi_\Theta(p) = 0$ otherwise). If $Y_\Theta^T \cdot M_0 \geq 1$ then $Y_\Theta^T \cdot M \geq 1$ for all marking M reachable from M_0.

Now let us consider the vector Y_Θ associated with a given trap Θ of a net, and a P-semiflow Y such that $Y - Y_\Theta \geq 0$ (it always exists for conservative nets). The following linear relation can be derived:

$$(Y - Y_\Theta)^T \cdot M \leq Y^T \cdot M_0 - 1 \tag{4.15}$$

for all marking M reachable from M_0 (thus the same relation holds for $\overline{M}$). Premultiplying inequality (4.1) by $Y - Y_\Theta$, the following lower bound for the mean cycle time of a transition t_1 is derived:

Theorem 4.1.3 *For any net $\mathcal{N}$ and for any trap Θ of $\mathcal{N}$, a lower bound for the mean cycle time $\Gamma_{(j)}$ of transition t_j is given by:*

$$\begin{array}{rll} \Gamma_{(j)} \geq \Gamma_{(j)}^{\Theta} = & \textit{maximize} & \dfrac{(Y - Y_\Theta)^T \cdot PRE \cdot \vec{D}^{(j)}}{Y^T \cdot M_0 - 1} \\ & \textit{subject to} & Y^T \cdot C = 0 \\ & & Y - Y_\Theta \geq 0 \\ & & Y_\Theta(p) = \chi_\Theta(p), \quad \forall p \in P \end{array} \tag{4.16}$$

In the next section we derive a linear programming problem for the computation of an improvement of the previous bound based on the concept of *implicit place*.

Going back to the net in figure 4.2, the unique minimal trap different from the P-semiflows is

$$\Theta = \{p_1, p_4, p_5\} \tag{4.17}$$

Considering the P-semiflow

$$Y = (2, 1, 1, 1, 1)^T \tag{4.18}$$

we have

$$Y \geq Y_\Theta = (1, 0, 0, 1, 1)^T \tag{4.19}$$

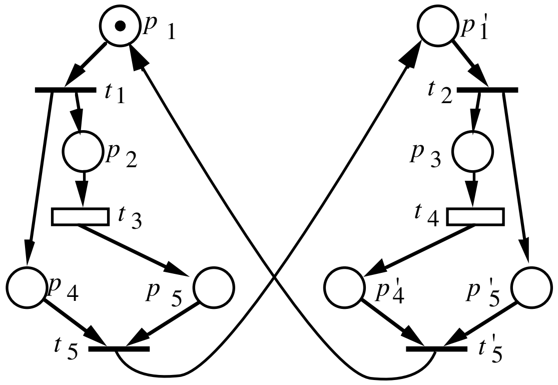


Figure 4.3: Behaviourally equivalent 1–bounded marked graph of the net in figure 4.2 for deterministic resolution of conflict and $q = 1/2$.

and theorem 4.1.3 can be applied:

$$\Gamma_{(5)} \geq q s_3 + (1-q) s_4 \tag{4.20}$$

Therefore the bound obtained in the previous section using only P-semiflows has been improved for the example (in fact the bound computed now is tight for this example, i.e., it coincides with the actual cycle time).

In order to explain in an intuitive way the reason of the previous improvement, let us derive a behaviourally equivalent 1–bounded marked graph (figure 4.3) for the free choice net of figure 4.2, assuming for the sake of simplicity that the resolution of conflict at place p_1 is deterministic with $q = 1/2$ (i.e., transitions t_1 and t_2 fire once each one, alternatively). The lower bound for the mean cycle time of this marked graph based on theorem 4.1.1 (i.e., using the P-semiflows) is $\Gamma_{\mathrm{MG}} \geq s_3 + s_4$ (in fact it is reached) and corresponds to the circuit $\langle p_1, p_2, p_5, p'_1, p_3, p'_4 \rangle$. Since transition t_5 appears instantiated twice in the marked graph, the obtained bound for the mean cycle time of this transition is $\Gamma_{(5)} \geq (s_3 + s_4)/2$. In the original free choice net there does not exist any minimal P-semiflow including both p_2 and p_3 in its support, thus the previous bound is not obtained. In other words, in this net there exists more linear information about the marking than the one derived from P-semiflows, that can be obtained from other

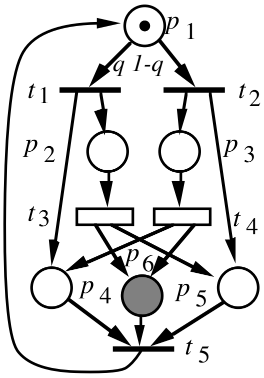


Figure 4.4: The net of figure 4.2 with the addition of the implicit place p_6.

structural relations (that of traps).

4.1.2 A new perspective: implicit places

In this section we recall the concept of *implicit place.* It allows us to reinterpret the improvement of the upper bound on throughput presented in the previous section, using the P-semiflows of a derived net instead of the trap structures in the original one.

4.1.2.1 Implicit places

An implicit place (see section 1.2.2) is one which never is the unique place that restricts the firing of its output transitions. Let $\mathcal{N}$ be a net and $\mathcal{N}^p$ be the net resulting from adding a place p to $\mathcal{N}$. If M_0 is an initial marking of $\mathcal{N}$, M_0^p denotes the initial marking of $\mathcal{N}^p$, i.e., $M_0^p(p') = M_0(p')$ for all $p' \neq p$.

Definition 4.1.1 (Implicit place) *Given a net $\langle \mathcal{N}^p, M_0^p \rangle$, the place p is implicit iff $L(\mathcal{N}^p, M_0^p) = L(\mathcal{N}, M_0)$ (i.e., it preserves the firing sequences).*

As an example let us consider the net in figure 4.4. Place p_6 is implicit since its elimination does not change the firing sequences of the net (i.e., its firing sequences coincide with those of the net in figure 4.2).

Implicit places are behaviourally defined. The structural counterpart of this concept is recalled in the next definition.

Definition 4.1.2 (Structurally implicit place) [CS89c] *Given a net $\mathcal{N}^p$, the place p is structurally implicit iff for all initial marking M_0 of $\mathcal{N}$ there exists an $M_0^p(p)$ such that p is an implicit place in $\langle \mathcal{N}^p, M_0^p \rangle$.*

In [CS89c], linear programming techniques are used for deriving necessary and sufficient conditions for a place to be structurally implicit as well as for computing an upper bound of the minimum initial marking that makes it implicit.

Theorem 4.1.4 [CS89c]

1. *A place p is structurally implicit in $\mathcal{N}^p$ iff $\exists Y \geq 0$ such that $Y^T \cdot C \leq l_p$, where C is the incidence matrix of $\mathcal{N}$ and l_p is the incidence vector of place p in $\mathcal{N}^p$.*
2. *Let p be a structurally implicit place and let define*

$$\begin{array}{rll} w \stackrel{\text{def}}{=} & \textit{minimize} & Y^T \cdot M_0 + \mu \\ & \textit{subject to} & Y^T \cdot C \leq l_p \\ & & Y^T \cdot PRE[t_k] + \mu \geq pre(p, t_k), \ \forall t_k \in p^\bullet \\ & & Y \geq 0 \end{array} \tag{LPP13}$$

If the initial marking $M_0^p(p)$ of place p is such that $M_0^p(p) \geq w$ then p is implicit.

As it is remarked in [CS89c], the previous theorem allows to detect stuctural implicit places (p is structurally implicit iff (LPP13) has a feasible solution) and implicit places (if the initial marking of p is greater than or equal to that computed by (LPP13)) in polynomial time.

The reason for the introduction of implicit places in this work and their relation with the linear marking inequalities presented above are explained in the next sections.

4.1.2.2 Reinterpretation of traps using implicit places

Let us consider once more the net in figure 4.2 and its behaviourally equivalent (for $q = 1/2$) marked graph depicted in figure 4.3. The elementary circuits (P-semiflows) of this marked graph are

$$\begin{array}{rcl} c_1 & = & \langle p_1, p_2, p_5, p'_1, p'_5 \rangle \\ c_2 & = & \langle p_1, p_4, p'_1, p_3, p'_4 \rangle \\ c_3 & = & \langle p_1, p_2, p_5, p'_1, p_3, p'_4 \rangle \\ c_4 & = & \langle p_1, p_4, p'_1, p'_5 \rangle \end{array} \tag{4.21}$$

The circuits c_1 and c_2 correspond with the elementary P-semiflows of the original net Y_1 and Y_2 (4.12), respectively. Thus, these circuits cannot contribute to the improvement of the bound computed for the original net based on the P-semiflows. This is not the case for the circuits c_3 and c_4. These circuits add linear information which is not reflected by P-semiflows in the original net. The circuit c_4 does not include any timed transition and must not be considered. On the other hand, the circuit c_3 reflects the sequentialization of transitions t_3 and t_4, and it gives the actual cycle time of the net.

A given elementary circuit of the derived marked graph does not correspond with any elementary P-semiflow of the original free choice net when it includes several instances of a unique transition and each instance has as input (or output) places which are instances of different original places. This is the case, for example, for the circuit c_3 of the marked graph of figure 4.3. It includes instances t_5 and t'_5 of a unique transition, and the input places of these transitions in circuit c_3 are p_5 and p'_4, respectively, which are instances of different original places.

Now, let us increment the number of circuits of the marked graph of figure 4.3, by adding the places p_6 and p'_6 as it is depicted in figure 4.5. Places p_6 and p'_6 are replicas of places p_5 and p'_4, respectively (thus they are implicit), and can be supposed to be different instances of a new (implicit) place in the original net (place p_6 of the net in figure 4.4). The addition of this place to the net in figure 4.2 generates a new elementary P-semiflow

$$Y_3 = (1, 1, 1, 0, 0, 1)^T \tag{4.22}$$

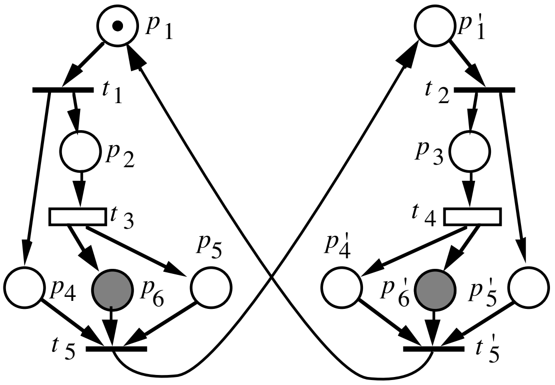


Figure 4.5: Addition of the implicit places p_6 and p_6' to the marked graph of figure 4.3.

With this P-semiflow, the lower bound for the mean cycle time computed with problem (LPP12) is

$$\Gamma_{(5)} \geq q s_3 + (1-q) s_4 \tag{4.23}$$

which is the same previously obtained using relations derived from trap structures (and, in fact, it is the actual cycle time).

Let us remark that the relation between the implicit place p_6 of the net in figure 4.4 and the trap $\Theta = \{p_1, p_4, p_5\}$ considered before is straightforward: $l_{p_6} = Y_\Theta \cdot C$, that is, the incidence vector of p_6 is the sum of those of places p_1, p_4, and p_5.

The following linear relation can be derived from the trap Θ (and the P-semiflow $Y = Y_1 + Y_2$):

$$(Y - Y_\Theta)^T \cdot M = M(p_1) + M(p_2) + M(p_3) \leq 1, \ \ \forall M \in R(\mathcal{N}, M_0) \tag{4.24}$$

Alternatively, this one follows from the new P-semiflow Y_3 that includes the implicit place p_6:

$$Y_3^T \cdot M = M(p_1) + M(p_2) + M(p_3) + M(p_6) = 1, \ \ \forall M \in R(\mathcal{N}, M_0) \tag{4.25}$$

It can be pointed out that the information given by relation (4.24) is included in that given by the new P-semiflow (equation (4.25)).

In the next section, technical details related with the addition of implicit places which improve the throughput upper bound computed by means of P-semiflows and traps are considered.

4.1.2.3 Implicit places improve traps-based bounds

Let us consider an initially marked trap Θ of a given net $\mathcal{N}$, and its associated vector Y_Θ defined as in previous sections. The following result, which follows from theorem 4.1.4.1, assures that a structurally implicit place p_Θ associated with Θ, can be added to $\mathcal{N}$.

Corollary 4.1.3 *Let Θ be an initially marked trap of $\mathcal{N}$, $Y_\Theta(p) = \chi_\Theta(p)$ for all place p, $Y_\Theta^T \cdot M_0 \geq 1$, and $\mathcal{N}^{p_\Theta}$ the net resulting from the addition of place p_Θ with incidence vector $l_{p_\Theta} = Y_\Theta^T \cdot C$ to $\mathcal{N}$. Then p_Θ is structurally implicit in $\mathcal{N}^{p_\Theta}$.*

The importance of the previous structural implicit place lies in the fact that, if a marking makes it implicit (e.g., the marking given by theorem 4.1.4.2), then the lower bound for the mean cycle time of a transition computed using P-semiflows of the augmented net can improve the bounds based on P-semiflows of the original net (theorem 4.1.1) and on the trap Θ (theorem 4.1.3). This result is stated in theorem 4.1.5. We firstly present a technical lemma.

Lemma 4.1.1 *Let Θ be an initially marked trap of $\mathcal{N}$, $Y_\Theta(p) = \chi_\Theta(p)$ for all place p. Let p_Θ be a place defined as $l_{p_\Theta} = Y_\Theta^T \cdot C$. Then Y_Θ, $\mu_\Theta = -1$ is a feasible solution of the problem (LPP13) and $M_0^{p_\Theta}(p_\Theta) \leq Y_\Theta^T \cdot M_0 + \mu_\Theta$ (place p_Θ is assumed to be pure, i.e., selfloop-free).*

Proof. Since $Y_\Theta^T \cdot C = l_{p_\Theta}$, then $\forall t \in {}^\bullet p_\Theta : Y_\Theta^T \cdot POST[t] - Y_\Theta^T \cdot PRE[t] = -pre(p_\Theta, t)$. Taking into account that $Y_\Theta(p)$ is the characteristic function of a trap we have in the last equality that $Y_\Theta^T \cdot PRE[t] > 0$ if and only if $Y_\Theta^T \cdot POST[t] > 0$. Therefore, $\forall t \in {}^\bullet p_\Theta$: $Y_\Theta^T \cdot PRE[t] > pre(p_\Theta, t)$ and from the problem (LPP13) we conclude that Y_Θ and $\mu_\Theta = -1$ are a feasible solution. From this linear programming problem we also conclude directly that $M_0^{p_\Theta}(p_\Theta) \leq Y_\Theta^T \cdot M_0 + \mu_\Theta$ because $Y_\Theta^T \cdot M_0 \geq 1$. ■

Theorem 4.1.5 *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked net, Θ an initially marked trap of $\mathcal{N}$, $Y_\Theta(p) = \chi_\Theta(p)$ for all place p, and $\langle \mathcal{N}^{p_\Theta}, M_0^{p_\Theta} \rangle$ the marked net resulting from the addition to the original net of the structural implicit place p_Θ with incidence vector $l_{p_\Theta} = Y_\Theta \cdot C$ and with $M_0^{p_\Theta}(p_\Theta)$ given by theorem 4.1.4.2. Then a lower bound $\Gamma_{(j)}^{p_\Theta}$ for the mean cycle time $\Gamma_{(j)}$ of transition t_j in $\langle \mathcal{N}, M_0 \rangle$ can be computed applying theorem 4.1.1 to the net $\langle \mathcal{N}^{p_\Theta}, M_0^{p_\Theta} \rangle$.*

Moreover, if $\Gamma_{(j)}^{PS}$ and $\Gamma_{(j)}^{\Theta}$ are the lower bounds of $\Gamma_{(j)}$ derived from the direct application of theorems 4.1.1 and 4.1.3, respectively, to the original net, then $\Gamma_{(j)}^{p_\Theta} \geq \Gamma_{(j)}^{PS}$ and $\Gamma_{(j)}^{p_\Theta} \geq \Gamma_{(j)}^{\Theta}$.

Proof. $\Gamma_{(j)}^{p_\Theta}$ is a lower bound for the mean cycle time of t_j in $\langle \mathcal{N}^{p_\Theta}, M_0^{p_\Theta} \rangle$ by theorem 4.1.1. Since p_Θ is implicit, t_j has the same mean cycle time in $\langle \mathcal{N}, M_0 \rangle$ and in $\langle \mathcal{N}^{p_\Theta}, M_0^{p_\Theta} \rangle$. Then, $\Gamma_{(j)}^{p_\Theta}$ is a lower bound for the mean cycle time of t_j in $\langle \mathcal{N}, M_0 \rangle$.

$\Gamma_{(j)}^{p_\Theta} \geq \Gamma_{(j)}^{PS}$ because if Y is a P-semiflow of $\mathcal{N}$, then $Z = [Y^T|0]^T$ is a P-semiflow of $\mathcal{N}^{p_\Theta}$.

Finally, we prove that $\Gamma_{(j)}^{p_\Theta} \geq \Gamma_{(j)}^{\Theta}$. Let Y be a P-semiflow of $\mathcal{N}$ such that $Y - Y_\Theta \geq 0$. Then $Z = [(Y - Y_\Theta)^T|1]^T$ is a P-semiflow of $\mathcal{N}^{p_\Theta}$. Now, applying equation (4.10) for $\Gamma_{(j)}^{p_\Theta}$:

$$\begin{aligned}
\Gamma_{(j)}^{p_\Theta} &\geq \frac{[(Y - Y_\Theta)^T|1] \cdot PRE^{p_\Theta} \cdot \vec{D}^{(j)}}{Y^T \cdot M_0 - Y_\Theta^T \cdot M_0 + M_0^{p_\Theta}(p_\Theta)} = \\
&= \frac{(Y - Y_\Theta)^T \cdot PRE \cdot \vec{D}^{(j)}}{Y^T \cdot M_0 - Y_\Theta^T \cdot M_0 + M_0^{p_\Theta}(p_\Theta)} + \\
&\quad + \frac{pre(p_\Theta) \cdot \vec{D}^{(j)}}{Y^T \cdot M_0 - Y_\Theta^T \cdot M_0 + M_0^{p_\Theta}(p_\Theta)}
\end{aligned} \tag{4.26}$$

And this value is greater than or equal to that given by equation (4.16) in theorem 4.1.3 because the second term of the above sum is non-negative and the first term in the above sum is greater than or equal to that given by equation (4.16) in theorem 4.1.3 (taking into account that $M_0^{p_\Theta}(p_\Theta) \leq Y_\Theta^T \cdot M_0 - 1$, by lemma 4.1.1, and then the denominator is less than or equal to $Y^T \cdot M_0 - 1$). ∎

In the previous section, the net of figure 4.2 is considered as an example in which the bound computed using the trap $\Theta = \{p_1, p_4, p_5\}$

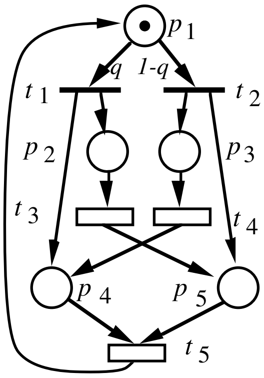


Figure 4.6: The throughput upper bounds given by theorems 4.1.1 and 4.1.3 are non-reachable, while the bound given by theorem 4.1.5 is reached.

is tight because it reaches the actual value of the mean cycle time. It is also shown that the same value can be derived, after the addition of the associated implicit place p_6 (figure 4.4), considering the new P-semiflow $(1, 1, 1, 0, 0, 1)^T$.

Let us consider the same net of figure 4.2, but assuming now that transition t_5 is not immediate but timed, with average service time equal to s_5 (as depicted in figure 4.6). If problem (LPP12) is applied to the net, the following bound is obtained:

$$\Gamma_{(5)}^{PS} = \max\{qs_3 + s_5, (1-q)s_4 + s_5\} \tag{4.27}$$

If trap $\Theta = \{p_1, p_4, p_5\}$ is considered, theorem 4.1.3 does not improve the above bound, since it gives the value:

$$\Gamma_{(5)}^{\Theta} = qs_3 + (1-q)s_4 \tag{4.28}$$

Finally, if the implicit place p_Θ associated with Θ is added to the net, theorem 4.1.5 gives the bound:

$$\Gamma_{(5)}^{p_\Theta} = qs_3 + (1-q)s_4 + s_5 \tag{4.29}$$

(for the P-semiflow $(1, 1, 1, 0, 0, 1)^T$), which improves both $\Gamma_{(5)}^{PS}$ and $\Gamma_{(5)}^{\Theta}$, and, in fact, it gives the actual cycle time of transition t_5 (i.e., it is tight

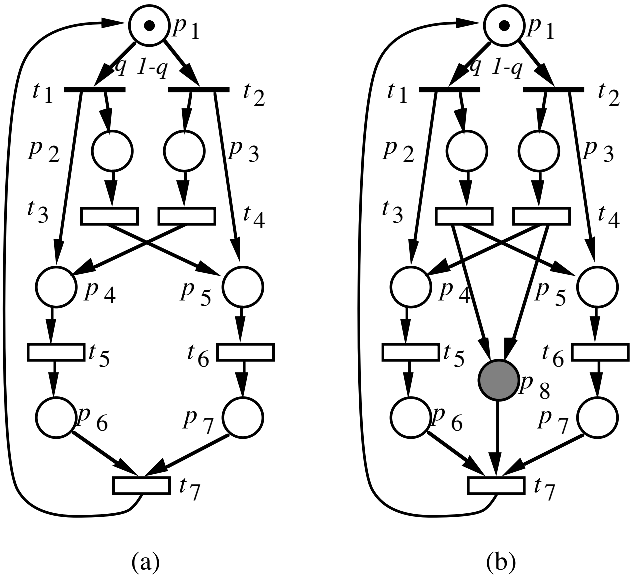


Figure 4.7: The throughput upper bounds given by theorems 4.1.1, 4.1.3, and 4.1.5 are non-reachable.

for this example). Note that, in this case, the improvement is due to the non-null second term of the expresion (4.26).

Unfortunately, the bound given by theorem 4.1.5 is not reachable in all cases. As an example, let us consider the net of figure 4.7.a. The lower bound for the mean cycle time of transition t_7 given by theorem 4.1.1 is

$$\Gamma^{PS}_{(7)} = \max\{qs_3 + s_6, (1-q)s_4 + s_5\} + s_7 \tag{4.30}$$

If the unique elementary trap of the net (different from the P-semiflows) is considered, $\Theta = \{p_1, p_4, p_5, p_6, p_7\}$, the application of theorem 4.1.3 gives the bound:

$$\Gamma^{\Theta}_{(7)} = qs_3 + (1-q)s_4 \tag{4.31}$$

Finally, if the implicit place p_8 associated with Θ is added to the net (see figure 4.7.b), theorem 4.1.5 gives the bound:

$$\Gamma^{p_\Theta}_{(7)} = \max\{qs_3 + s_6, (1-q)s_4 + s_5, qs_3 + (1-q)s_4\} + s_7 \tag{4.32}$$

While the lowest mean cycle time, which is reached for deterministic timing, is

$$\begin{array}{rcl}\Gamma_{(7)} & = & q\max\{s_5, s_3+s_6\}+(1-q)\max\{s_4+s_5, s_6\}+s_7 = \\ & = & \max\{\; qs_3+s_6, \\ & & \qquad (1-q)s_4+s_5, \\ & & \qquad qs_3+(1-q)s_4+(1-q)s_5+qs_6, \\ & & \qquad qs_5+(1-q)s_6\}+ \\ & & +s_7 \end{array} \tag{4.33}$$

and it is clearly greater than the value $\Gamma_{(7)}^{p\ominus}$. Therefore, for the net in figure 4.7.a, the lower bound for the mean cycle time given by theorem 4.1.5 is non-reachable.

In the next section, a new method for the computation of a reachable throughput upper bound for live and 1–bounded free choice nets is presented, based on the structural concept of *multiset of circuits.*

4.1.3 Multisets of circuits: derivation of a reachable upper bound

In this section, let us consider live and 1–bounded free choice nets with arbitrary service times associated with transitions. The conflicts resolution policy is also arbitrary, but with some given routing rates. In fact, without loss of generality, we can restrict to deterministic resolution policies, which for 1–bounded free choice nets give the same performance than any probabilistic routing, in steady-state.

First, we give an algorithm to derive a live and 1–bounded marked graph which is behaviourally equivalent to the live and 1–bounded free choice net with deterministic routing. For this marked graph, the results presented in chapter 3 can be applied for the computation of bounds. After that, we interpret the computation of bounds for the behaviourally equivalent marked graph considering some collections of circuits, or *multisets* of the original net.

A deterministic resolution of the conflict between two transitions t_1 and t_2 is a rule that fixes which transition of them will be authorized to fire at the successive markings enabling both. Thus, in some sense, the resulting *interpreted net* can be considered as a *conflict-free net.*

In the next paragraph, we present an algorithm for the computation of bounds for a live and 1–bounded free choice net $\langle \mathcal{N}, M_0 \rangle$ with deterministic routing, based on the fact that the behaviour of a 1–bounded conflict-free net can be represented by means of an equivalent marked graph [Ram74], for which the results of chapter 3 can be applied.

Step 0. From the given deterministic resolution policy, compute the vector of visit ratios $\vec{v}^{(j)}$ in the net system $\langle \mathcal{N}, M_0 \rangle$, using the theorem 2.1.2.

Step 1. Steady-state markings must be *home states*. Let M_h be one of the home states (there always exist some for live and bounded free choice nets, according to theorem 2.1.11), and select it as the initial marking (i.e., $\langle \mathcal{N}, M_h \rangle$ is *reversible*).

Step 2. Apply the algorithm presented in [Ram74], with the initial marking of Step 1, in order to compute a behaviourally equivalent marked graph of 1–bounded conflict-free nets, with the following modifications: (1) each time one place enables more than one transition, select the transition authorized by the deterministic resolution policy; (2) select one *slice* [Ram74] of the *behaviour graph* (among those that occur repeatedly, according to [Ram74, lemma 3.4.2]) including the same places marked at the initial home state and such that the number of instances of each transition in the *frustum* [Ram74] is the same multiple of its corresponding entry in the vector of visit ratios.

Step 3. Compute the lower bound for the mean cycle time of the marked graph obtained in Step 2, in which all instances of a same transition have a service time equal to that of the original one. The computation is made by solving the linear programming problem (LPP3). (Observe that the vector of visit ratios for the marked graph is $\vec{v} = \mathbb{1}$, but the number of instances of each transition is equal to the same multiple of its corresponding entry in the vector of visit ratios computed in Step 0.)

Step 4. A lower bound for the mean cycle time of a given transition in the original net is computed by dividing the value obtained in

Step 3 by the number of instances of this transition in the derived equivalent marked graph.

Observe that the smallest behaviourally equivalent marked graph that can be derived with previous algorithm is obtained by firing a sequence of transitions of the original net whose firing count vector is a multiple of the vector of visit ratios (let us denote as $\vec{v}$) such that: all its components are integer and their greatest common divisor es equal to 1. On the other hand, from the deterministic routing assumption follows that the only repetitive sequences of the interpreted net are such having a multiple of the vector of visit ratios as firing count vector. Therefore, for a given transition, the number of instances of it in the behaviourally equivalent marked graph is equal to its corresponding entry in vector $\vec{v}$. This is the reason why in Step 4 of above algorithm the value obtained from (LPP3) in Step 3 is divided by the number of instances of the considered transition.

It must be pointed out that since the marked graph derived in Step 2 is behaviourally equivalent to the original free choice net with deterministic conflict resolution policy then, in particular, their exact mean cycle times are equal. Therefore, the lower bound for the mean cycle time of transitions of the original free choice net (with the given deterministic conflicts resolution policy) can be derived from the mean cycle time of the marked graph, after a normalization operation (dividing by the number of instances of the selected transition).

The bound computed for the marked graph by means of (LPP3) given by theorem 3.1.1 is reachable (see theorem 3.1.3). This provides a method for the computation of a reachable lower bound for the mean cycle time of transitions for live and 1–bounded free choice nets.

As an example, let us consider the live and 1–bounded free choice net depicted in figure 4.8. Suppose that the deterministic conflict resolution policy at place p_1 is: "authorize twice transition t_1, then once transition t_2, and repeat it" (i.e., the routing rate for transition t_1 is twice the routing rate of t_2). The application of Step 0 gives the vector $\vec{v}^{(7)} = (2/3, 1/3, 2/3, 1/3, 1, 1, 1)^T$. The initial marking is already a home state and therefore the output of Step 1 is verified. The application of Step 2 gives the behaviourally equivalent marked graph depicted in figure 4.9. The lower bound for the mean cycle time of the derived marked graph

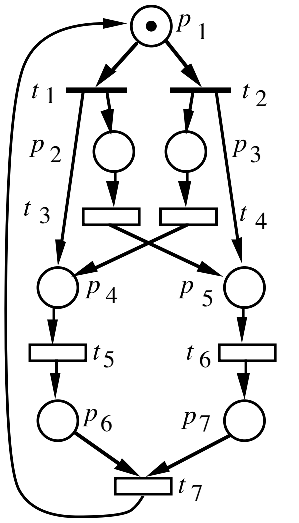


Figure 4.8: A live and 1–bounded free choice net.

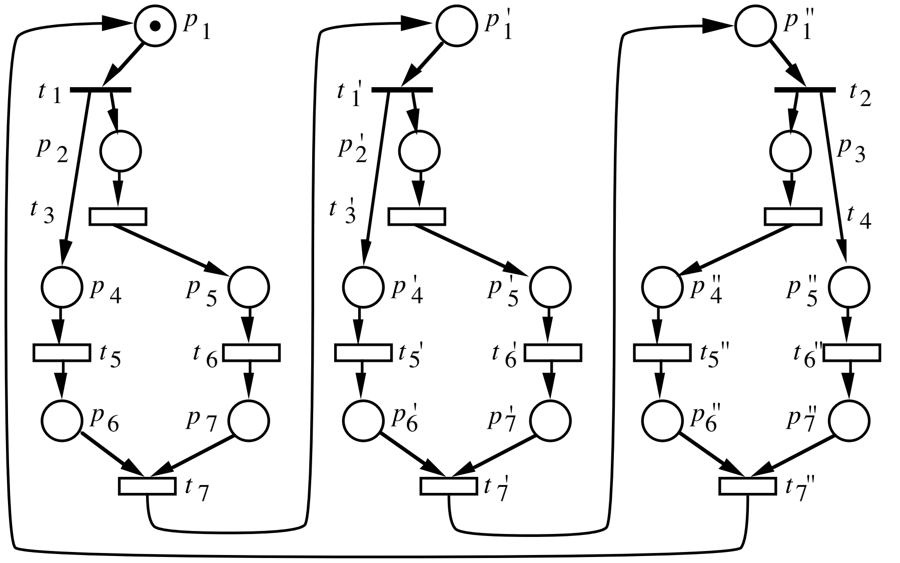


Figure 4.9: Behaviourally equivalent marked graph of the net in figure 4.8 for a deterministic resolution of conflict (the routing associated with t_1 is equal to twice the routing associated with t_2).

(which is equal for all its transitions) is, according to Step 3, equal to:

$$\begin{aligned}\Gamma_{(7)} \;=\; \max\{\; & 2s_3 + 3s_6 + 3s_7,\\ & s_4 + 3s_5 + 3s_7,\\ & 2s_3 + s_4 + s_5 + 2s_6 + 3s_7,\\ & 2s_5 + s_6 + 3s_7\}\end{aligned} \tag{4.34}$$

Finally, the application of Step 4 (divide by the number of instances of transition t_7) leads to a lower bound for the mean cycle time of transition t_7 in the free choice net of figure 4.8 equal to the value given by (4.33) with $q = 2/3$.

As can be seen in the example, the proposed method can become very expensive in amount of memory and computational time. Just consider, as an example, the net in figure 4.8 but with routing rates of transitions t_1 and t_2 being 21 and 11, only a bit different from the considered before (2 and 1, respectively). In this case the equivalent marked graph would have 160 transitions and 192 places!

The next paragraphs of this section are devoted to analyze the method to compute the lower bound for the mean cycle time of the equivalent marked graph derived from the original 1–bounded free choice net. This is done in order to translate the underlying structural property to an equivalent structural property on the free choice net. This property is used in the last part of the section to derive a polynomial method to compute the lower bound of the mean cycle time for the original net without the generation of the marked graph.

It has been shown (theorem 3.1.2) that the problem of finding a lower bound for the mean cycle time of transitions in a strongly connected stochastic marked graph can be solved looking at the cycle times associated with each minimal P-semiflow (circuits for marked graphs) of the net, considered in isolation (in fact, the simplex method used to solve the problem of Step 3 proceeds in this way). These cycle times can be computed making the summation of the average service times of all the transitions involved in the P-semiflow, and dividing by the number of tokens present in it. Therefore, the performance computation of Step 3 looks at the circuits of the behaviourally equivalent marked graph.

On the other hand, a circuit of the marked graph is composed by one or several instances of circuits of the original free choice net. This collection of circuits, including one or several instances of each circuit, is called *multiset* of circuits, and will be formally defined below.

For the previous example, these multisets are

$$\begin{array}{lcl}
\mathcal{M}_1 & = & \{\langle p_1t_1p_4t_5p_6t_7\rangle, \langle p_1t_1p_4t_5p_6t_7\rangle, \langle p_1t_2p_3t_4p_4t_5p_6t_7\rangle\} \\
\mathcal{M}_2 & = & \{\langle p_1t_1p_4t_5p_6t_7\rangle, \langle p_1t_1p_4t_5p_6t_7\rangle, \langle p_1t_2p_5t_6p_7t_7\rangle\} \\
\mathcal{M}_3 & = & \{\langle p_1t_1p_2t_3p_5t_6p_7t_7\rangle, \langle p_1t_1p_2t_3p_5t_6p_7t_7\rangle, \langle p_1t_2p_3t_4p_4t_5p_6t_7\rangle\} \\
\mathcal{M}_4 & = & \{\langle p_1t_1p_2t_3p_5t_6p_7t_7\rangle, \langle p_1t_1p_2t_3p_5t_6p_7t_7\rangle, \langle p_1t_2p_5t_6p_7t_7\rangle\} \\
\mathcal{M}_5 & = & \{\langle p_1t_1p_2t_3p_5t_6p_7t_7\rangle, \langle p_1t_1p_4t_5p_6t_7\rangle, \langle p_1t_2p_3t_4p_4t_5p_6t_7\rangle\} \\
\mathcal{M}_6 & = & \{\langle p_1t_1p_2t_3p_5t_6p_7t_7\rangle, \langle p_1t_1p_4t_5p_6t_7\rangle, \langle p_1t_2p_5t_6p_7t_7\rangle\}
\end{array}$$

The reader can notice that multisets $\mathcal{M}_5$ and $\mathcal{M}_6$ (that we will call *non-minimal*) need not to be considered in order to obtain the slowest path because if $\langle p_1t_1p_2t_3p_5t_6p_7t_7\rangle$ is selected for the first time, it will be selected again instead of $\langle p_1t_1p_4t_5p_6t_7\rangle$. We remark also that circuits $\langle p_1t_1p_4t_5p_6t_7\rangle$ and $\langle p_1t_1p_2t_3p_5t_6p_7t_7\rangle$ appear twice in multisets $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$, and $\mathcal{M}_4$, while circuits $\langle p_1t_2p_3t_4p_4t_5p_6t_7\rangle$ and $\langle p_1t_2p_5t_6p_7t_7\rangle$ appear only once, according to the routing rates associated with transitions t_1 and t_2. As an example, multiset $\mathcal{M}_2$ is depicted in figure 4.10. It can be interpreted as a path as follows: (1) a token, initially placed at p_1 enables transitions t_1 and t_2; (2) transition t_1 is authorized for firing, according to the given conflict resolution policy; (3) after the firing of t_1, the token splits into two tokens; (4) we follow one of them; for instance, the one that places at p_4; (5) after the firing of t_5 and t_7, it returns to p_1; (6) according to the conflict resolution policy, t_1 is authorized once more; (7) we follow the same path than in steps (4) and (5), until the token returns to p_1 again; (8) now, transition t_2 is autorized; (9) the path $\langle t_2p_5t_6p_7t_7\rangle$ is followed; (10) the situation now is the same than in step (1), so the previous steps can be executed ad infinitum.

The mean cycle time of execution of previous path if the multiset of circuits $\mathcal{M}_2$ is considered in isolation is equal to the execution time of the corresponding isolated circuit $\langle p_1t_1p_4t_5p_6t_7p'_1t'_1p'_4t'_5p'_6t'_7p''_1t_2p''_5t''_6p''_7t''_7\rangle$ of the marked graph depicted in figure 4.9, and it is, in general, a lower bound for the exact mean cycle time. Therefore, a lower bound for the mean cycle time can be computed taking the maximum among the mean cycle time of execution of those multisets of circuits satisfying the routing rates, considered in isolation. In the particular case of marked graphs (considered in the previous chapter), since no decision exists for such nets, multisets of circuits were reduce to circuits, and these could

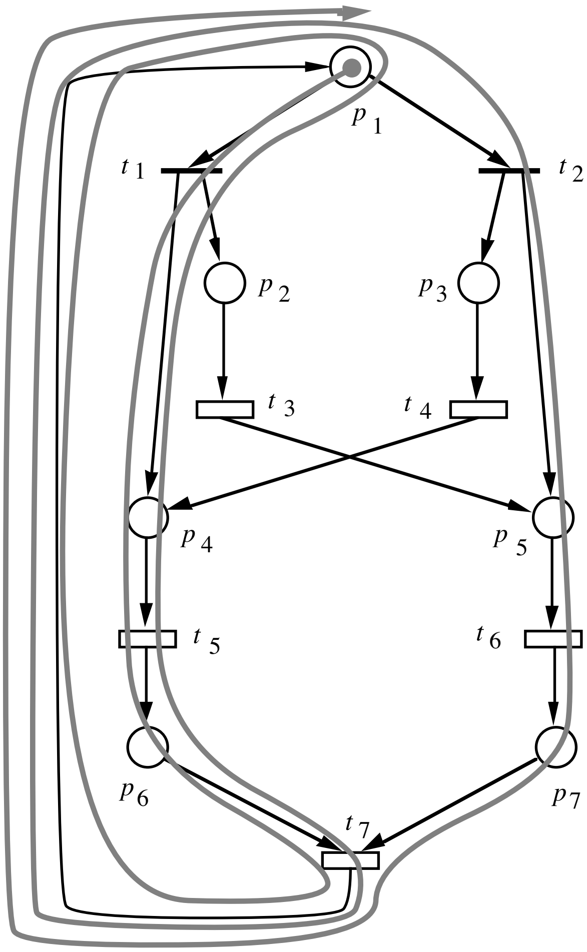


Figure 4.10: A multiset of circuits of the net in figure 4.8.

be algebraically characterized as P-semiflows of the net (see theorems 2.1.6.1 and 3.1.1).

The next step is to construct another net (with only a linear size increase of the original size) for which the previously considered multisets of circuits of live and 1–bounded free choice nets can be algebraically characterized (in fact, computed as P-semiflows). This can be done in a similar way to that presented in [Lau87] for the polynomial computation of the *minimal traps* of a net. In the rest of this section we formalize the concept of multiset of circuits and present a net transformation for the efficient computation of mean cycle time on multisets.

A *multiset* is a collection of elements that may contain several copies of an element. More formally, if $\mathcal{S}$ is a set, a multiset $\mathcal{M}$ of elements of $\mathcal{S}$ is an application $\mathcal{M} : \mathcal{S} \to \{0, 1, \ldots\}$.

If $\mathcal{N} = \langle P, T, Pre, Post \rangle$ is a Petri net and $\mathcal{M}$ a multiset of circuits of $\mathcal{N}$ (in what follows of this section we write circuit instead of minimal circuit):

- $\mathcal{M}(y)$ denotes the number of circuits of $\mathcal{M}$ which pass through the node $y \in P \cup T$.

- $\mathcal{M}(p, t)$ (respectively $\mathcal{M}(t, p)$) denotes the number of circuits of $\mathcal{M}$ which pass through the arc (p, t) (respectively (t, p)), if $Pre(p, t) > 0$ (respectively $Post(p, t) > 0$).

In the next definitions, we limit the class of multisets of circuits to those that will correspond exactly with the circuits of the behaviourally equivalent marked graph that can give the optimum of the problem (LPP3).

Definition 4.1.3 ($\mathcal{R}$–multiset of circuits) *Let $\mathcal{N} = \langle P,\ T,\ Pre,\ Post \rangle$ be a net, $\mathcal{R}$ the definition of routing rates at conflicts, and $\mathcal{M}$ a non empty multiset of circuits of $\mathcal{N}$. $\mathcal{M}$ is called an $\mathcal{R}$–multiset of circuits iff for all $p \in P$ such that $|p^\bullet| > 1$ and $\mathcal{M}(p) > 0$: $r_j \mathcal{M}(p, t_i) = r_i \mathcal{M}(p, t_j)$, for all $t_i, t_j \in p^\bullet$, where r_i and r_j are the routing rates of transitions t_i and t_j in the conflict at place p.*

$\mathcal{R}$–multiset of circuits will be abreviated to $\mathcal{R}$–mc. The above definition constraints the multiset to contain the different circuits of the net according to the visit ratios for transitions derived from the routing rates at conflicts.

We define now the concept of set of nodes covered by an $\mathcal{R}$–mc and introduce *minimal* $\mathcal{R}$–mcs.

Definition 4.1.4 (Support of $\mathcal{R}$–multisets of circuits) *The support of an $\mathcal{R}$–mc $\mathcal{M}$ is the set of nodes $||\mathcal{M}|| \subseteq P \cup T$ covered by $\mathcal{M}$.*

Definition 4.1.5 (Minimal $\mathcal{R}$–multisets of circuits) *An $\mathcal{R}$–mc is called minimal iff:*

a) its support does not contain the support of an $\mathcal{R}$–mc as a proper subset and

b) if $\mathcal{M}'$ is an $\mathcal{R}$–mc with $||\mathcal{M}'|| = ||\mathcal{M}||$, then $\mathcal{M}(y) \leq \mathcal{M}'(y)$, $\forall y \in P \cup T$.

In the above definition we denote the support with the same symbol than the support of a vector, as they are closely related concepts. We consider that the context eliminates confusion.

The consideration of minimal $\mathcal{R}$–mc discards the possibility of including two circuits that contain different output places of a *fork* transition (a transition with more than one output place). This constraint is not a problem in order to find the slowest path (with deterministic service times) because if a given output place of the fork transition is selected for the first time, it must be selected also the rest of the times.

Lemma 4.1.2 *Let $\langle \mathcal{N}, M_0 \rangle$ be a live and 1–bounded free choice net with deterministic conflict resolution policy, $\langle \mathcal{N}^{MG}, M_0^{MG} \rangle$ its behaviourally equivalent marked graph derived in previous paragraphs of this section, and $\mathcal{M}$ a multiset of circuits of $\mathcal{N}$. $\mathcal{M}$ is a minimal $\mathcal{R}$–mc of $\mathcal{N}$ iff*

1. *there exists a circuit (minimal P-semiflow) c^{MG} of $\mathcal{N}^{MG}$ such that for all $t_1, t_2 \in c^{MG}$ instances of the same $t \in T$, then $c^{MG} \cap t_1^\bullet$ and $c^{MG} \cap t_2^\bullet$ are instances of the same $p \in P$, and*
2. *there exists an integer $k \geq 1$ such that $\mathcal{M}(x) = k.i_x$ where i_x is the number of instances of x in c^{MG}, for all $x \in P \cup T$.*

Proof sketch. Let c^{MG} be a circuit of $\mathcal{N}^{MG}$. If c^{MG} contains several instances of a place of $\mathcal{N}$, find a subpath of c^{MG} that begins at an instance p_1 of a given place $p \in P$ and ends at another instance p_2 of the same place and such that it does not include more than one instance of any other place (its existency is obvious). This path corresponds with a circuit of the net $\mathcal{N}$. Substitute the subpath of c^{MG} by the single place p_1. Repeat the procedure of finding subpaths which correspond with circuits of the original net until c^{MG} has been reduced to a circuit without more than one instance of any place. It corresponds with a circuit of $\mathcal{N}$. Therefore, to each circuit of $\mathcal{N}^{MG}$ corresponds a multiset of circuits of $\mathcal{N}$. Moreover, by the method of derivation of $\mathcal{N}^{MG}$ (taking into account the deterministic routing at conflicts) the multiset of circuits of $\mathcal{N}$ corresponding with a circuit of $\mathcal{N}^{MG}$ is an $\mathcal{R}$–mc. If the number of copies of all the circuits in the multiset is a multiple of a given integer k, we consider the multiset obtained after dividing all the number of copies by k, and this multiset (which is also an $\mathcal{R}$–mc) verifies condition (b) of definition of minimality. Now, if condition (1) of the lemma is assumed for c^{MG}, condition (a) of minimality of the derived multiset follows.

Conversely, let $\mathcal{M}$ be a minimal $\mathcal{R}$–mc of $\mathcal{N}$. Then there exists a *non-minimal* circuit c of $\mathcal{N}$ with the same support than $\mathcal{M}$ and such that $c(p) = \mathcal{M}(p)$ for all $p \in ||\mathcal{M}|| \cap P$. Moreover, by minimality of $\mathcal{M}$, the circuit c verifies a condition analogue to (1), that is: if $t \in c$ then there exists only one $p \in t^\bullet$ such that $p \in c$. By the method of derivation of $\mathcal{N}^{MG}$ from $\mathcal{N}$ and since the original multiset was an $\mathcal{R}$–mc, there exists a path c^{MG} of $\mathcal{N}^{MG}$ equal (except instances) to the circuit c. Finally, either the obtained path of $\mathcal{N}^{MG}$ is a circuit that verifies (1) and $\mathcal{M}(x) = i_x$ (where i_x is the number of instances of x in c^{MG}), for all $x \in P \cup T$, or there exists a circuit of $\mathcal{N}^{MG}$ that consists of k repeated instances of the path c^{MG}, and that circuit verifies also (1) and (2). ■

Now, we define an expansion of a given live and 1–bounded stochastic free choice net with deterministic resolution of conflicts which allows a polynomial computation of its minimal $\mathcal{R}$–mcs.

Definition 4.1.6 (Expansion of a stochastic net) *Let* $\mathcal{N} =$

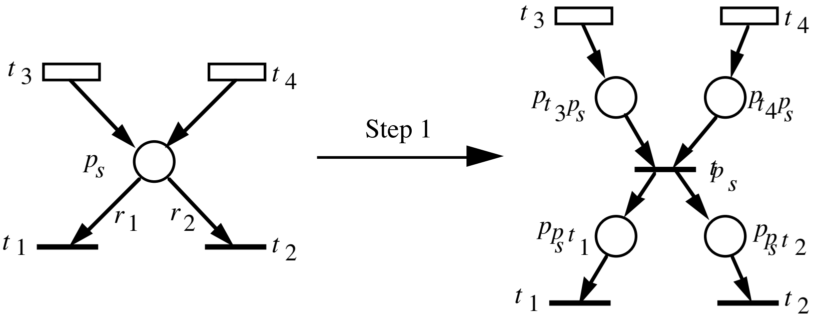


Figure 4.11: The replacement of a shared place: Step 1.

$\langle P, T, Pre, Post\rangle$ be a free choice net. The expanded net of $\mathcal{N}$, denoted as $\widehat{\mathcal{N}}$, is obtained from $\mathcal{N}$ after the following steps:

Step 1. *(Lautenbach expansion) Let $\overline{\mathcal{N}}$ be initially equal to $\mathcal{N}$. Replace each shared place $p_s \in P$ (i.e., such that $|{}^\bullet p_s| > 1 \vee |p_s^\bullet| > 1$) as follows (see figure 4.11):*

$$\begin{array}{l}
\overline{P} := (\overline{P} \setminus \{p_s\}) \cup \bigcup_{t \in {}^\bullet p_s} p_{tp_s} \cup \bigcup_{t \in p_s^\bullet} p_{p_st} \\
\overline{T} := \overline{T} \cup t_{p_s} \\
\overline{Pre}(p_{tp_s}, t_{p_s}) = 1, \ \forall t \in {}^\bullet p_s \\
\overline{Post}(p_{p_st}, t_{p_s}) = 1, \ \forall t \in p_s^\bullet \\
\overline{Post}(p_{tp_s}, t) = Post(p_s, t), \ \forall t \in {}^\bullet p_s \\
\overline{Pre}(p_{p_st}, t) = Post(p_s, t), \ \forall t \in p_s^\bullet
\end{array}$$

Step 2. *Derive a new net $\widehat{\mathcal{N}}$ from $\overline{\mathcal{N}}$ as follows: for each output-shared place $p_s \in P$ (i.e., such that $|p_s^\bullet| > 1$) and for each pair of output transitions t_1, t_2 of p_s add the transition $t_{p_st_1t_2}$ as follows (see figure 4.12):*

$$\begin{array}{l}
\widehat{T} := \overline{T} \cup \{t_{p_st_1t_2}\} \\
\widehat{Pre}(p_{p_st_1}, t_{p_st_1t_2}) = r_2 \\
\widehat{Post}(p_{p_st_2}, t_{p_st_1t_2}) = r_1
\end{array}$$

where r_1, r_2 are positive integer numbers proportional to the routing rates associated with t_1, t_2 in the conflict at place p_s.

Step 3. *Associate to each $t \in \widehat{T}$ the parameter $\widehat{s}(t)$ such that: $\widehat{s}(t_i) = s_i$ if $t_i \in \widehat{T} \cap T$ (where s_i is the average service time of transition t_i in the original net), and $\widehat{s}(t) = 0$ if $t \in \widehat{T} \setminus T$.*

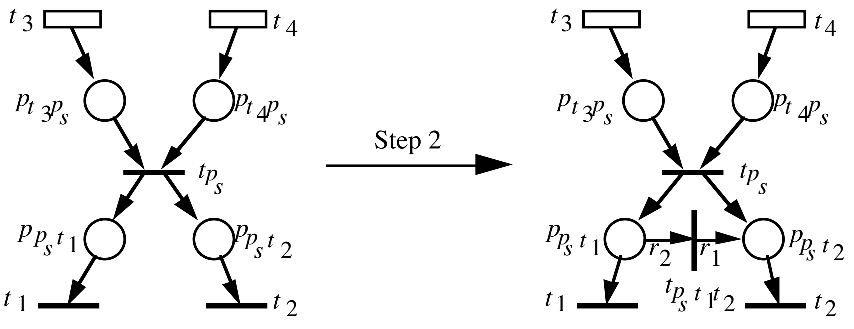


Figure 4.12: Additional constraint for an output-shared place: Step 2.

In chapter 2 (theorem 2.1.6.1) a graph theoretical concept (circuit) was related with another one of algebraic nature (minimal P-semiflow) for marked graphs. Now, multisets of circuits are related with (non necessarily minimal) P-semiflows.

Lemma 4.1.3 [Lau87] *Let $\mathcal{N} = \langle P, T, Pre, Post \rangle$ be a marked graph. Then Y is a P-semiflow of $\mathcal{N}$ iff there exists a multiset $\mathcal{M}$ of circuits of $\mathcal{N}$ such that $Y(p) = \mathcal{M}(p)$ for all $p \in P$.*

Now, the following result can be derived from lemma 4.1.3.

Theorem 4.1.6 *Let $\mathcal{M}$ be a multiset of circuits of a free choice net $\mathcal{N}$. $\mathcal{M}$ is a minimal $\mathcal{R}$–mc of $\mathcal{N}$ iff there exists a minimal P-semiflow $\widehat{Y}$ of the expanded net $\widehat{\mathcal{N}}$ such that:*

1. *For all $p \in P$ that is not a shared place then $\mathcal{M}(p) = \widehat{Y}(p)$.*
2. *For all $p \in P$ that is a shared place then $\mathcal{M}(p, t_i) = \widehat{Y}(p_{pt_i})$ and $\mathcal{M}(t_j, p) = \widehat{Y}(p_{t_j p})$ for all $t_i \in p^\bullet$ and for all $t_j \in {}^\bullet p$.*

The paragraphs below contain all technical details in order to prove this theorem. Previously, we introduce a simple definition and a technical lemma.

Definition 4.1.7 *Let $\mathcal{N} = \langle P, T, Pre, Post \rangle$ be a free choice net, $\widehat{\mathcal{N}}$ its corresponding expanded net, and $\widehat{Y}$ a P-semiflow of $\widehat{\mathcal{N}}$. The restricted support of $\widehat{Y}$ respect to places P of the original net $\mathcal{N}$, $||\widehat{Y}||_{\mathcal{N}}$, is given by:*

(a) If p is non-shared, then $p \in ||\widehat{Y}||_{\mathcal{N}} \Leftrightarrow p \in ||\widehat{Y}||$.

(b) If p is shared, then $p \in ||\widehat{Y}||_{\mathcal{N}}$ if and only if all places p_{pt} resulting from the expansion of the output arcs of p belong to $||\widehat{Y}||$.

Now, as in [Lau87], the following result can be derived from lemma 4.1.3.

Lemma 4.1.4 *Let $\mathcal{M}$ be a multiset of circuits of a free choice net. $\mathcal{M}$ is an $\mathcal{R}$–mc of $\mathcal{N}$ iff there exists a P-semiflow $\widehat{Y}$ of $\widehat{\mathcal{N}}$ such that:*

1. *For all $p \in P$ that is non-shared $\mathcal{M}(p) = \widehat{Y}(p)$.*
2. *For all $p \in P$ that is a shared place $\mathcal{M}(p, t_i) = \widehat{Y}(p_{pt_i})$ and $\mathcal{M}(t_j, p) = \widehat{Y}(p_{t_jp})$ for all $t_i \in p^\bullet$ and for all $t_j \in {}^\bullet p$.*

Proof. Let $\widehat{Y}$ be a P-semiflow of $\widehat{\mathcal{N}}$. $\widehat{Y}$ is also a P-semiflow of $\overline{\mathcal{N}}$ because $\overline{\mathcal{N}}$ and $\widehat{\mathcal{N}}$ differ only in the transitions added in Step 2 of definition 4.1.6. By construction, the net $\overline{\mathcal{N}}$ is a marked graph and therefore (by lemma 4.1.3) there exists a multiset $\overline{\mathcal{M}}$ of circuits of $\overline{\mathcal{N}}$ such that for all $p \in \overline{P}$: $\widehat{Y}(p) = \overline{\mathcal{M}}(p)$.

Let us suppose that $\overline{\mathcal{M}}(t_p) > 0$, where t_p is generated in the expansion of an output shared place p and $t_p^\bullet = \{p_{pt_i} \mid t_i \in p^\bullet, 1 \leq i \leq v\}$ In $\widehat{\mathcal{N}}$, between two places $p_{pt_j}, p_{pt_{j+1}}$ $(1 \leq j \leq v-1)$ there exists a transition $t_{pt_jt_{j+1}}$ that verifies $Pre(p_{pt_j}, t_{pt_jt_{j+1}}) = r_j$ and $Post(p_{pt_{j+1}}, t_{pt_jt_{j+1}}) = r_{j+1}$ (see Step 2 in definition 4.1.6). Therefore, $\widehat{Y}$ verifies that $r_j\widehat{Y}(p_{pt_{j+1}}) = r_{j+1}\widehat{Y}(p_{pt_j})$. This implies that $r_j\overline{\mathcal{M}}(p_{pt_{j+1}}) = r_{j+1}\overline{\mathcal{M}}(p_{pt_j})$.

Finally, it is easy to see that, to $\overline{\mathcal{M}}$ corresponds a multiset of circuits $\mathcal{M}$ of $\mathcal{N}$ with $||\mathcal{M}|| \cap P = ||\widehat{Y}||_{\mathcal{N}}$, such that $\mathcal{M}(p, t_i) = \overline{\mathcal{M}}(p_{pt_i})$, $1 \leq i \leq v$ (being p a shared place), $\mathcal{M}(t_j, p) = \overline{\mathcal{M}}(p_{t_jp})$ and for all non-shared place $\mathcal{M}(p) = \overline{\mathcal{M}}(p)$. Therefore, $\mathcal{M}$ is an $\mathcal{R}$–mc.

The converse implication can be obtained by reversing the previous arguments. ■

From the above lemma, we can deduce the following obvious result:

Corollary 4.1.4 *Let $\mathcal{M}$ be an $\mathcal{R}$–mc of a free choice net $\mathcal{N}$ and $\widehat{Y}$ the corresponding P-semiflow in the expanded net $\widehat{\mathcal{N}}$. Then $||\mathcal{M}|| \cap P = ||\widehat{Y}||_{\mathcal{N}}$.*

Now, we prove the theorem 4.1.6.

Proof of theorem 4.1.6. ($\Rightarrow$) By lemma 4.1.4, there exists a P-semiflow $\widehat{Y}_1$ of $\widehat{\mathcal{N}}$ such that $||\mathcal{M}|| = ||\widehat{Y}_1||_{\mathcal{N}}$ and the conditions (1) and (2) of the theorem hold. If $\widehat{Y}_1$ is minimal, we are done by taking $\widehat{Y} = \widehat{Y}_1$. Assume $\widehat{Y}_1$ is not minimal. Then, by definition of minimality, there exists a minimal P-semiflow $\widehat{Y}_2$ such that $||\widehat{Y}_2|| \subset ||\widehat{Y}_1||$. Moreover, by lemma 4.1.4 there exists an $\mathcal{R}$–mc $\mathcal{M}'$ that satisfies the conditions (1) and (2) of the theorem and $||\mathcal{M}'|| \cap P = ||\widehat{Y}_2||_{\mathcal{N}}$. Since $||\widehat{Y}_2|| \subset ||\widehat{Y}_1||$ implies that $||\widehat{Y}_2||_{\mathcal{N}} \subseteq ||\widehat{Y}_1||_{\mathcal{N}}$, it follows $||\mathcal{M}'|| \subseteq ||\mathcal{M}||$. Because the minimality of $\mathcal{M}$, the equality holds. Then we take $\widehat{Y} = \widehat{Y}_2$.

($\Leftarrow$) By lemma 4.1.4, there exists an $\mathcal{R}$–mc $\mathcal{M}_1$ of $\mathcal{N}$ such that $||Y||_{\mathcal{N}} = ||\mathcal{M}_1|| \cap P$ and the conditions (1) and (2) hold. If $\mathcal{M}_1$ is minimal, we are done by taking $\mathcal{M} = \mathcal{M}_1$. Assume that $\mathcal{M}_1$ is not minimal. Then, by definition of minimality, there exists a minimal $\mathcal{R}$–mc $\mathcal{M}_2$ such that the $\mathcal{R}$–mc $\mathcal{M}_2$ is enclosed into the $\mathcal{R}$–mc $\mathcal{M}_1$ (this inclusion is stronger than the one of the support of places). Moreover, by the previous implication (demostrated in this theorem) there exists a minimal P-semiflow $\widehat{Y}'$ that satisfies conditions (1) and (2) of the theorem. Since $\mathcal{M}_2$ is enclosed into $\mathcal{M}_1$ and this means inclusion at level of places, transitions, and arcs, $||\widehat{Y}'|| \subseteq ||\widehat{Y}||$ is verified. Because the minimality of $\widehat{Y}$, the equality holds. Then we take $\widehat{Y} = \widehat{Y}'$. ■

The next theorem gives a lower bound for the mean cycle time of a transition of a live and 1–bounded free choice net, using the P-semiflows of the expanded net defined above.

Theorem 4.1.7 *A lower bound for the mean cycle time of transition t_j of a live and 1–bounded free choice net $\langle \mathcal{N}, M_0 \rangle$ is:*

$$\Gamma_{(j)}^{min} = \frac{k_{\widehat{Y}^*} \, \gamma(\widehat{Y}^*)}{k_{v(j)}}$$

where:

- $\gamma(\widehat{Y}^*)$ *can be obtained by solving the following linear programming* *problem:*

$$\begin{array}{rll} \gamma(\widehat{Y}^*) = & \textit{maximize} & \widehat{Y}^T \cdot \widehat{PRE} \cdot \vec{\widehat{s}} \\ & \textit{subject to} & \widehat{Y}^T \cdot \widehat{C} = 0 \\ & & 1\!\!1^T \cdot \widehat{Y} = 1 \\ & & \widehat{Y} \geq 0 \end{array} \tag{LPP14}$$

where $(\widehat{PRE})$ $\widehat{C}$ *is the (pre-) incidence matrix of the expanded net* $\widehat{\mathcal{N}}$ *of* $\mathcal{N}$ *and* $\vec{\widehat{s}}$ *is the vector defined in Step 3 of expansion of* $\mathcal{N}$*.*

- $k_{\widehat{Y}^*}$ *is a non-negative number such that* $k_{\widehat{Y}^*}\widehat{Y}^* = \widehat{Y}^*_Z$ *where the vector* $\widehat{Y}^*_Z \in \mathbb{Z}^{|\widehat{P}|}$ *and the greatest common divisor of its components is equal to 1.*
- $k_{v^{(j)}}$ *is a non-negative number such that* $k_{v^{(j)}}\vec{v}^{(j)} = \vec{v}_Z$ *where the vector* $\vec{v}_Z \in \mathbb{Z}^{|T|}$ *and the greatest common divisor of its components is equal to 1.*

Proof. The optimum solution of (LPP14) is always reached for a minimal P-semiflow because, taking into account [Mur83, theorem 3.3], if (LPP14) has an optimum feasible solution, then it has a basic feasible solution $\widehat{Y}$ that is optimum. Therefore, the set of rows that are used by $\widehat{Y}$ is linearly independent (i.e., full rank). Considering that $\widehat{Y}^T \cdot \widehat{C} = 0$, we obtain that the number of non-null entries of vector $\widehat{Y}$ (i.e., the number of rows used by $\widehat{Y}$) is equal to the rank of rows of $\widehat{C}$ used by $\widehat{Y}$ plus one. This last statement is precisely the characterization of a minimal P-semiflow, presented in [CS89b].

Multiplying the optimum value of the above linear programming problem by the constant $k_{\widehat{Y}^*}$, we obtain the sum of average service times of transitions covered by a minimal $\mathcal{R}$–mc. This is because the vector $\widehat{Y}^*_Z = k_{\widehat{Y}^*}\widehat{Y}^*$ is a minimal P-semiflow whose components are integer and minimal, thus there exists a minimal $\mathcal{R}$–mc, according to theorem 4.1.6.

Then the above computation can be rewritten in terms of minimal $\mathcal{R}$–mc in the following way:

$$\begin{array}{rll} k_{\widehat{Y}^*}\,\gamma(\widehat{Y}^*) = & \text{maximize} & \mathcal{M}|_P \cdot PRE \cdot \vec{s} \\ & \text{such that} & \mathcal{M} \text{ is a minimal } \mathcal{R}\text{–mc of } \mathcal{N} \end{array} \tag{4.35}$$

where $\mathcal{M}|_P$ denotes the row vector with components $\mathcal{M}|_P(p) = \mathcal{M}(p)$ for all place p of the original net.

Now, we consider the behaviourally equivalent (for a given deterministic conflicts resolution policy) marked graph derived in the first paragraphs of this section. According to lemma 4.1.2, to each k multiple of a minimal $\mathcal{R}$–mc of the original net corresponds a circuit (i.e., minimal P-semiflow) of this marked graph, and each of them contains only one token. Then $k\, k_{\widehat{Y}^*}\, \gamma(\widehat{Y}^*)$ is the value computed in the Step 3 of the algorithm for the derivation of the behaviourally equivalent marked graph. Finally, since the number of instances of t_j in the behaviourally equivalent marked graph is $k\, \vec{v}_Z(t_j)$ (lemma 4.1.2) then, dividing $k\, k_{\widehat{Y}^*}\, \gamma(\widehat{Y}^*)$ by this constant, we obtain the mean cycle time of transition t_j computed in Step 4 of the algorithm for the derivation of the behaviourally equivalent marked graph, hence being a lower bound for the mean cycle time of t_j in the original net. ■

In fact, from the reachability of the bound for strongly connected marked graphs (see theorem 3.1.3) the reachability of the bound given by theorem 4.1.7 follows for live and 1–bounded free choice nets.

Theorem 4.1.8 *For live and 1–bounded free choice nets with arbitrary values of average service times of transitions and arbitrary routing rates defining the resolution of conflicts, the lower bound for the mean cycle time obtained from theorem 4.1.7 is reachable.*

Proof. The result follows from the following considerations: (1) deterministic service times and deterministic routing are particular cases of timing and conflict resolution policy, respectively; (2) for such policy, theorem 4.1.7 can be applied; and (3) for the case of marked graphs with deterministic timing, the derived bound is reached. ■

As in the case of strongly connected marked graphs (cfr. theorem 3.1.4), a characterization of liveness and 1–boundedness for structurally live and structurally bounded free choice nets can be derived:

Theorem 4.1.9 *Liveness and 1–boundedness of structurally live and structurally bounded free choice nets can be characterized in polynomial time.*

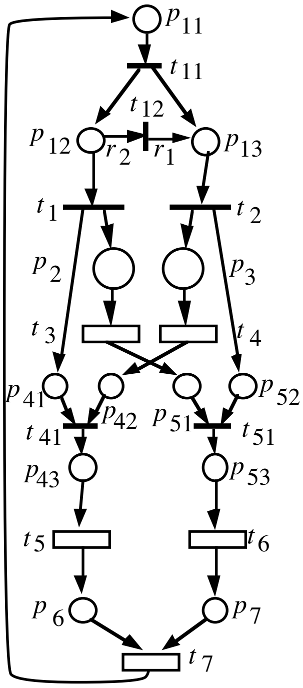


Figure 4.13: Expanded net of the one depicted in figure 4.7.a (the weights r_1 and r_2 are such that $r_1/r_2 = q/(1-q)$).

Proof. 1–boundedness can be characterized computing the structural marking bound of places which is equal to the actual marking bound for structurally live and structurally bounded free choice nets (cfr. theorem 2.1.12). Liveness can be characterized checking the boundedness of the problem (LPP14): the value given by theorem 4.1.7 is a lower bound for the mean cycle time; if this value is infinite the mean cycle time is unbounded, and the net is non-live; if the value given by theorem 4.1.7 is finite, since it is reachable (cfr. theorem 4.1.8), the net must be *deadlock-free*. We know that for structurally live and structurally bounded free choice nets, liveness and deadlock-freeness are equivalent (cfr. property2.1.3). Thus the finiteness of the value given by theorem 4.1.7 is sufficient to establish the liveness of a 1–bounded structurally live and structurally bounded free choice net. ■

As an example, let us consider once more the live and 1–bounded free choice net depicted in figure 4.7.a. Its expanded net according to definition 4.1.6 is depicted in figure 4.13. The application of theorem 4.1.7 for this net gives the value:

$$\begin{aligned}\Gamma_{(7)} &= q\max\{s_5, s_3+s_6\} + (1-q)\max\{s_4+s_5, s_6\} + s_7 = \\ &= \max\{\; qs_3+s_6, \\ &\qquad\quad (1-q)s_4+s_5, \\ &\qquad\quad qs_3+(1-q)s_4+(1-q)s_5+qs_6, \\ &\qquad\quad qs_5+(1-q)s_6\} + \\ &\quad\; +s_7\end{aligned} \tag{4.36}$$

which is exactly the actual cycle time of the net for deterministic service time of transitions.

The natural extension of the results presented in this section would consist on the computation of lower bounds for the mean cycle time of transitions of live and k–bounded ($k > 1$) free choice nets. Now, we argue that such extension cannot be obtained applying the techniques used for 1–bounded nets.

Let us consider the Petri net depicted in figure 4.2, but now with initial marking of place p_1 equal to 2 tokens. Suppose the following deterministic conflict resolution policy at place p_1: "select twice transition t_1, then once transition t_2, and repeat it" (i.e., the routing rate for transition t_1 is twice the routing rate of t_2). A direct extension of the method presented above for the computation of a lower bound for the mean cycle time of t_5 would be the following:

1. Derive the expanded net.
2. Apply the theorem 4.1.7 with $j = 5$ and divide by 2 the obtained value (i.e., divide the cycle time of the slowest circuit by the number of contained tokens).

But the obtained value is not a lower bound for the mean cycle time of t_5 in the original net, in general. For instance, if transitions t_3 and t_4 are supposed to be exponentially timed with averages $s_3 = s_4 = 1$, the value that can be derived from theorem 4.1.7, dividing by 2, is $(2/3 + 1/3)/2 = 0.5$, while the actual cycle time is $\Gamma_{(5)} = 0.387$.

The reason of this bad result is the following: the value obtained from theorem 4.1.7, dividing by 2, is equal to the one obtained developing the behaviourally equivalent marked graph, putting 2 tokens in place p_1, computing the mean cycle time applying (LPP3), and dividing by 3 (because t_5 is instantiated three times in the marked graph). But this marked graph (with 2 tokens at place p_1) is not behaviourally equivalent to the original free choice net. Actually, the marked graph is slower than the original net. This is because in the marked graph the *T-components* of the net are completely sequentialized one after the other, while in the original net they share the places p_4 and p_5, and this fact makes faster the synchronization at t_5.

The above result is not always true. That is, the value obtained from theorem 4.1.7 dividing by the number of tokens present in the optimum P-semiflow is not greater than the exact mean cycle time, in all cases. For instance, if the net depicted in figure 4.8 is considered with deterministic resolution of conflicts (2 times t_1, then once t_2) and exponentially distributed service times of transitions t_3, t_4, t_5, t_6, and t_7 (with means equal to 1), the value that can be derived from theorem 4.1.7 dividing by 2 is $(2/3 + 1/3 + 2/3 + 2/3 + 2/3)/2 = 1.5$, while the actual cycle time is $\Gamma_{(7)} = 1.52$

4.2 Lower bounds for the steady-state throughput

In this section, lower bounds on throughput are proposed, independent of the higher moments of the service time probability distribution functions, based on the computation of the transition *liveness bounds*, defined in section 1.2.3.

The trivial lower bound on throughput consisting of the inverse of the sum of the service times of all transitions, has been improved for strongly connected marked graphs in section 3.2.2, based on the knowledge of the liveness bound $L(t)$ for all transitions t of the marked graph.

Moreover, this lower bound for the throughput has been shown to be reachable for any marked graph topology and for some assignement of probability distribution functions to the service time of transitions (cfr. theorem 3.2.3).

This lower bound for the throughput of transitions can be applied for live and bounded free choice nets in the following way: weighting the average service time s_i of t_i with the corresponding visit ratio $v_i^{(j)}$ (for a fixed t_j); in other words, considering the average service demand $D_i^{(j)}$ for transition t_i (as defined in section 4.1.1).

Theorem 4.2.1 *For any live and bounded free choice net with a specification of the average service time s_i for each transition t_i it is not possible to assign probability distribution functions to the transition service times such that the mean cycle time of transition t_j is greater than*

$$\Gamma_{(j)}^{max} = \sum_{i=1}^{m} \frac{D_i^{(j)}}{L(t_i)} = \sum_{i=1}^{m} \frac{v_i^{(j)} s_i}{L(t_i)} \tag{4.37}$$

independently of the topology of the net.

Moreover, this upper bound for the mean cycle time is reachable for any live and bounded free choice net, for some conflicts resolution policy, and for some assignement of probability distribution functions to the service time of transitions (i.e., the bound cannot be improved).

Proof. Derive a marked graph (for deterministic conflicts resolution policy) with the same vector of visit ratios (as in section 4.1.3) and apply the bound obtained in theorem 3.2.2. Different instances of a given transition are considered in the relative throughput of the corresponding component in the vector of visit ratios. Thus, the bound obtained for the derived marked graph applying theorem 3.2.2 coincides with the bound obtained for the original net using the formula stated in this theorem. The bound is reachable because, for a deterministic conflict resolution, the throughput of the derived marked graph and of the original net are equal and in this case the lower bound for the throughput of marked graphs is reachable (see theorem 3.2.3). ■

We recall (cfr. theorem 2.1.13) that in the case of live and bounded free choice nets, the liveness bound equals the enabling and the structural enabling bounds for each transition. Therefore, the reachable lower bound for the throughput of these nets, presented in the above theorem, can be computed as follows:

Theorem 4.2.2 *For any live and bounded free choice net with a specification of the average service time s_i for each transition t_i, the reachable upper bound for the mean cycle time of transition t_j given by theorem 4.2.1 can be computed as:*

$$\Gamma_{(j)}^{max} = \sum_{i=1}^{m} \frac{D_i^{(j)}}{SE(t_i)} = \sum_{i=1}^{m} \frac{v_i^{(j)} s_i}{SE(t_i)} \tag{4.38}$$

where $SE(t_i)$ is the structural enabling bound of t_i.

From the above theorem, the next result follows:

Corollary 4.2.1 *The computation of the upper bound for the mean cycle time of a transition of a live and bounded free choice net presented in theorem 4.2.1 has polynomial complexity on the net size.*

Really, the computation of $\Gamma_{(j)}^{max}$ can be achieved by solving the linear programming problems (LPP1) that define the structural enabling bound of transitions (definition 1.2.3).

4.3 Conclusions

Upper and lower bounds for the steady-state throughput of transitions of live and bounded free choice nets have been derived in this chapter.

Concerning the throughput upper bound, a direct application of the same result that gave a reachable bound for live and bounded marked graphs (using P-semiflows) does not lead to a tight bound. In order to improve the bound, other marking invariants have been considered, as some ones derived from traps. Unfortunately, the bounds obtained from these invariants are not always tighter than the ones based on P-semiflows. A real improvement has been obtained after the addition of implicit places to the net. The bound computed using P-semiflows of the augmented net (with implicit places) does really improve the previous bound, but it is not reachable either. A reachable throughput upper bound has been derived for the case of 1–bounded nets using another "natural" generalization of the marked graphs result: multisets of circuits of free choice nets instead of circuits of marked graphs. A polynomial complexity algorithm for the computation of this bound has been obtained after the introduction of a particular net transformation, which is also of linear complexity.

In the case of the upper bound for the mean cycle time (throughput lower bound), the sum of the average service demands for transitions divided by their corresponding liveness bounds has been shown to be a reachable value. For a deterministic resolution policy and for some distribution functions with arbitrary mean values and increasing the coefficients of variation, the bound tends to the exact value. The computation of this bound has also polynomial complexity, since for live and bounded free choice nets the computation of the average service demands and the liveness bounds of transitions can be reduced to the solution of a linear system of equations and of some (no more than the number of transitions) linear programming problems, respectively.

Chapter 5

Extensions to other net subclasses

In chapter 2, a classification of stochastic Petri nets attending to the computability of visit ratios from different net parameters was presented. After that, the main results on the computation of bounds for marked graphs and live and bounded free choice nets have been derived in chapters 3 and 4, respectively.

Now, we extend some of these results for other net subclasses presented in chapter 2. The first one (section 5.1) is that of live and structurally bounded mono-T-semiflow nets. After that, in section 5.2, we present bounds for general live and structurally bounded FRT-nets. In section 5.3, totally open deterministic systems of sequential processes, that are defined not only from the structure but also from the initial marking are considered. Their exact analysis is possible in polynomial time on the net size. Bounds for live and bounded persistent nets are considered in section 5.4. These nets are behaviourally characterized and require the expansion of the reachability graph for the computation of the vector of visit ratios, hence for the computation of performance bounds.

5.1 Mono-T-semiflow nets

Let us now consider live and structurally bounded mono-T-semiflow nets and give upper and lower bounds for their steady-state throughput.

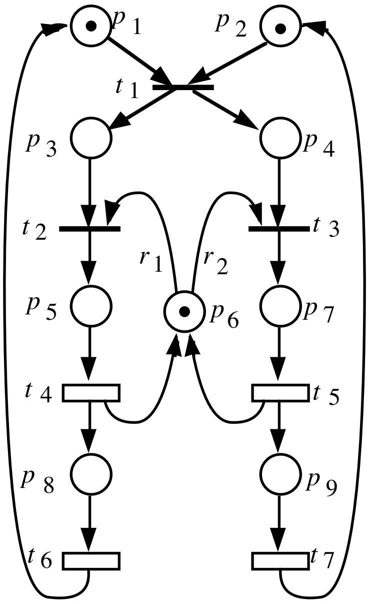


Figure 5.1: Live and 1–bounded mono-T-semiflow net whose cycle time depends on the conflict resolution policy.

Since mono-T-semiflow nets are structurally characterized, they can be recognized without a previous behavioural analysis. According to the results in section 2.1.2, they can be detected by computing their unique minimal T-semiflow X, in polynomial time. Since mono-T-semiflow nets are FRT-nets, weak ergodicity of its firing process is assured (see theorem 2.1.3) but, unfortunately, the ergodicity of the marking process is not guaranteed (property 2.1.8). However, even in the case in which the marking process is non-ergodic, the computation of the throughput bounds makes sense. The values that we compute in this section are bounds for all possible steady-state behaviours of the marking process of the net.

Let us remark also that for live structurally bounded mono-T-semiflow nets, the vector of visit ratios is fixed from the structure, like in the particular case of marked graphs (unity vector in that case). But the exact cycle time of a mono-T-semiflow net depends on the probabilities defining the resolution of conflicts and, of course, on the service times.

As an example, let us consider the live and 1–bounded mono-T-semiflow net depicted in figure 5.1. Let t_1, t_2, and t_3 be immediate transitions. Let $s_4 = 1$, $s_5 = 2$, $s_6 = 2$, and $s_7 = 3$ be the deterministic service times of transitions t_4, t_5, t_6, and t_7, respectively. Let q and $1 - q$ be the probabilities of firing transitions t_2 and t_3, respectively, *when they are simultaneously enabled* ($0 \leq q \leq 1$). If the conflict between t_2 and t_3 is almost surely solved in favour of t_2 (i.e., if $q = 1$), the mean cycle time of the net is:

$$\Gamma_{(q=1)} = 1 + \max\{2, 2 + 3\} = 6 \tag{5.1}$$

On the other hand, if t_3 always fires before t_2 (i.e., $q = 0$), the mean cycle time is:

$$\Gamma_{(q=0)} = 2 + \max\{1 + 2, 3\} = 5 \tag{5.2}$$

Therefore, the mean cycle time depends on the conflict resolution policy.

5.1.1 Lower bound for the mean cycle time

Let $\langle \mathcal{N}, M_0 \rangle$ be a live and structurally bounded mono-T-semiflow net and X its unique minimal T-semiflow. Then, the visit ratio for transition t_i ($i = 1, \ldots, m$) normalized, for instance, for transition t_j is (by theorem 2.1.5):

$$v_i^{(j)} = \frac{X(i)}{X(j)} \tag{5.3}$$

If s_i denotes the average service time of transition t_i, the average service demand of transition t_i ($i = 1, \ldots, m$) is:

$$D_i^{(j)} \overset{\text{def}}{=} v_i^{(j)} s_i \tag{5.4}$$

Then, from Little's law (equation (3.4) in section 3.1.1) and considering the P-semiflows of $\mathcal{N}$, the following lower bound for the mean cycle time of transition t_1 can be derived, as in section 4.1.1.2:

$$\Gamma_{(j)} \geq \max_{Y \in \{P-semiflow\}} \frac{Y^T \cdot PRE \cdot \vec{D}^{(j)}}{Y^T \cdot M_0} \tag{5.5}$$

where $\vec{D}^{(j)}$ denotes the vector with components $D_i^{(j)}$, $i = 1, \ldots, m$.

Theorem 5.1.1 *For any live and structurally bounded mono-T-semiflow net, a lower bound for the mean cycle time of transition t_j can be computed by the following linear programming problem:*

$$\begin{array}{rll} \Gamma_{(j)} \geq & \textit{maximize} & Y^T \cdot PRE \cdot \vec{D}^{(j)} \\ & \textit{subject to} & Y^T \cdot C = 0 \\ & & Y^T \cdot M_0 = 1 \\ & & Y \geq 0 \end{array} \qquad \text{(LPP15)}$$

As in the case of free choice nets, if the solution of (LPP15) is unbounded (i.e., if there exists an unmarked P-semiflow), since it is a lower bound for the mean cycle time of transition t_j, the non-liveness can be assured (infinite cycle time). In fact, if (LPP15) is unbounded, then the net reaches a deadlock because liveness and deadlock-freeness are equivalent properties for mono-T-semiflow nets (because they are FRT-nets, see property 2.1.3).

Nevertheless, the bound given by theorem 5.1.1 is not reachable, in general. Moreover, a mono-T-semiflow net can be non-live and the obtained lower bound for the mean cycle time be finite. In other words:

Property 5.1.1 *For mono-T-semiflow nets, liveness is not characterized by the finiteness of the lower bound of the mean cycle time computed by means of (LPP15).*

This can be easily checked by considering the net depicted in figure 5.2. It is non-live, so that the actual steady-state cycle time is infinite, even if the obtained bound is finite.

At this point, the techniques developed in section 4.1 for the improvement of the throughput upper bound of live and bounded free choice nets can be applied for live and structurally bounded mono-T-semiflow nets. Let us illustrate that possible improvement by considering the live and structurally bounded mono-T-semiflow depicted in figure 5.3.a. The unique minimal T-semiflow of this net is:

$$X = (2, 2, 1, 1, 1, 1)^T = \vec{v}^{(3)} \qquad (5.6)$$

The elementary P-semiflows are:

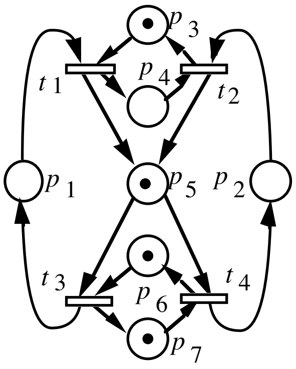


Figure 5.2: Non-live mono-T-semiflow net, even if all P-semiflows are marked and thus the bound given by (LPP15) is finite.

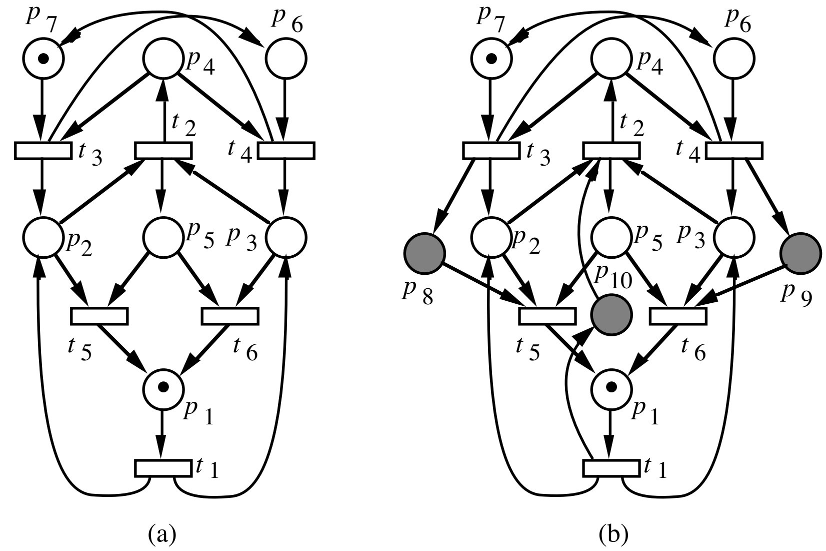


Figure 5.3: The bound given by theorem 5.1.1 is non-reachable for the original net while it gives the actual cycle time after the addition of implicit places.

$$\begin{aligned} Y_1 &= (2,1,1,1,1,0,0)^T \\ Y_2 &= (0,0,0,0,0,1,1)^T \end{aligned} \tag{5.7}$$

Then, the application of the problem (LPP15) stated in theorem 5.1.1 gives the bound:

$$\begin{aligned} \Gamma_{(3)} \geq \max \{ \ & 2s_1 + 2s_2 + \tfrac{1}{2}s_3 + \tfrac{1}{2}s_4 + s_5 + s_6, \\ & s_3 + s_4 \ \} \end{aligned} \tag{5.8}$$

Let us consider now the addition of the implicit places p_8, p_9, and p_{10} (associated with the minimal traps $\{p_1, p_2, p_5\}$, $\{p_1, p_2, p_3, p_4\}$, and $\{p_1, p_3, p_5\}$, respectively) computed in [CS89c] for the elimination of all *spurious solutions* from the linear state equation of the net (solutions that correspond to non-reachable markings), as it is depicted in figure 5.3.b. This net has six elementary P-semiflows:

$$\begin{aligned} Y_1 &= (2,1,1,1,1,0,0,0,0,0)^T \\ Y_2 &= (0,0,0,0,0,1,1,0,0,0)^T \\ Y_3 &= (1,0,1,1,0,0,0,1,0,0)^T \\ Y_4 &= (1,1,0,1,0,0,0,0,0,1)^T \\ Y_5 &= (1,0,0,0,1,0,0,0,1,0)^T \\ Y_6 &= (1,0,0,1,0,0,0,1,1,1)^T \end{aligned} \tag{5.9}$$

And if theorem 5.1.1 is applied to it, gives the bound:

$$\begin{aligned} \Gamma_{(3)} \ \geq \ & \max \{ \ 2s_1 + 2s_2 + \tfrac{1}{2}s_3 + \tfrac{1}{2}s_4 + s_5 + s_6, \\ & \qquad s_3 + s_4, \\ & \qquad 2s_1 + 2s_2 + s_3 + s_4 + s_5 + s_6, \\ & \qquad 2s_1 + 2s_2 + s_3 + s_4 + s_5 + s_6, \\ & \qquad 2s_1 + 2s_2 + s_5 + s_6, \\ & \qquad 2s_1 + 2s_2 + s_3 + s_4 + s_5 + s_6 \ \} \ = \\ = \ & 2s_1 + 2s_2 + s_3 + s_4 + s_5 + s_6 \end{aligned} \tag{5.10}$$

which is reached for all timing interpretations of the net, i.e., it is the exact mean cycle time for transition t_3, independently of the probability distribution functions associated with the service time of transitions.

5.1.2 Upper bound for the mean cycle time

Concerning the upper bound for the mean cycle time, only the "trivial" one, given by the sum of the average service times of all the transitions weighted by the vector of visit ratios, can be computed, so far.

If the net is live all transition must be firable, and the sum of all average service times multiplied by the number of occurrences of each transition in the (unique) average cycle of the model corresponds to any *complete sequentialization* of all the activities represented in the model. This *pesimistic* behaviour can be reached in some particular cases (e.g., for live and 1–bounded marked graphs, see section 3.2.1) if random variables with arbitrarily large coefficient of variation are conveniently selected (theorem 3.2.1).

Theorem 5.1.2 *For any live and structurally bounded mono-T-semiflow net with a specification of the average service time s_i for each transition t_i it is not possible to assign probability distribution functions to the transition service times such that the mean cycle time of transition t_j is greater than*

$$\sum_{i=1}^{m} D_i^{(j)} = \sum_{i=1}^{m} v_i^{(j)} s_i \tag{5.11}$$

independently of the topology of the net.

In order to improve the previous bound, an intuitive idea could be to take into account that some work can be done in parallel at each transition, since infinite-server semantics is assumed. From a queueing theory perspective and considering the steady-state behaviour, the number of servers at each *station* (transition) is equal to the corresponding enabling bound in steady-state (i.e., liveness bound), and the contribution of each transition to the duration of the complete sequentialization of all activities can be divided by its liveness bound. Thus, we could conjecture the following upper bound for the mean cycle time of t_j:

$$\Gamma_{(j)} \overset{?}{\leq} \sum_{i=1}^{m} \frac{D_i^{(j)}}{L(t_i)} = \sum_{i=1}^{m} \frac{v_i^{(j)} s_i}{L(t_i)} \tag{5.12}$$

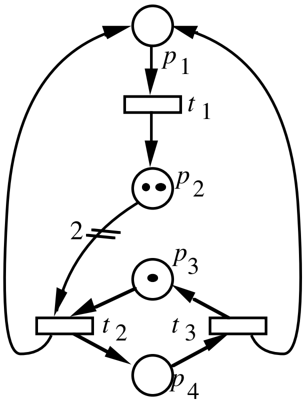


Figure 5.4: "Non-trivial" upper bound for the mean cycle time cannot be applied.

The same value would be obtained taking the algorithm used for the computation of the lower bound for the mean cycle time (theorem 5.1.1), substituting in it the "max" operator with the sum of the average service times of all transitions involved, and making some manipulation to avoid counting more than once the contribution of the same transition.

The conjecture (5.12) has been shown to be true for strongly connected marked graphs, in section 3.2.2. In fact, for this subclass of nets the upper bound for the mean cycle time given by (5.12) has been shown to be reachable for any net topology, for any specification of the average service times, and for some assignement of probability distribution functions to the service times of transitions, in section 3.2.3.

Concerning mono-T-semiflow nets, the conjecture (5.12) is false. This can be shown considering, for example, the mono-T-semiflow net depicted in figure 5.4 with average service times s_1, s_2, s_3 for transitions t_1, t_2, t_3, respectively. For this net, the vector of visit ratios normalized for transition t_2 is $\vec{v}^{(2)} = (2, 1, 1)^T$, and the liveness bounds of transitions are given by $L(t_1) = 2$, $L(t_2) = 1$, and $L(t_3) = 1$. Thus, the conjecture (5.12) would give the value $s_1 + s_2 + s_3$ as upper bound for $\Gamma_{(2)}$. If exponentially distributed random variables (with means s_1, s_2, s_3; $s_1 \neq s_3$) are associated with transitions, the steady-state cycle time for transition t_2 is

$$\Gamma_{(2)} = s_1 + s_2 + s_3 + \frac{s_1^2}{2(s_1 + s_3)} \tag{5.13}$$

which is greater than the value obtained applying (5.12), thus the conjecture is false.

Unfortunately, the trivial bound given by theorem 5.1.2 is non-reachable in general, and in some cases its value can be too pesimistic. An improving of this bound would probably require more information about the probability distribution functions of service times than their mean values, and this approach is not within the scope of this work.

5.2 FRT-nets

In this section, we derive performance bounds for the steady-state behaviour of live and structurally bounded FRT-nets (see definition 2.1.3). Lower bounds for the mean cycle time are derived in section 5.2.1, while in section 5.2.2 the computation of upper bounds for the mean cycle time is considered.

5.2.1 Lower bound for the mean cycle time

Let us consider a live and structurally bounded FRT-net; let s_i denote the average service time of transition t_i, $i = 1, \ldots, m$; let $\vec{v}^{(j)}$ be the vector of visit ratios for transitions normalized, for instance, for transition t_j (computed using equation (2.8) given by theorem 2.1.2); and let $\vec{D}^{(j)}$ be the vector of service demands for transitions (i.e., with components $D_i^{(j)} = v_i^{(j)} s_i$). Then, as in the previous section for mono-T-semiflow nets, the following can be stated:

Theorem 5.2.1 *For any live and structurally bounded FRT-net, a lower bound for the mean cycle time of transition t_j can be computed by the following linear programming problem:*

$$\begin{array}{rll} \Gamma_{(j)} \geq & \textit{maximize} & Y^T \cdot PRE \cdot \vec{D}^{(j)} \\ & \textit{subject to} & Y^T \cdot C = 0 \\ & & Y^T \cdot M_0 = 1 \\ & & Y \geq 0 \end{array} \tag{LPP16}$$

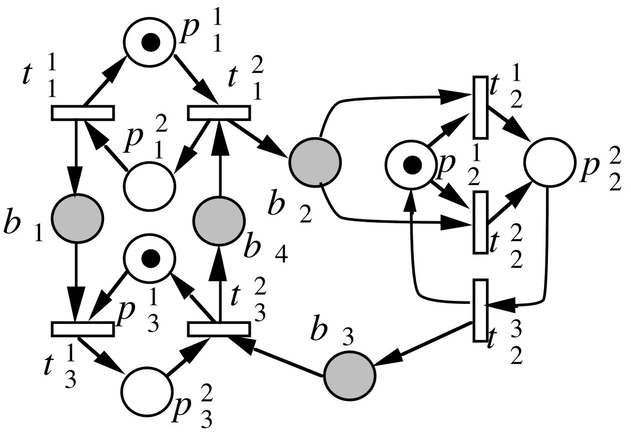


Figure 5.5: A live and structurally bounded FRT-net.

As an example, let us consider the live and structurally bounded FRT-net depicted in figure 5.5 (which can be seen as a system of three FRT-nets communicating through private buffers). Let us suppose that the conflict at place p_2^1 is solved equitably in favour of t_2^1 and t_2^2 (i.e., with probabilities 1/2 and 1/2). Then, the vector of visit ratios for transitions normalized for t_2^1 (computed using equation (2.8) given by theorem 2.1.2) is:

$$\vec{v}^{(t_2^1)} = (2, 2, 1, 1, 2, 2, 2)^T \tag{5.14}$$

The elementary P-semiflows of the net are:

	p_1^1	p_1^2	p_2^1	p_2^2	p_3^1	p_3^2	b_1	b_2	b_3	b_4
Y_1	1	1	0	0	0	0	0	0	0	0
Y_2	0	0	1	1	0	0	0	0	0	0
Y_3	0	0	0	0	1	1	0	0	0	0
Y_4	0	1	0	0	0	1	1	0	0	1
Y_5	0	0	0	1	0	0	0	1	1	1

Then a lower bound for the mean cycle time of transition t_2^1 given by theorem 5.2.1 is:

$$
\begin{aligned}
\Gamma_{(t_2^1)} \geq \; & \max \{ \; 2s_1^1 + 2s_1^2, \\
& \qquad s_2^1 + s_2^2 + 2s_2^3, \\
& \qquad 2s_3^1 + 2s_3^2, \\
& \qquad 2s_1^1 + 2s_1^2 + 2s_3^1 + 2s_3^2, \\
& \qquad 2s_1^2 + s_2^1 + s_2^2 + 2s_2^3 + 2s_3^2 \; \} = \\
= \; & 2s_3^2 + 2s_1^2 + \max \{2s_1^1 + 2s_3^1, s_2^1 + s_2^2 + 2s_2^3\}
\end{aligned}
\tag{5.15}
$$

where s_j^i denotes the average service time of transition t_j^i. For this example, the bound is tight. It is exactly the actual cycle time if deterministic service is assumed for transitions.

5.2.2 Upper bound for the mean cycle time

In what concerns the upper bound for the mean cycle time of transitions, the improvement of the "trivial" bound obtained for marked graphs and free choice nets dividing by the liveness bounds of transitions cannot be applied for live and structurally bounded FRT-nets (we have seen in section 5.1.2 that this improvement is not valid for mono-T-semiflows nets, which are FRT-nets). Therefore, as in the case of mono-T-semiflow nets, only the following result can be stated:

Theorem 5.2.2 *For any live and structurally bounded FRT-net with a specification of the average service time s_i for each transition t_i it is not possible to assign probability distribution functions to the transition service times such that the mean cycle time of transition t_j is greater than*

$$\sum_{i=1}^{m} D_i^{(j)} = \sum_{i=1}^{m} v_i^{(j)} s_i \tag{5.16}$$

independently of the topology of the net.

For example, if the above theorem is applied to the net depicted in figure 5.5, the following upper bound for the mean cycle time of transition t_2^1 is obtained:

$$\Gamma_{(t_2^1)} \leq 2s_1^1 + 2s_1^2 + s_2^1 + s_2^2 + 2s_2^3 + 2s_3^1 + 2s_3^2 \tag{5.17}$$

which corresponds to a complete sequentialization of the model.

5.3 Totally open deterministic systems of sequential processes

Totally open deterministic systems of sequential processes with Markovian timing of transitions, defined in section 2.1.4.2, are considered in this section. First of all, we remark that those nets are unbounded (in particular, the buffers are unbounded). Therefore, ergodicity of the marking process must be assured before the computation of steady-state performance measures. A characterization of this ergodicity is stated in section 5.3.1, using analogous arguments to that presented in section 3.3 for unbounded marked graphs. Sequential processes can be seen as *complex servers* that *produce/consume* tokens which are stored at the buffers. The marking of a buffer will be ergodic if the input rate of tokens is less than the service rate of the output complex server.

Once ergodicity characterization has been checked for a given totally open system, a polynomial time computation of the exact mean cycle time of transitions can be achieved with a similar procedure to that used for unbounded marked graphs in section 3.3: the exact mean cycle time of "source complex servers" can be computed firstly (because complex servers are state machines) and, after that, for each output buffer of a complex server, the input flow of tokens must be equal (in steady-state) to the output flow. In this way the throughput of the output transitions can be computed, and the procedure is repeated until all the state machines have been considered. This computation is presented in section 5.3.2.

5.3.1 Characterization of ergodicity

In [FN89b], an ergodicity theorem is proved for a particular class of open synchronized queueing networks. Let us recall now the concept of *saturated net* and the adaptation of the above-mentioned theorem for totally open deterministic systems of sequential processes with Markovian timing.

Definition 5.3.1 (Saturated system) *Let $\langle \mathcal{N}, M_0 \rangle$ be a totally open deterministic system of sequential processes with Markovian timing and b one of its buffers. The net obtained from $\langle \mathcal{N}, M_0 \rangle$ by deleting the* *buffer b and its adjacent arcs is called the saturated system according to b.*

Note that the saturated system according to b behaves like $\langle \mathcal{N}, M_0 \rangle$ in the case in which the buffer b is always marked.

Theorem 5.3.1 [FN89b] *Let $\langle \mathcal{N}, M_0 \rangle$ be a totally open deterministic system of sequential processes with Markovian timing, and $\overline{X}^{(b)}$ be the limit vector of transition throughputs of the saturated net according to b.*

1. *Let $B' \subseteq B$ be the subset of buffers the marking of which can vary independently. If*

$$POST[b] \cdot \overline{X}^{(b)} < PRE[b] \cdot \overline{X}^{(b)}, \quad \forall b \in B' \tag{5.18}$$

then the associated Markov process is positive recurrent.

2. *If there exists a buffer b such that*

$$POST[b] \cdot \overline{X}^{(b)} > PRE[b] \cdot \overline{X}^{(b)} \tag{5.19}$$

then the associated Markov process is transient.

Part 1 of theorem 5.3.1 means that for each buffer (queue) the input flow (arrival rate) must be less than the service rate of the output state machine.

As it is remarked in [FN89b], the application of this ergodicity criterion requires the computation of the steady-state behaviour of all saturated systems that can be obtained from $\langle \mathcal{N}, M_0 \rangle$. This computation is not possible (so far) for all open synchronized queueing networks (in fact, it is possible just for those nets having at most two unbounded places). However the computation is possible for totally open deterministic systems of sequential processes with Markovian timing, because for these nets an efficient method (in fact, polynomial on the number of nodes of the net) for computing the steady-state behaviour exists (it is explained below), and all saturated systems of a totally open deterministic system of sequential processes are totally open deterministic systems of sequential processes again.

Let us illustrate the numerical computation of the ergodicity criterion with the net in figure 5.6. The left and right hand side expressions

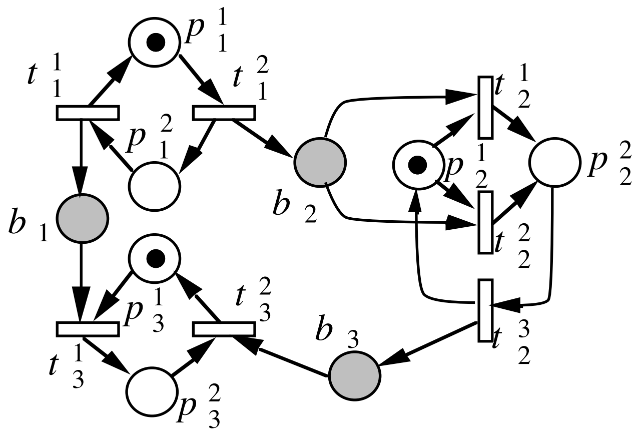


Figure 5.6: A totally open deterministic system of sequential processes.

of theorem 5.3.1.1 for buffer b_2 can be computed considering the state machines $\mathcal{M}_1$ and $\mathcal{M}_2$ in isolation:

$$\begin{aligned} POST[b_2] \cdot \overline{X}^{(b_2)} &= \frac{\lambda_1^1 \lambda_1^2}{\lambda_1^1 + \lambda_1^2}; \\ PRE[b_2] \cdot \overline{X}^{(b_2)} &= \frac{(\lambda_2^1 + \lambda_2^2)\lambda_2^3}{\lambda_2^1 + \lambda_2^2 + \lambda_2^3} \end{aligned} \tag{5.20}$$

where λ_j^i is the rate of the exponentially distributed random variable associated with transition t_j^i. We assume that the conflict at place p_2^1 is solved in favour of transition t_2^1 with probability $\lambda_2^1/(\lambda_2^1 + \lambda_2^2)$ and in favour of t_2^2 with probability $\lambda_2^2/(\lambda_2^1 + \lambda_2^2)$.

The same computation for the buffer b_1 leads to the expresions:

$$\begin{aligned} POST[b_1]T \cdot \overline{X}^{(b_1)} &= \frac{\lambda_1^1 \lambda_1^2}{\lambda_1^1 + \lambda_1^2}; \\ PRE[b_1]T \cdot \overline{X}^{(b_1)} &= \frac{\lambda_3^1 \lambda_3^2}{\lambda_3^1 + \lambda_3^2} \end{aligned} \tag{5.21}$$

The marking of buffer b_3 linearly depends on the marking of the other buffers, so it must not be considered (see section 2.1.4.2, where the following equation was deduced for this net: $M(b_3) = M(b_1) + M(p_1^2) + M(p_3^2) - M(b_2) - M(p_2^1)$).

Then, the system is ergodic if and only if:

$$\frac{\lambda_1^1\lambda_1^2}{\lambda_1^1+\lambda_1^2} < \min\left\{\frac{(\lambda_2^1+\lambda_2^2)\lambda_2^3}{\lambda_2^1+\lambda_2^2+\lambda_2^3}, \frac{\lambda_3^1\lambda_3^2}{\lambda_3^1+\lambda_3^2}\right\} \tag{5.22}$$

5.3.2 Computing the steady-state performance measures

Let us suppose in this section that the system ergodicity conditions given in theorem 5.3.1 are satisfied. The following theorem gives a method for computing efficiently the steady-state behaviour of the connected machines of a totally open deterministic system of sequential processes with Markovian timing. The idea is analogous to that presented in section 3.3 for the case of unbounded marked graphs. A partial order relation can be introduced on the set of strongly connected components of the system (set of sequential processes) as follows: a sequential process $\mathcal{M}_i$ is "greater" than other $\mathcal{M}_j$ iff there exists a directed path from nodes of $\mathcal{M}_i$ to nodes of $\mathcal{M}_j$. Maximal elements of this partial order are sequential processes that have not any input buffer.

Theorem 5.3.2 *Let $\langle\mathcal{N}, M_0\rangle$ be a totally open deterministic system of sequential processes with Markovian timing. If its marking process is ergodic then:*

i) If $\mathcal{M}_i$ has not any input buffer then the limit average marking of each place and the limit throughput of transitions can be computed solving the following marking invariant and flow equations:

$$\begin{array}{l}\sum_{p\in P_i}\overline{M}(p)=1;\\ \overline{X}(t)=\lambda_t\overline{M}(p),\ \textit{if } Pre(p,t)=1, \forall t\in T_i\\ \sum_{t\in{}^\bullet p}\overline{X}(t)=\sum_{t\in p^\bullet}\overline{X}(t), \forall p\in P_i\end{array} \tag{5.23}$$

where $\overline{X}(t)$ is the limit throughput of transition t, λ_t is the rate of the exponentially distributed random variable associated with t, and $\overline{M}(p)$ is the limit average marking of place p.

ii) If $\mathcal{M}_j$ has input buffers that are output buffers of the state machines $\mathcal{M}_{i_1},\ldots,\mathcal{M}_{i_r}$ then the limit average marking of each place *and the limit throughput of transitions can be computed solving the equations:*

$$\begin{array}{l}\sum_{p\in P_j}\overline{M}(p)=1;\\ \overline{X}(t)=\lambda_t\overline{M}(p),\ \textit{if } Pre(p,t)=1,\forall t\in T_j : {}^\bullet t\cap B=\emptyset;\\ \sum_{t\in p^\bullet}\overline{X}(t)=\sum_{t'\in {}^\bullet p}\overline{X}(t'),\forall b\in B: b^\bullet\subset T_j\wedge {}^\bullet b\subset T_{i_1}\cup\ldots\cup T_{i_r};\\ \sum_{t\in {}^\bullet p}\overline{X}(t)=\sum_{t\in p^\bullet}\overline{X}(t),\forall p\in P_j\end{array} \tag{5.24}$$

We remark that the application of the method described in the above theorem has polynomial complexity:

Corollary 5.3.1 *The computation of the limit average marking of places and of the limit throughput of transitions of a totally open deterministic system of sequential processes given by theorem 5.3.2 has polynomial complexity on the net size.*

As an example, let us consider once more the net in figure 5.6. In this case, there exists one state machine without input buffers: $\mathcal{M}_1$. Marking invariant and flow equations for this machine have the form:

$$\begin{array}{l}\overline{M}(p_1^1)+\overline{M}(p_1^2)=1;\\ \overline{X}(t_1^1)=\lambda_1^1\overline{M}(p_1^2);\\ \overline{X}(t_1^2)=\lambda_1^2\overline{M}(p_1^1);\\ \overline{X}(t_1^1)=\overline{X}(t_1^2)\end{array} \tag{5.25}$$

This system can be solved, obtaining:

$$\begin{array}{l}\overline{M}(p_1^1)=\dfrac{\lambda_1^1}{\lambda_1^1+\lambda_1^2};\\ \overline{M}(p_1^2)=\dfrac{\lambda_1^2}{\lambda_1^1+\lambda_1^2};\\ \overline{X}(t_1^1)=\overline{X}(t_1^2)=\dfrac{\lambda_1^1\lambda_1^2}{\lambda_1^1+\lambda_1^2}\end{array} \tag{5.26}$$

Now, for computing the steady-state measures of the other state machines under the assumption of ergodicity (5.22), it is necessary to take into account that

$$\overline{X}(t_1^2)=\overline{X}(t_2^1)+\overline{X}(t_2^2)\ \text{ and }\ \overline{X}(t_3^1)=\overline{X}(t_1^1) \tag{5.27}$$

that is, the input flow of tokens to each buffer in steady-state must be equal to the output flow, and:

$$\begin{array}{l}\overline{M}(p_2^1) = 1 - \overline{M}(p_2^2); \;\; \overline{M}(p_2^2) = \dfrac{\lambda_1^1\lambda_1^2}{(\lambda_1^1+\lambda_1^2)\lambda_2^3}; \\ \overline{X}(t_2^1) = \dfrac{\lambda_1^1\lambda_1^2\lambda_2^1}{(\lambda_1^1+\lambda_1^2)(\lambda_2^1+\lambda_2^2)}; \;\; \overline{X}(t_2^2) = \dfrac{\lambda_1^1\lambda_1^2\lambda_2^2}{(\lambda_1^1+\lambda_1^2)(\lambda_2^1+\lambda_2^2)}; \\ \overline{X}(t_2^3) = \overline{X}(t_3^1) = \overline{X}(t_3^2) = \dfrac{\lambda_1^1\lambda_1^2}{\lambda_1^1+\lambda_1^2}\end{array} \qquad (5.28)$$

5.4 Persistent nets

For live and bounded persistent nets, weak ergodicity of the firing and marking processes is assured (theorem 2.2.2.1). Thus, for these nets a unique limit behaviour exists, and bounds can be computed for the steady-state throughput.

5.4.1 Lower bound for the mean cycle time

As remarked in section 2.2.1, persistent nets are behaviourally defined. This means that a behavioural analysis must be made before computing performance bounds in order to check for the persistency of the net. Few results are known in the literature related to bounds for the performance of bounded persistent nets. A partial result was presented in [Ram74] for 1–bounded persistent nets with deterministic timing. For these nets a behaviourally equivalent 1–bounded marked graph (*behaviour graph*) can be built.

The method consists in drawing the initially marked places and enabled transitions. After that, firing all transitions and drawing the output places and repeating the procedure until a marking in the process is re-found (see figure 5.7). Then, the methods explained in chapter 3 can be applied for computing the bounds for this marked graph and so for the steady-state performances of the initial persistent net. Unfortunately, this analysis is not possible for bounded (k–bounded with $k > 1$) nets when non-deterministic timing is considered.

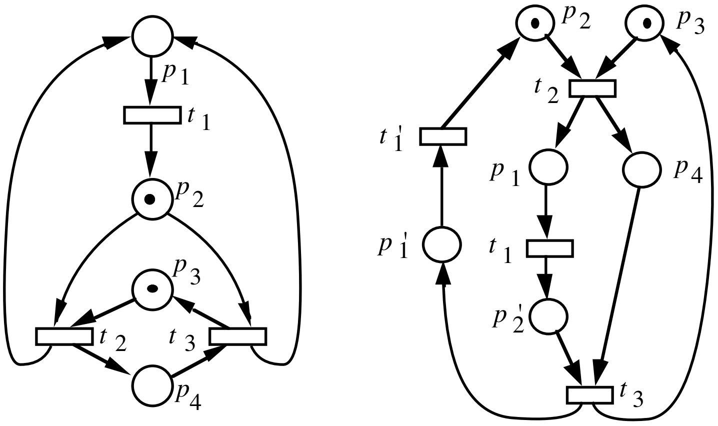


Figure 5.7: Behaviourally equivalent marked graph for 1–bounded persistent net.

Let us now introduce some general results useful for computing bounds for the performance of bounded persistent nets. Later we shall improve some of these results.

Let us consider live bounded persistent nets without implicit places. According to theorem 2.2.1 consistent firing count vectors are proportional to a unique $\vec{\sigma}_R$ (remember definition 2.2.2). Thus the following theorem can be applied for computing a lower bound of the steady-state cycle time of a selected transition t_j taking into account that the vector of visit ratios $\vec{v}^{(j)} = k\vec{\sigma}_R$ is a T-semiflow (non-minimal if there exist more than one) with $v_j^{(j)} = 1$.

Theorem 5.4.1 *For any live and bounded persistent net, a lower bound for the mean cycle time of transition t_j can be computed by the following linear programming problem:*

$$\begin{array}{rll} \Gamma_{(j)} \geq & \textit{maximize} & Y^T \cdot PRE \cdot \vec{D}^{(j)} \\ & \textit{subject to} & Y^T \cdot C = 0 \\ & & Y^T \cdot M_0 = 1 \\ & & Y \geq 0 \end{array} \qquad \text{(LPP17)}$$

where $\vec{D}^{(j)}$ denotes the vector with components $D_i^{(j)} = v_i^{(j)} s_i$, $i = 1, \ldots, m$.

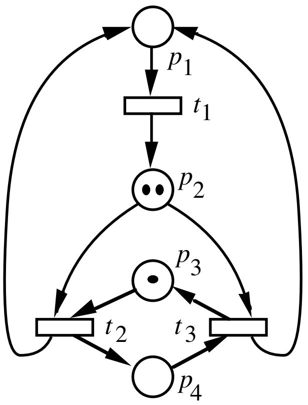


Figure 5.8: Bound (LPP17) non-reachable.

The optimal value of the previous problem is a non-reachable bound in general (i.e., there exist net models such that no stochastic interpretation allows to reach the computed bound). To see it, let us consider for example the net in figure 5.8. Selecting transition t_2, the vector of visit ratios is $\vec{v}^{(2)} = (2, 1, 1)^T$ and the obtained bound is

$$\max\left\{s_2 + s_3, \frac{2s_1 + s_2 + s_3}{2}\right\} \tag{5.29}$$

Now, considering deterministic timing for all transitions with $s_1 = 2$, $s_2 = 60$, and $s_3 = 1$, the obtained bound is 61 while the actual cycle time for transition t_2 is greater because of the sequence $t_1 t_2$ which takes 62 units of time. Nevertheless, for 1–bounded (and ordinary, in order to be live) persistent nets, the bound given by (LPP17) can be always reached: it would be obtained by deriving the equivalent marked graph (according to [Ram74]) and computing the bound, using (LPP3).

Even though the cycle time bound obtained from (LPP17) can be non-reachable for k–bounded (with $k > 1$) persistent nets, it can be pointed out that the bound is finite if and only if the actual cycle time is finite, and this trivially characterizes the liveness of the model.

Theorem 5.4.2 *Let $\langle \mathcal{N}, M_0 \rangle$ be a bounded persistent net. The following three statements are equivalent:*

i) The optimal value of (LPP17) for $\langle \mathcal{N}, M_0 \rangle$ is finite.

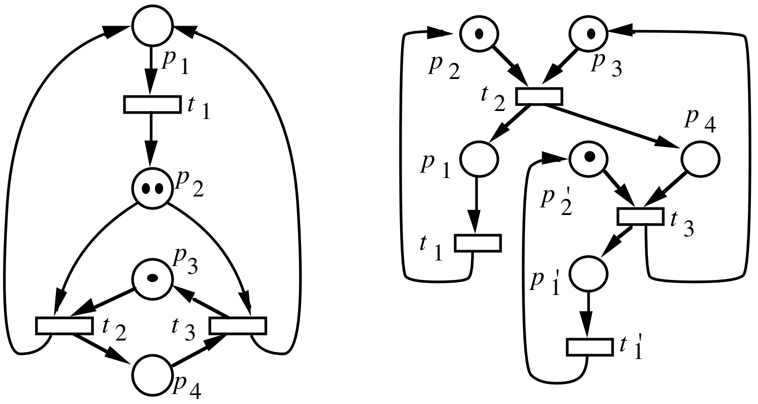


Figure 5.9: Behaviourally equivalent marked graph for deterministic timing.

ii) The actual cycle time of $\langle \mathcal{N}, M_0 \rangle$ *is finite.*

iii) $\langle \mathcal{N}, M_0 \rangle$ *is live.*

The above result is not true for other net classes. In property 5.1.1 it is proved that for mono-T-semiflow nets the actual cycle time can be infinite (so that the net be non-live) while the lower bound obtained from (LPP15) is finite.

5.4.1.1 A reachable bound

Let us now describe a method to compute a reachable lower bound for the mean cycle time of live and bounded *ordinary* persistent nets. Deterministic timing yields the best performance for a given persistent net and average service time associated to transitions. If we consider only deterministic timing, a behaviourally equivalent marked graph can be derived in an analogous way to that proposed in [Ram74] (see the example depicted in figure 5.9):

Step 1. Split the places into instances in such a way that their 1–bounded markings represent conditions for the enabling of transitions.

Step 2. Develop the behaviour graph of the net (under deterministic timing assumption) from the initial marking. Since the original net is live, the behaviour graph *could be* indefinitely extended.

Step 3. To avoid an infinite size of the behaviour graph, identify those instances of places that must be superposed, in such a way that the relative throughput of transitions is preserved: a live and 1–bounded marked graph has been derived.

Considering general timing distributions, the original net and the derived marked graph are not behaviourally equivalent. In fact, the mean cycle time for the marked graph is less than or equal to the one of the original net. Nevertheless, for deterministic timing the equality holds and this provides the following method for computing a reachable lower bound for the mean cycle time of live and bounded ordinary persistent nets with general distribution timing:

Step 1. Develop the behaviour graph for the deterministic case (i.e., the behaviourally equivalent marked graph for deterministic timing).

Step 2. Compute the lower bound for the mean cycle time of the marked graph using (LPP3).

The above computed value is a reachable bound for the mean cycle time of the original net, because in the deterministic case the minimum cycle time for the net is always obtained, and under this condition the cycle times of the behaviour graph and of the original net are equal.

5.4.2 Upper bound for the mean cycle time

Concerning the upper bound for the mean cycle time, the "trivial" one, given by the sum of the average service times of all the transitions weighted by the vector of visit ratios, can be computed, as in the case of mono-T-semiflow nets:

$$\Gamma_{(j)} \leq \sum_{i=1}^{m} D_i^{(j)} = \sum_{i=1}^{m} v_i^{(j)} s_i \tag{5.30}$$

The non-trivial upper bound (dividing by the liveness bound of transitions) which is valid (and reachable) for strongly connected marked graphs, cannot be applied to persistent nets. This can be shown using the same arguments than in section 5.1.2 for live mono-T-semiflow nets, considering the net depicted in figure 5.4, because this net is also persistent.

5.5 Conclusions

In this chapter, extensions of the upper and lower bounds for the mean cycle time (or its inverse, the throughput) of transitions considered in chapters 3 and 4 have been presented for other net subclasses. Unfortunately, the computed bounds are not reachable, in general, for live and structurally bounded FRT-nets (in fact, they are not reachable even for live and structurally bounded mono-T-semiflow nets).

In the particular case of totally open deterministic systems of sequential processes, if Markovian timing is considered, exact computation of performance measures is possible in polynomial time on the net size. The method is similar to that presented in chapter 3 for the case of unbounded marked graphs, but now the strongly connected components are state machines, thus their exact mean cycle time in isolation can be efficiently computed.

Finally, bounds for live and bounded persistent nets have been derived. The method presented here is a generalization to live and bounded persistent nets with stochastic timing of a partial result obtained by C. Ramchandani in [Ram74] for 1–bounded persistent nets for deterministic timing. The lower bound for the mean cycle time has been shown to be reachable.

Further wotk is needed for other behavioural extensions of nets considered here such as, for instance, “réseaux à choix non imposé” (see chapter 2).

Chapter 6

Additional bounds and improvements

In previous chapters we considered the computation of upper and lower bounds for the steady-state throughput of transitions (or its inverse, the mean cycle time). The first part of this chapter (section 6.1) gives an idea about how to obtain bounds for other performance indexes of interest, using the computed throughput bounds and some well-known laws from queueing theory. In particular, bounds for the mean length of queues are derived (mean marking of places, with Petri nets terminology), as well as for the mean response time at places (sojourn time of a token in a place).

In section 6.2, an improvement of the steady-state throughput upper bounds presented in previous chapters is achieved for the particular case of Coxian distributions (having rational Laplace transform) for the service times of transitions. The improvement is obtained by considering the subnets generated by P-semiflows of the net as "almost isolated" nets. By "almost isolated" we mean that transitions of the isolated subnets are considered with finite server semantics with the number of servers equal to their liveness bound in the whole net. The method is specially useful in the case of live and bounded free choice nets because, in this case, the P-semiflows generate subnets having state machine structure (they can be seen as *embedded queueing networks*) which cover the whole net and their mean cycle times can be efficiently computed. In the general case of structurally live and structurally bounded nets, some implicit places can be added to the net in such a way that it is covered by state machines generated by P-semiflows.

6.1 Bounds for other performance indexes

From the knowledge of upper and lower bounds for the steady-state throughput of transitions and from well-known queueing theory laws (such as Little's formula) fast bounds for other performance indexes of interest can be derived. In section 6.1.1, we compute bounds for the mean length of queues. In section 6.1.2, bounds for the maximum length of queues are presented. In the case of live and bounded free choice nets, these bounds give in fact the exact maximum marking of places. In section 6.1.3, bounds for the mean response time at places are derived (mean sojourn time of a token in a place).

6.1.1 Bounds for the mean length of queues

In this section, a fast computation of upper and lower bounds for the limit average marking of places (i.e., length of queues including the customers in service) is proposed.

In section 3.1.1, the following inequality was derived from Little's formula for stochastic Petri nets:

$$\Gamma_{(j)}\overline{M} \geq PRE \cdot \vec{D}^{(j)} \tag{6.1}$$

where $\Gamma_{(j)}$ is the mean cycle time for transition t_j, $\overline{M}$ is the limit average marking of places, PRE is the pre-incidence matrix of the net, and $\vec{D}^{(j)}$ is the vector with the average service demands of transitions as components ($D_i^{(j)} = v_i^{(j)} s_i$, $i = 1, \ldots, m$).

For the net classes considered in this work, the average service times of transitions s_i, $i = 1, \ldots, m$, are known and the visit ratios $v_i^{(j)}$, $i = 1, \ldots, m$, can be computed from the net definition. Then, a lower bound for the average marking of places in steady-state can be computed, from the knowledge of an upper bound for the mean cycle time of transitions.

Theorem 6.1.1 *For any net, a lower bound for the average marking of places in steady-state is:*

$$\overline{M} \geq \overline{M}^{min} = \frac{PRE \cdot \vec{D}^{(j)}}{\Gamma_{(j)}^{max}} \tag{6.2}$$

where $\vec{D}^{(j)}$ *is the vector of service demands of transitions and* $\Gamma_{(j)}^{max}$ *is an upper bound for the mean cycle time of transition* t_j *computed in previous chapters.*

For the computation of an upper bound for the average marking of a given place p_1 in steady-state, let us consider a P-semiflow $Y = (y_1, \ldots, y_n)^T$ whose support includes that place (i.e., $y_1 \neq 0$). We have:

$$Y^T \cdot M_0 = Y^T \cdot \overline{M} \tag{6.3}$$

Therefore,

$$Y^T \cdot M_0 \geq y_1 \overline{M}(p_1) + (y_2, \ldots, y_n) \cdot (\overline{M}^{min}(p_2), \ldots, \overline{M}^{min}(p_n))^T \tag{6.4}$$

Thus,

$$\overline{M}(p_1) \leq \overline{M}^{min}(p_1) + \frac{1}{y_1} Y^T \cdot (M_0 - \overline{M}^{min}) \tag{6.5}$$

and the same condition holds for each P-semiflow including place p_1. Then, the computation of an upper bound for the average marking of places can be formulated in terms of a linear programming problem as follows:

$$\begin{array}{rll} \overline{M}^{max}(p) = & \text{minimize} & \overline{M}^{min}(p) + Y^T \cdot (M_0 - \overline{M}^{min}) \\ & \text{subject to} & Y^T \cdot C = 0 \\ & & Y^T \cdot e_p = 1 \\ & & Y \geq 0 \end{array} \tag{6.6}$$

where $e_p = (0, \ldots, 0, \overbrace{1}^{p}, 0, \ldots, 0)^T$, and the restriction $Y^T \cdot e_p = 1$ allows us to omit the denominator y_p which is assumed to be non-null.

The bound can also be computed from a dual version of the previous problem. For conservative nets, the dual problem is equivalent to the following one, that admits a nice direct interpretation.

Theorem 6.1.2 *For any conservative net, an upper bound for the average marking of place p in steady-state is:*

$$\begin{array}{rll} \overline{M}^{max}(p) = & \textit{maximize} & M(p) \\ & \textit{subject to} & B^T \cdot M = B^T \cdot M_0 \\ & & M \geq \overline{M}^{min} \end{array} \qquad \text{(LPP18)}$$

where the rows of B^T are a basis of the left annullers of C.

In this problem, the maximum average marking of place p is computed, subject to the following restrictions: the average marking must satisfy the place invariant equations, and it must be greater than or equal to the lower bound computed in theorem 6.1.1.

6.1.2 Maximum capacity of queues

In practice, an interesting information for the implementation of the correct dimension of the system is the maximum capacity of queues that is needed for the execution of the processes from the fixed initial state. For live and bounded free choice nets, it is possible to compute in polynomial time on the net size, the exact maximum marking that can be reached from the initial state in each place, solving a linear programming problem. This is based on the fact that the behavioural bound of p, $B(p)$, is equal to the structural bound, $SB(p)$ (cfr. theorem 2.1.12).

Because live and bounded free choice nets are conservative, the problem (LPP2) that defines $SB(p)$ can be easily rewritten leading to the following statement:

Theorem 6.1.3 *For live and bounded free choice nets, the reachable marking bound of places coincides with the structural marking bound obtained solving the following linear programming problem:*

$$\begin{array}{rll} SB(p) = & \textit{maximize} & M(p) \\ & \textit{subject to} & B^T \cdot M = B^T \cdot M_0 \\ & & M \geq 0 \end{array} \qquad \text{(LPP19)}$$

with B^T a basis of the left annullers of C.

The reader is invited to compare the linear programming problems in theorems 6.1.2 and 6.1.3. The first is more constrained: $M \geq \overline{M}^{min} \geq 0$. Therefore, as expected, $\overline{M}^{max}(p) \leq SB(p) = B(p)$.

6.1.3 Other computable bounds

Using fundamental laws of queueing theory, bounds for other performance figures can be computed. As an example, let us consider the computation of bounds for the mean response time at places.

The mean response time $\overline{R}(p_i)$ at a place p_i is the mean value of the sojourn time of a token in this place (i.e., sum of waiting plus service time). From the knowledge of bounds for the throughput of transitions and for the average marking of places, and applying Little's law, upper and lower bounds for the response time at places can be deduced as follows:

$$\begin{aligned} \overline{R}^{max}(p_i) &= \frac{\overline{M}^{max}(p_i)}{PRE[p_i] \cdot \overline{X}^{min}} \\ \overline{R}^{min}(p_i) &= \frac{PRE[p_i] \cdot \vec{D}^{(j)}}{PRE[p_i] \cdot \vec{v}^{(j)}} \end{aligned} \tag{6.7}$$

where $\overline{X}^{min}$, $\overline{M}^{max}$ are bounds computed in previous sections, and $\vec{D}^{(j)}$ and $\vec{v}^{(j)}$ are the vectors of service demands and of visit ratios, respectively, normalized for transition t_j.

6.2 Improving the bounds for Coxian timing

In this section, an improvement of the throughput upper bound of a given transition is presented, based on the computation of the *actual* (i.e., exact) cycle time of the subnets generated by the places involved in some P-semiflows, considered in isolation. Transitions of the isolated subnets are considered with finite server semantics with the number of servers equal to their liveness bound in the whole net.

Basically, we consider those P-semiflows whose support generates a *state machine.* As we remarked in chapter 2, stochastic state machines are nets that can be seen as closed monoclass queueing networks, from the queueing theory point of view. In order to make possible the computation of such exact values, the timing of transitions must be restricted to those distributions that assure the *product form solution* for closed monoclass queueing networks. This condition is obtained if *Coxian* random variables are considered for the timing of transitions. Coxian distributions are characterized by having rational Laplace transform:

Definition 6.2.1 (Coxian distributions) *Let X be a non-negative random variable and $f(t)$ its probability density function. X is said to be Coxian iff the Laplace transform of $f(t)$, $f^*(s) = \int_0^\infty e^{-st} f(t)dt$, is a rational function.*

The Coxian family is generated from exponential distributions by convolutions (*generalized Erlangs*) and mixtures (*hyperexponential* distributions) [Cox55]. One advantage of the use of this family of distributions is that any distribution function can be approximated arbitrarily closely, preserving mean and higher moments, by a Coxian [GP87]. Therefore, the theoretical restriction to these random variables is not really significant. The other main profit of Coxian distributions is that their association to service time of stations of closed monoclass queueing networks with constant routing probabilities assures the product form solution.

In what follows of this chapter, we consider stochastic Petri nets with Coxian distributions associated with service time of transitions. In section 6.2.1 we study the particular case of structurally live and structurally bounded free choice nets, while non-free choice nets are considered in section 6.2.2.

6.2.1 Free choice case

Let us recall from chapter 2 the important result in the structure theory of free choice nets which assures that each minimal P-semiflow of a structurally live and structurally bounded free choice net generates a P-component (cfr. theorem 2.1.8.1). Let us recall also theorem 4.1.1 that gives a lower bound for the mean cyle time of a transition of live and bounded free choice nets in terms of the P-semiflows of the net, by solving the problem (LPP12).

At this point we are able to interpret from a queueing theory point of view the linear programming problem (LPP12) for the case of live and bounded free choice nets:

> *Let Y be a minimal P-semiflow of the net and*
>
> $$\Gamma^{Y}_{(j)} = \frac{Y^T \cdot PRE \cdot \vec{D}^{(j)}}{Y^T \cdot M_0} \tag{6.8}$$
>
> *its corresponding value of the objective function in the problem (LPP12). Then $\Gamma^{Y}_{(j)}$ is the exact cycle time for station (transition) t_j if the closed monoclass queueing network generated by Y (P-component) is considered in isolation, with delay stations (infinite-server semantics for all transitions) and general service time distributions. Moreover, each optimum solution Y^* of (LPP12) corresponds with a slowest closed queueing network (P-component) embedded in the net.*

Let us remark that the throughput of transitions (with infinite-server semantics) in the isolated P-component is *insensible* (i.e., independent) to the distribution of service time of transitions. This is not the case for multiple-server (but finite) semantics. For those servers, the response time at places is due not only to service time but also to the waiting time in queue, and the actual throughput does depend on the number of servers at each station and on the form of the service distributions.

As it is remarked above, the exact cycle time of isolated P-components is computed in (LPP12) assuming infinite-server semantics. A more realistic computation of the cycle time of P-components in the net than that obtained from the complete isolation of these components is considered now.

The knowledge of the liveness bound of transitions of a given net allows to improve the throughput upper bound computed in theorem 4.1.1, for Coxian distributions of service time of transitions (and assuming first-come first-served service discipline).

Theorem 6.2.1 *Let $\langle \mathcal{N}, M_0 \rangle$ be a live and bounded free choice net with constant routing probabilities defining the conflicts resolution policy and Coxian distributions for service time of transitions. For each transition t, let $L(t)$ be its liveness bound. Let Y be a feasible solution of the problem (LPP12) and $\Gamma_{(j)}^{Y_\infty}$ the corresponding value of the objective function. Let $\Gamma_{(j)}^{Y_L}$ be the exact mean cycle time of t_j computed for the isolated P-component generated by Y, with $L(t)$–server semantics for each involved transition t. Then:*

$$\Gamma_{(j)} \geq \Gamma_{(j)}^{Y_L} \geq \Gamma_{(j)}^{Y_\infty} \tag{6.9}$$

where $\Gamma_{(j)}$ is the exact cycle time of t_j in the whole net. Moreover, $\Gamma_{(j)}^{Y_L} = \Gamma_{(j)}^{Y_\infty}$ if and only if the considered P-component contains $\min\{L(t) \mid t \in$ *P-component$\}$ tokens.*

We remark that the above theorem holds also for non-free choice nets if $\Gamma_{(j)}^{Y_L}$ denotes the exact cycle time of t_j computed for the isolated *subnet* (instead of P-component) generated by Y. We restrict ourselves to live and bounded free choice nets because for such nets the subnets generated by minimal P-semiflows are P-components (cfr. theorem 2.1.8.1), and this means that efficient computation of exact values $\Gamma_{(j)}^{Y_L}$ is possible for them (because P-components have state machine structure).

As an example, let us consider the live and bounded free choice net depicted in figure 6.1. Assume that routing rates are equal to 1/3 for t_1, t_2, and t_3, and that t_7, t_8, t_9, t_{10}, t_{11}, t_{12} have exponentially distributed service times with mean values $s_7 = s_8 = s_9 = 10$, $s_{10} = s_{11} = s_{12} = 1$. The P-semiflows of the net are:

$$\begin{array}{l} Y_1 = (1,1,1,1,1,1,1,0,0,0,0,0,0)^T \\ Y_2 = (0,0,0,0,1,0,0,1,0,0,1,0,0)^T \\ Y_3 = (0,0,0,0,0,1,0,0,1,0,0,1,0)^T \\ Y_4 = (0,0,0,0,0,0,1,0,0,1,0,0,1)^T \end{array} \tag{6.10}$$

Then, if the initial marking of p_{11}, p_{12}, and p_{13} is 1 token, and the initial marking of p_1 is N tokens, the lower bound for the mean cycle time derived from (LPP12) is:

$$\Gamma_{(1)}^{(LPP12)} = \max\{30/N, 11, 11, 11\} \tag{6.11}$$

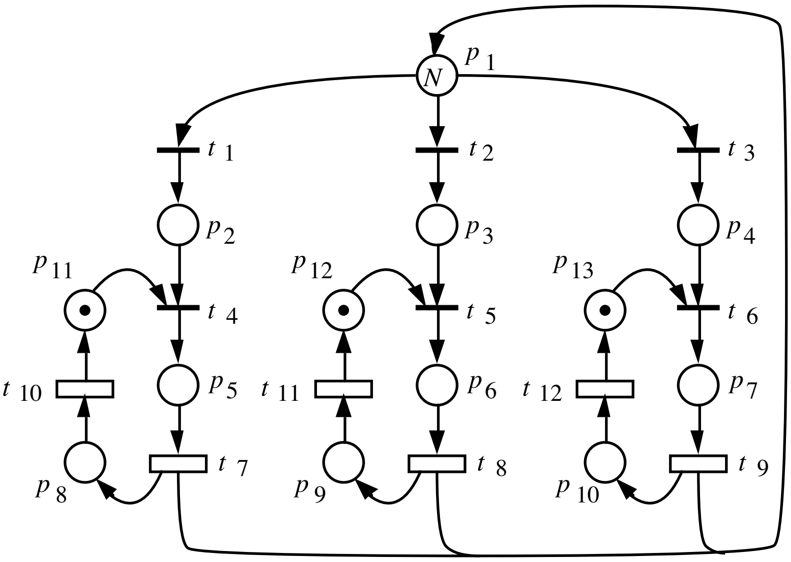


Figure 6.1: A live and bounded free choice net.

For $N = 1$, the previous bound, obtained from Y_1, gives the value 30, while the exact mean cycle time is 31.0626. For $N = 2$, the bound is 15 and it is derived also from Y_1 (mean cycle time of the P-component generated by Y_1, considered in isolation with infinite server semantics for transitions). This bound does not take into account the queueing time at places due to synchronizations (t_4, t_5, and t_6), and the exact cycle time of t_1 is $\Gamma_{(1)} = 21.0513$. For larger values of N, the bound obtained from (LPP12) is equal to 11 (and is given by P-semiflows Y_2, Y_3 and Y_4). Thist bound can be improved if the P-component generated by Y_1 is considered with liveness bounds of transitions t_4, t_5, t_6, t_7, t_8, and t_9 reduced to 1 (which is the liveness bound of these transitions in the whole net). The results obtained for different values of N are collected in table 6.1.

Since the statement of the theorem 6.2.1 holds for every feasible solution Y of (LPP12), it holds for each optimum solution Y^* and the bound computed in theorem 4.1.1 can be eventually improved for Coxian distributions as follows:

Corollary 6.2.1 *An improvement of lower bound for the mean cycle time computed in theorem 4.1.1 can be obtained computing the value* $\Gamma_{(j)}^{Y_L^*}$

N	$\Gamma_{(1)}^{(LPP12)}$	$\Gamma_{(1)}^{Y_{1L}}$	$\Gamma_{(1)}$
1	30	30	31.0626
2	15	20	21.0513
3	11	16.6667	17.7073
4	11	15	16.0336
5	11	14	15.0283
10	11	12	13.0161
15	11	11.3357	12.3481

Table 6.1: Bounds obtained using (LPP12), improvements derived from theorem 6.2.1, and exact values, for different initial markings of p_1 in the net of figure 6.1.

of theorem 6.2.1 for any optimum solution Y^* of the problem (LPP12). Moreover, the improvement is strict if and only if the P-component generated by Y^* contains more than $\min\{L(t) \mid t \in$ *P-component*$\}$ *tokens.*

Let us remark that the reason to constraint the service time distributions to Coxian lies in the fact that, in this case, the exact cycle time for the isolated P-component with $L(t)$–server semantics of transitions can be efficiently computed (for instance, *mean value analysis* algorithm [BB80] can be applied, and it has $O(A^2B)$ worst case time complexity, where $A = Y^* \cdot M_0$ is the number of tokens at the P-component and $B = Y^T \cdot PRE \cdot \mathbb{1}$ is the number of involved transitions). We remark also that any other technique for the computation of a lower bound for the mean cycle time of a product form queueing network can be applied to the P-component generated by Y^*. In particular, *hierarchies of bounds* have been developed for product form queueing networks that guarantee any level of accuracy (including the exact solution), by investing the necessary computational effort [ES83,Sur84,ES86,Sri87]. This provides also a hierarchy of bounds for the mean cycle time of a live and bounded free choice net.

We give now an interpretation of this improvement, from a queueing theory point of view:

> *Both the bound presented in chapter 4 and the presented in this section are based on the computation of the exact mean cycle time of P-components considered in isolation. In the first case, since infinite-server semantics is considered for the isolated subnet, the real (unknown) response time at places is lowerly bounded by the service time of transitions, but waiting time due to synchronizations is not considered at all. Now, the bound for the response time at places is improved taking into account not only the service time but also a part of the queueing time due to synchronizations: that obtained assuming that $L(t)$ severs are always available at each transition t.*

Taking into account that the number of optimum solutions of (LPP12) can be theoretically exponential on the net size, the next question that can be considered is: Which optimum solution(s) of problem (LPP12) should be considered in order to obtain a greater improvement with the application of corollary 6.2.1?

We present now an algorithm for the computation of an improvement of bound given by problem (LPP12), based on a possible heuristic for the selection of some optimum solutions of this problem. Later we justify this heuristic.

Step 0. Compute $L(t)$ for each t, solving the problem (LPP1) of chapter 1.

Step 1. Solve the problem (LPP12). Let $\Gamma^{PS}_{(j)}$ be its optimum value.

Step 2. For $k := 1$ to K solve the linear programming problem (LPP$_k$):

$$\begin{array}{ll} \text{maximize} & Y^T \cdot PRE \cdot G^k_{(j)} \\ \text{subject to} & Y^T \cdot PRE \cdot \vec{D}^{(j)} = \Gamma^{PS}_{(j)} \\ & Y^T \cdot C = 0 \\ & Y^T \cdot M_0 = 1 \\ & Y \geq 0 \end{array}$$

where $G^k_{(j)}$ is a vector with dimension equal to the number of transitions and

$$G_{(j)}^{k}(t_i) = \frac{v_i^{(j)} s_i}{1 + k(L(t_i) - 1)/K}$$

Let Y_k be one optimum solution of (LPP$_k$), $k = 1, \ldots, K$.

Step 3. For $k := 1$ to K solve the exact cycle time $\Gamma_{(j)}^{k}$ of the isolated P-component associated with Y_k assuming $L(t)$–server semantics for each transition t, using (for instance) the mean value analysis algorithm.

Step 4. $\max\{\Gamma_{(j)}^{k} \mid k = 1, \ldots, K\} \geq \Gamma_{(j)}^{PS}$.

The following considerations can be made about the previous algorithm:

a) Step 2 is a heuristic for the selection of a subset of at most K different optimum solutions of (LPP12) (and K can be freely selected). This is because all the feasible solutions of the problems (LPP$_k$) are optimum solutions of (LPP12) (this fact is imposed with the constraints of (LPP$_k$), for which the previous computation of $\Gamma_{(j)}^{PS}$ is necessary, thus it has been done at Step 1).

b) Step 3 is just the application of corollary 6.2.1 to the previously selected optimum solutions. The previous computation of the liveness bounds of transitions is necessary, thus it has been done at Step 0.

c) In Step 4, the “best” of the selected solutions is computed (i.e., the one which gives greater cycle time).

d) The heuristic for the selection of optimum solutions given by Step 2 is based on the computation of the exact cycle time of the isolated P-components, for infinite-server semantics, but with service time associated with t_i:

$$\frac{s_i}{1+(L(t_i)-1)/K}, \quad \text{for } k=1$$
$$\frac{s_i}{1+2(L(t_i)-1)/K}, \quad \text{for } k=2$$
$$\ldots$$
$$\frac{s_i}{L(t_i)}, \quad \text{for } k=K$$

The last case would correspond with saturated P-components, i.e., all transitions t always working with their L(t) servers. The other cases ($k < K$) are intermediate situations.

e) Let us remark that, in the particular case in which the liveness bounds of all transitions were equal, the problems (LPP$_k$) would not select any "better" solution, because all the objective functions would be the same but divided by a different constant. In this case the heuristic used by the above algorithm is not good. Fortunately, this case is easy to detect (at Step 0), and there exists an alternative heuristic for the selection of an optimum solution of (LPP12):

$$\begin{array}{ll} \text{minimize} & Y^T \cdot M_0 \\ \text{subject to} & Y^T \cdot (PRE \cdot \vec{D}^{(j)} - \Gamma^{PS}_{(j)} M_0) = 0 \\ & Y^T \cdot C = 0 \\ & Y \geq 0 \end{array}$$

That is, since all P-components include transitions with the same maximum number of servers, we can expect that the slowest P-component is that with minimum number of tokens, and thus with minimum use of the servers at transitions.

Last but not least, let us remark that the structure-based improvement derived in section 4.1.2 can be taken into account before the application of the algorithm presented above. In other words, the addition of implicit places can generate new slower P-components in the net, that must be considered as feasible solutions for problems (LPP$_k$) of the algorithm.

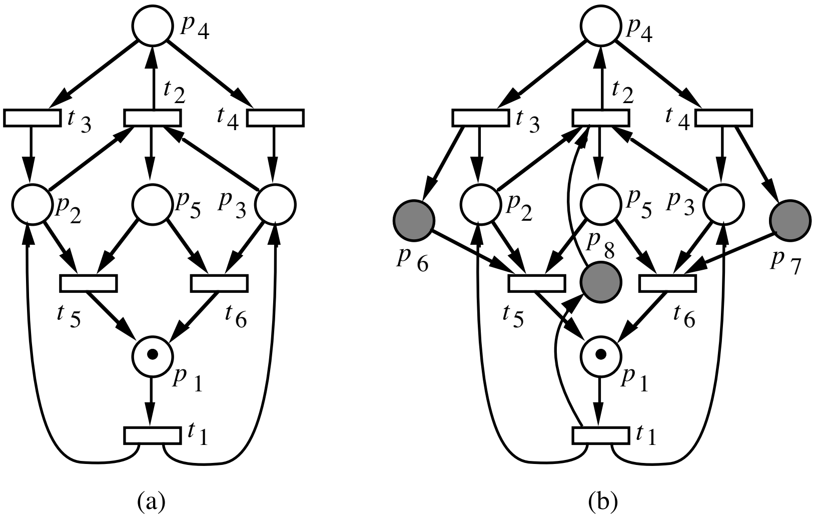


Figure 6.2: Net initially non-covered by P-components, which is covered by four P-components after the addition of three implicit places.

6.2.2 Non-free choice case

For other subclasses of nets (e.g., FRT-nets), P-semiflows do not correspond in general with P-components of the net. In this case, the subnets generated by the support of the P-semiflows have not product form solution and cannot be analysed using the mean value analysis algorithm. In order to solve this problem, a technique consisting of the addition of some implicit places can be used. In fact, for structurally live and structurally bounded nets, a set of implicit places can be added to the net such that it can be covered by P-components, and the algorithm presented in the previous section can be applied to the P-semiflows corresponding with those P-components.

We refer the reader to [CS89c]. Figure 6.2.a has been taken from that paper, and shows a live and bounded FRT-net (which is not a state machine) with a unique minimal P-semiflow which covers all the places:

$$Y_0 = (2, 1, 1, 1, 1)^T \tag{6.12}$$

Therefore, it does not generate a P-component. However, three implicit places can be added (figure 6.2.b) in such a way that four new P-semiflows are created:

$$\begin{array}{l} Y_1 = (1,0,1,1,0,1,0,0)^T \\ Y_2 = (1,1,0,1,0,0,0,1)^T \\ Y_3 = (1,0,0,0,1,0,1,0)^T \\ Y_4 = (1,0,0,1,0,1,1,1)^T \end{array} \qquad (6.13)$$

that generate four P-components which cover the whole net. The approach presented in the previous section can be applied to the resulting net.

6.3 Conclusions

Two different objectives have been considered in this chapter. First, the computation of steady-state bounds for other performance indexes, different from throughput. Second, the improvement of the derived bounds under some assumptions on probability density functions that are usual in queueing theory (rational Laplace transform).

Concerning the first objective, upper and lower bounds for the steady-state mean marking of places (or average length of queues) and for the steady-state mean sojourn time of a token in a place (or average response time) have been algebraically derived, using the upper and lower throughput bounds and Little's law. For live and bounded free choice nets, a reachable upper bound for the maximum capacity of queues has been presented, based on the efficient (structural) computation of the marking bound of places for these nets.

Related to the second objective, an improvement of the lower bound for the mean cycle time has been achieved for Coxian distributed service time of transitions. The improvement is based on considering the cycle time of the subnets generated by P-semiflows of the net, with multiple server semantics for transitions (number of servers equal to the liveness bounds of transitions in the whole net). In the case of live and bounded free choice nets, since all minimal P-semiflows generate P-components (i.e., state machine structure subnets), classical techniques from product form queueing networks can be applied: computation of hierarchies of lower bounds for the steady-state cycle time or, by investing the necessary computational effort, the computation of the exact mean cycle time using, for instance, the mean value analysis. For more general net subclasses, minimal P-semiflows do not generate subnets with state machine structure. Nevertheless, for structurally live and structurally bounded nets, a set of implicit places can be added such that the resulting net is covered by P-components generated by minimal P-semiflows.

Further work must be done in order to improve the throughput lower bound presented in previous chapters for the case of Coxian distributed service times.

Chapter 7

Applications to distributed systems

In previous chapters, we have derive upper and lower bounds for the steady-state performance of stations (transitions) of different classes of synchronized queueing networks, using the formalism of stochastic Petri nets (or for the inverse of the throughput, that we call cycle time of the transition).

Now, we present some application examples, taken from the literature, in the fields of distributed computing systems (section 7.1) and manufacturing systems (section 7.2). By taking some already existing examples developed without consideration of the structural restrictions posed by the techniques proposed in this work, we hope to convince the reader that "in nature" there exist some interesting and non-trivial problems that satisfy such restrictions. Many other interesting examples can be shown in the fields of computer architecture, communications, and manufacturing systems.

7.1 Distributed computing systems

In this section, we analise some examples taken from the literature, in the field of distributed computing system. The first of them is the *alternating bit communication protocol*, that can be modelled by means of a strongly connected marked graph. The second one is taken from software applications: an *Ada tasking system* is modelled with a

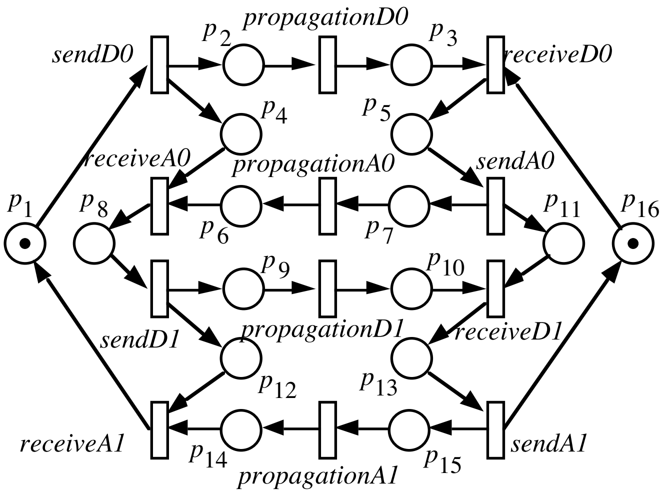


Figure 7.1: The alternating bit protocol.

mono-T-semiflow net. The third is an example of application of marked graphs for the performance evaluation of a complex multiprocessor computer system: the *PADMAVATI machine*. Finally, a *dataflow graph* is analised using a free choice net isomorphic representation.

7.1.1 The alternating bit protocol

Let us consider first a very simple communication protocol, the *alternating bit protocol*, modelled by means of a strongly connected marked graph in [DA84]. Two processes, the *sender* and the *receiver*, exchange messages. Data are sent by the sender together with a *control bit*, 0 or 1. The transfer is divided into two phases. In the first one, the sender sends data together with the 0. Then, it wait for an acknowledgement. During the second phase, the same behaviour occurs with the control bit set to 1. The representation of the behaviour of the protocol is depicted in figure 7.1 using a strongly connected marked graph.

In order to compute performance bounds of the model and to evaluate their accuracy, let us consider that the dimension of data packages is 1 Kbyte, while acknowledgement messages consist of 32 byte. The communication channel speed is 64 Kbit per second. Therefore, sending data and acknowledgement takes 0.125 and 2^{-8} seconds, respectively.

transition	service time
sendD0,D1	0.125
sendA0,A1	2^{-8}
receiveD0,D1	10^{-4}
receiveA0,A1	10^{-4}
propagationD0,D1	0.25
propagationA0,A1	0.25

Table 7.1: Average service times (in seconds) for transitions in the alternating bit protocol.

Receiving time for data or acknowledgement is the same, it takes 10^{-4} seconds. The propagation speed is $2.5 \cdot 10^8$ metres per second. We evaluate the performance of communication for different values of distance between the two processes, ranging from 10^4 to 10^7 metres. Therefore, the propagation time ranges from $2.5 \cdot 10^{-4}$ to 0.25 seconds. Average service times of transitions are collected in table 7.1.

The net in figure 7.1 has a P-semiflow (circuit)

$$Y_1 = (1,1,1,0,1,1,1,1,1,1,1,0,0,1,1,1,0)^T \tag{7.1}$$

whose pre-incidence function covers all the transitions. Therefore, the upper and the lower bounds computed in chapter 3 for the throughput of strongly connected marked graphs have the same value. This means that both bounds give the exact throughput, independently of the probability distribution functions associated with service time of transitions. The throughput and the mean cycle time of the system are detailed in table 7.2, for different values of distance between the two processes.

7.1.2 A software example

In this section, we take an example from software applications. In the thesis of G. Ciardo [Cia89], the effect of different policies on the throughput of an *Ada tasking system* consisting of N_p *producer tasks*, N_c *consumer tasks*, and one *buffer task*, is studied. For simplicity, it

distance (metres)	throughput	cycle time (seconds)
10^4	3.857839	0.259212
10^5	3.728387	0.268213
10^6	2.791639	0.358212
10^7	0.794778	1.258213

Table 7.2: Performance of communication process for different values of distance.

is assumed that each task resides on a different processor. The Ada system is the following:

```
task PRODUCER;
task CONSUMER;
task BUFFER is
     entry DEPOSIT(X : MESSAGE);
     entry REMOVE(X : out MESSAGE);
end BUFFER;
task body PRODUCER is
     begin
       loop
         -- P_1 instructions
         BUFFER.DEPOSIT(X);
         -- P_2 instructions
       end loop;
     end PRODUCER;
task body CONSUMER is
     begin
       loop
         -- C_1 instructions
         BUFFER.REMOVE(X);
         -- C_2 instructions
       end loop;
     end CONSUMER;
task body BUFFER is
     EMPTY : NATURAL := K;
```

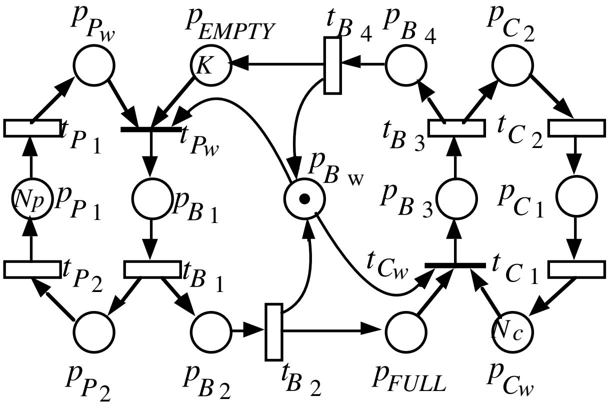


Figure 7.2: Petri net model for the Ada tasking system.

```
begin
  loop
    select
      when EMPTY > 0 =>     --‘‘not full’’
      accept DEPOSIT(X : MESSAGE) do
        -- B_1 instructions (store X)
      end DEPOSIT;
      EMPTY := EMPTY - 1;   -- B_2 instruction
    or
      when EMPTY < K =>     --‘‘not empty’’
      accept REMOVE(X : out MESSAGE) do
        -- B_3 instructions (copy X)
      end REMOVE;
      EMPTY := EMPTY + 1;   -- B_4 instruction
    end select;
  end loop;
end BUFFER;
```

The mono-T-semiflow Petri net model for the above system is depicted in figure 7.2. The data flows from the producer tasks to the buffer task and from there to the consumer tasks. Each producer task executes locally (place p_{P_1}) and occasionally makes an entry call to the buffer task (place p_{P_w}); after a “rendez-vous” with the buffer task,

transition	service time
t_{P_1}	1
t_{P_2}	1/5
t_{C_1}	1
t_{C_2}	1/3
t_{B_1}	1/50
t_{B_2}	1/30
t_{B_3}	1/50
t_{B_4}	1/30

Table 7.3: Average service times for timed transitions in the Ada tasking system.

it performs some other local actions (place p_{P_2}), then restarts the cycle. The behaviour of the consumer tasks is analogous. The buffer task uses a vector with K positions to store data items. The tokens in places p_{EMPTY} and p_{FULL} represent the number of empty and full positions, respectively. An empty call from a producer (consumer) task can be accepted only when there are empty (full) positions. *Guards* "`when EMPTY > 0`" and "`when EMPTY < K`" enforce this constraint. If the vector is neither full nor empty ($0 < M(p_{EMPTY}) = K - M(p_{FULL}) < K$) and both producer and consumer tasks are waiting to "rendez-vous" ($M(p_{P_w}) > 0$ and $M(p_{C_w}) > 0$), when the buffer task becomes ready to accept an entry call ($M(p_{B_w}) = 1$) both guardes are satisfied and the buffer task may "rendez-vous" with either a producer or a consumer task. The Ada language does not specify a criterion for this choice, but leaves the decision to the implementation.

In [Cia89], the effect of enforcing different policies for the choice between producer and consumer tasks is investigated. Here, we do not have to chose one particular policy because the net is mono-T-semiflow, thus the vector of visit ratios is independent of decisions (in this case it is the unity vector; see section 2.1.2). The bounds that we compute are independent of the arbitrary conflict resolution policy and are valid for all of them. In other words, we compute bounds for the purely non-deterministic Ada tasking system.

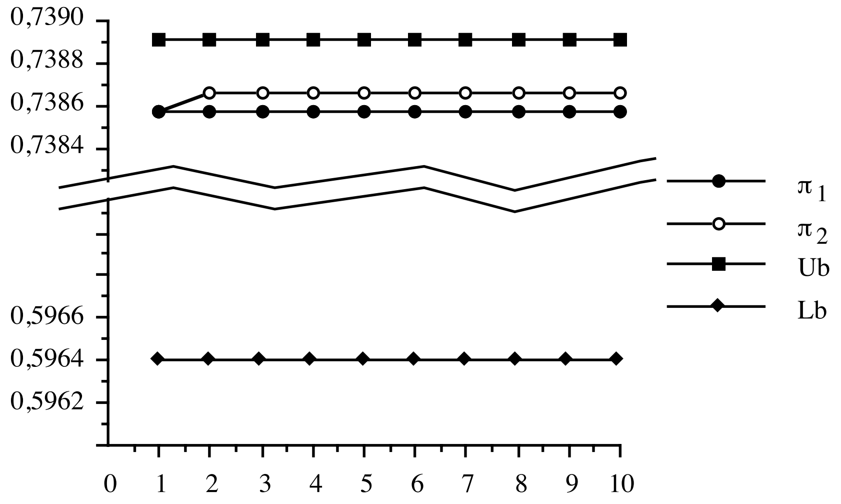


Figure 7.3: Throughput bounds and exact values for policies π_1 and π_2 as a function of K, for $N_p = 5$ and $N_c = 1$.

Average service times of timed transitions are collected in table 7.3. Transitions t_{P_w} and t_{C_w} are immediate.

In figures 7.3, 7.4, 7.5, and 7.6 the throughput bounds obtained using the results of section 5.1 are summarized, in comparison with exact throughputs for exponentially distributed service times of transitions, for different values of N_p, N_c, and K. Two different policies taken from [Cia89] have been used for the computation of exact values:

π_1: It gives absolute priority to the producer tasks when a conflict is reached, so it is optimal when the producer tasks are the bottleneck, since the average throughput of the system is decreased whenever the bottleneck waits.

π_2: It gives absolute priority to the consumer tasks when a conflict is reached, so it is optimal when the consumer tasks are the bottleneck.

The exact values for exponential timing have been computed using the *GreatSPN* software package [Chi87] for the analysis and solution of GSPN models.

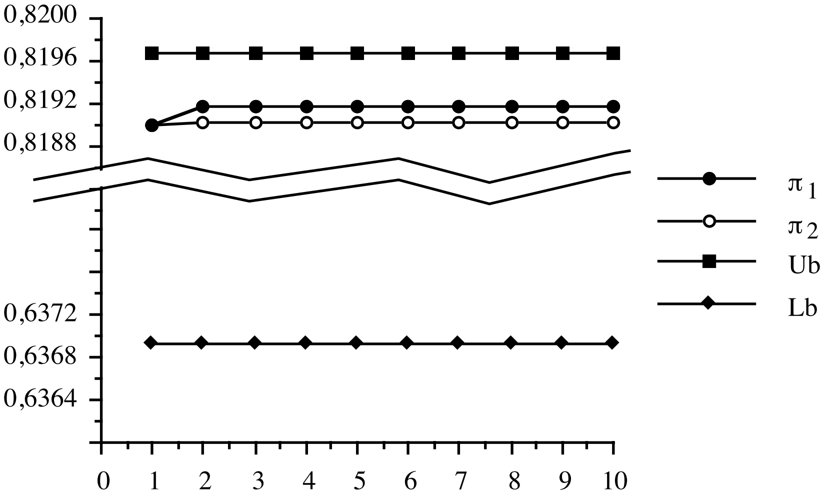


Figure 7.4: Throughput bounds and exact values for policies π_1 and π_2 as a function of K, for $N_p = 1$ and $N_c = 5$.

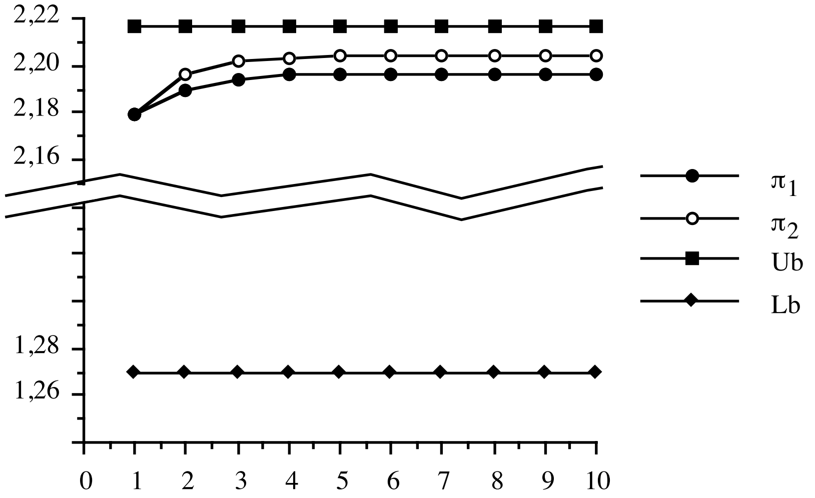


Figure 7.5: Throughput bounds and exact values for policies π_1 and π_2 as a function of K, for $N_p = 5$ and $N_c = 3$.

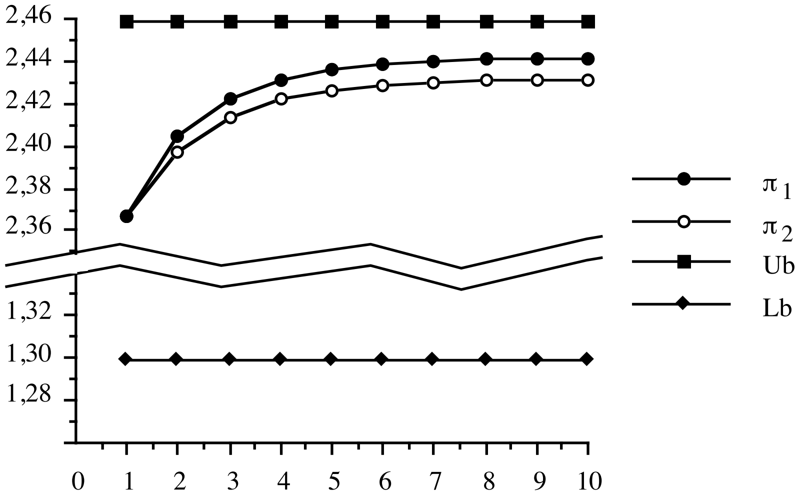


Figure 7.6: Throughput bounds and exact values for policies π_1 and π_2 as a function of K, for $N_p = 3$ and $N_c = 5$.

As can be seen in the figures, the throughput upper bound is quite good for exponentially distributed models, while this is not the case for the lower bound. This is because exponential case (coefficient of variation equal one) "is not very different" from deterministic (null variance). On the other hand, the lower bound on throughput is better for some assignment of distribution timing with coefficient of variation tending to infinity (see section 3.2.3).

7.1.3 The PADMAVATI architecture

As an example of application of marked graphs for the performance evaluation of complex multiprocessor computer systems, let us consider a non-trivial model taken from the literature.

In particular, we consider one of the *coloured Petri net models* of the base software architecture of the PADMAVATI machine (*Parallel Associative Development Machine As a Vehicle for ArTificial Intelligence*), presented in [AMBCC87b]. PADMAVATI is an MIMD modular multiprocessor system based on the *Transputer T424* microprocessor. In [AMBCC87b], a class of Petri net model was derived directly from a pseudo-code specification of the base software implementing the inter-

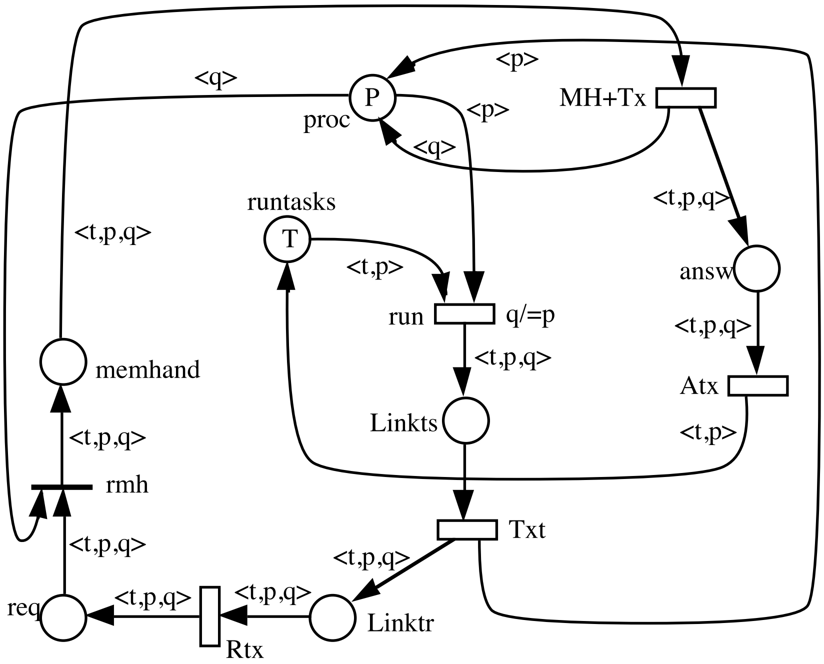


Figure 7.7: CPN model of two-CPU per processor PADMAVATI architecture.

processor communication software. The models were then completed by adding constraints representing the hardware resources.

We report here in figure 7.7 the coloured Petri net model in case of a multiprocessor architecture in which each processor is composed of two Transputer microprocessors, one devoted to the execution of communication and memory handler processes, and the other one devoted to the execution of “client” application tasks. The unfolding of this coloured model yields the marked graph depicted in figure 7.8 in case of a two-processor configuration. In [AMBCC87b] it was shown that a “tandem” model composed of only two processors could be used to accurately estimate the performance of a larger multiprocessor configuration, so that the marked graph model in figure 7.8 can be considered as an accurate performance model of the considered architecture independently of the number of processors.

In the case studied in [AMBCC87b], the evaluation was made be-

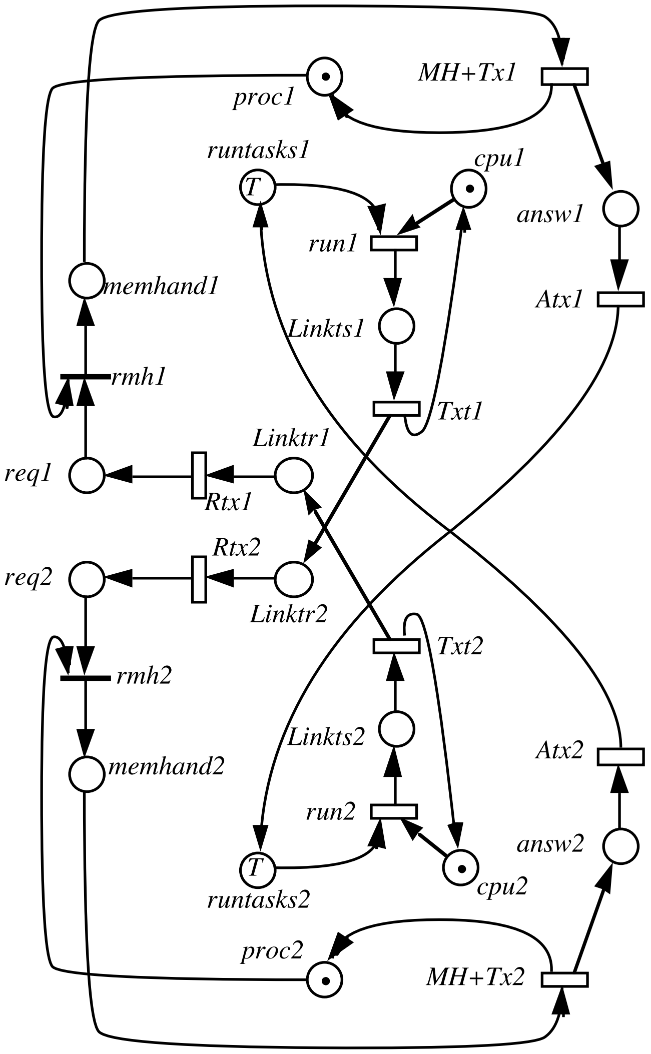


Figure 7.8: Unfolded marked graph model of two-CPU per processor PADMAVATI tandem architecture.

transition	service time
run	$0.1 \div 1$ ms
Txt	11 μs
MH+Tx	50 μs
Rtx	5.6 μs
Atx	16.8 μs

Table 7.4: Average service times for timed transitions in the marked graph model of PADMAVATI architecture.

fore the actual implementation of the prototype of the machine, and the objective of the performance study was the assessment of the effectiveness of multiprogramming in compensating for the large latency of the multistage interconnection network. Only estimates of the average delays of the components (based on their hardware characteristics) were available; no information was instead available on the higher moments and on the form of the probability distributions. In the original work an exponential distribution assumption was adopted in order to apply Markovian analysis techniques, but this choice was clearly arbitrary.

This example represents a classical case in which the computation of performance bounds based on the assumption that only mean values are known is a better answer to the questions posed by the system designers: the "true" value computed by exact numerical solution of a Markov chain is neither needed nor particularly meaningful in this case.

The obtained results for exponentially distributed timing of transitions of the model of figure 7.8 with average service times collected in table 7.4 are summarized in figure 7.9. The exact values, the upper and the lower bounds for the throughput of this marked graph are superposed, for different values of the average service time of transition "*run*", and for different number of tasks (T). It must be pointed out that, while the bounds can be computed practically in zero time independently of the number of tasks, the computation of the exact values increases exponentially with T. See table 7.5 for the CPU time measured on a SUN 3/60 workstation using *GreatSPN* [Chi87].

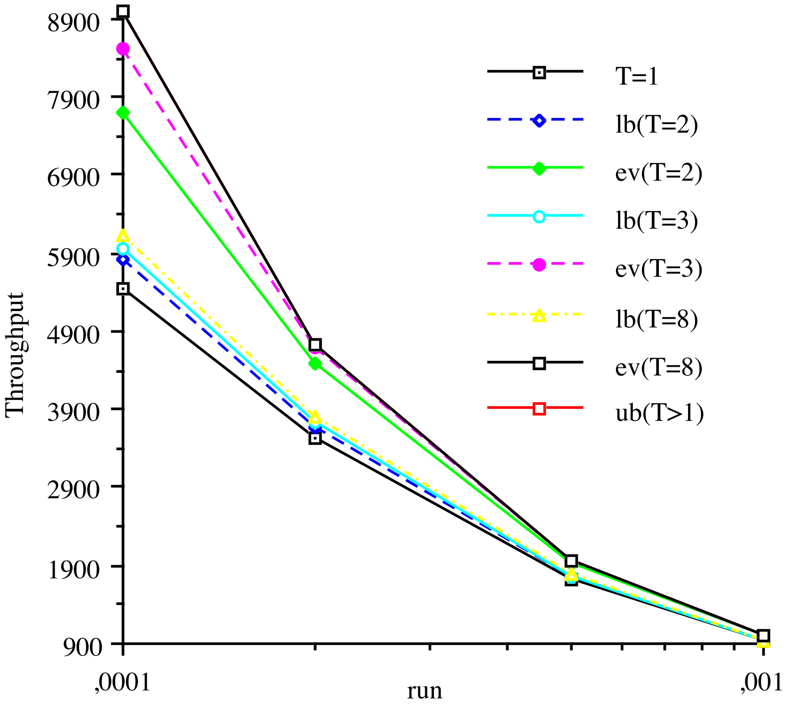


Figure 7.9: Exact values, upper and lower bounds for the throughput of the marked graph model of PADMAVATI architecture, for different service time of transition "*run*", and different number of tasks ($T = 1, 2, 3, 8$).

T	markings	CPU (sec.)
1	25	< 1
2	196	2.1
3	900	11
4	3025	47
8	81225	2540

Table 7.5: Number of reachable markings and CPU time for the computation of the exact values (in seconds, for a SUN 3/60 workstation), for different number of tasks.

Related with the accuracy of the bounds, in the case of only one task ($T = 1$) the lower and the upper bound are equal (thus equal to the exact value). Assuming average service time of transition "*run*" equal to 10^{-4} and $T = 2$, the exact value of the throughput is 7690, while the lower and the upper bounds are 5807 and 9009, respectively, i.e., the exact value is not very close to none of the bounds. For a number of tasks greater than or equal to 8, the exact value coincides with the upper bound (both curves are superposed in the figure 7.9). This means that for higher token populations (i.e., under satutarion conditions), that are the cases in which the Markovian analysis is practically intractable, the upper bound becomes a perfect approximation of the exact value.

7.1.4 A dataflow graph

In [KBB86], *dataflow graph model* is defined as a generalized model of computation. It can be seen as a language for the representation of parallel algorithms, with potential to represent any computational structure, including computer structures (of parallel processors). The chief advantages of dataflow graphs as a computational schema are their compactness and amenability to direct interpretation. Unfortunately, the analysis techniques for dataflow graphs are not yet well-developed.

In [KBB87], performance analysis of computer architectures represented as dataflow graphs via timed Petri nets is proposed. In particular, uninterpreted dataflow graphs are considered. This means that the semantics of the data tokens are removed. The introduced non-determinism is represented by the assignment of routing probabilities at decision points. Uninterpreted dataflow graphs with non-determinism resolved via probabilities are shown to be isomorphic to extended free choice nets.

As an example, the structurally live and structurally bounded free choice net isomorphic to the uninterpreted dataflow graph of figure 7.10 is depicted in figure 7.11. In [KBB87], this net is analysed for the initial marking $M(p_1) = M(p_2) = 1$ and $M(p_i) = 0$ if $i > 2$. Deterministic timing is considered as follows: $s_i^j = \lceil i/3 \rceil$ and $s_i = \lceil i/3 \rceil$ are the service times associated with transitions t_i^j and t_i, respectively. Concerning the conflicts, routing rates $r_k^1 = 1/k$ and $r_k^2 = 1 - 1/k$ are associated with each pair of transitions in conflict t_k^1 and t_k^2, respectively.

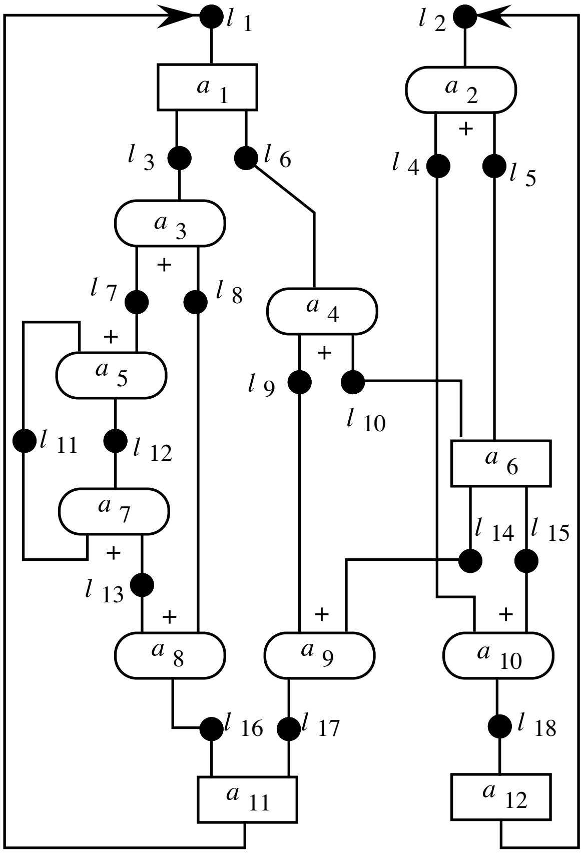


Figure 7.10: An uninterpreted dataflow graph.

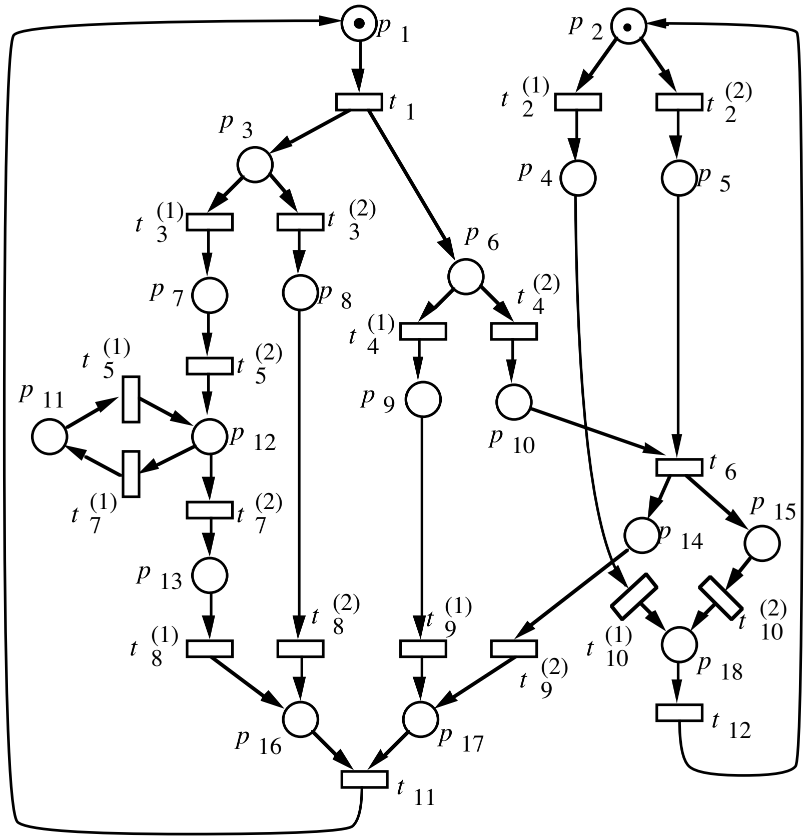


Figure 7.11: A net isomorphic to the uninterpreted dataflow graph.

The vector of visit ratios for transitions can be computed according to theorem 2.1.2, and the upper and lower bounds for the throughput of transition t_1 presented in theorems 4.1.1 and 4.2.1 give the values 0.06667 and 0.032316, respectively. In this case, the obtained throughput upper bound is reached for a deterministic conflicts resolution policy and deterministic timing of transitions (it can be checked after the computation of the reachable bound derived in theorem 4.1.7 from the expanded net).

For this net, if probabilistic routing and exponentially distributed service time of transitions (with the above specified mean values) is assumed, the exact throughput of transition t_1 is 0.05256, which is not very close to none of the bounds.

7.2 Manufacturing systems

Steady-state performance evaluation of some repetitive automated manufacturing systems [CCS90c] modelled by means of stochastic Petri nets is considered in this section. Linear programming problems derived in previous chapters are used to compute tight upper and lower bounds for the performance measures of *job-shop* systems and decision-free *kanban* systems in polynomial time on the net size. The results can be extended to other models in which some decisions are allowed, such as *producer-consumer* systems with *mutual exclusion*.

7.2.1 A job-shop system

In a *job-shop* system, a production route through a sequence of machines is carried on for each job. The set of different products as well as the sequences of visits to machines must be completely defined.

In [HP89], performance evaluation of such systems modelled with marked graphs is studied under a deterministic assumption of the time spent by the jobs on the machines. Using the results presented in this work, the deterministic assumption can be relaxed and reachable bounds for the performance of stochastic models can be computed in polynomial time, from the knowledge of the mean values of the duration of jobs.

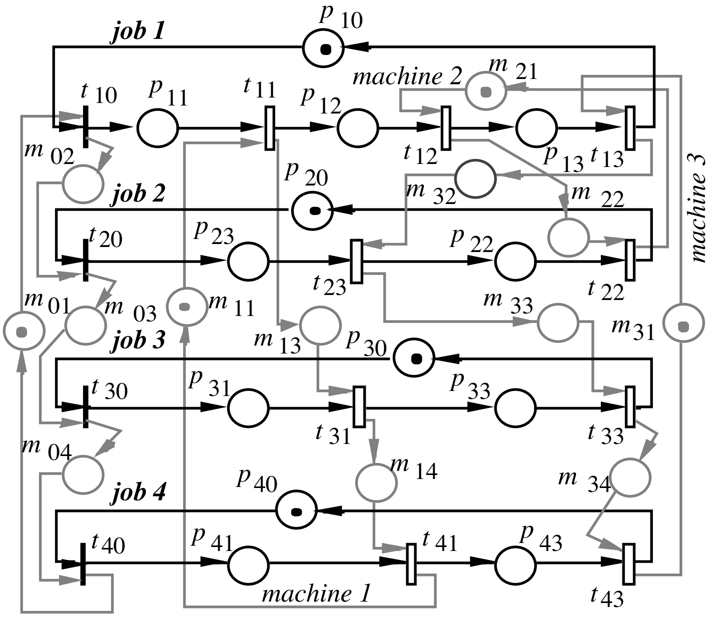


Figure 7.12: A job-shop system modelled with a marked graph.

Let us consider, for instance, the model depicted in figure 7.12. A job-shop system with three machines and four jobs is considered (see [HP89]). The routings of jobs through the machines are modelled with the following horizontal circuits:

Job 1: p_{10}, t_{10}, p_{11}, t_{11}, p_{12}, t_{12}, p_{13}, t_{13}, p_{10};

Job 2: p_{20}, t_{20}, p_{23}, t_{23}, p_{22}, t_{22}, p_{20};

Job 3: p_{30}, t_{30}, p_{31}, t_{31}, p_{33}, t_{33}, p_{30};

Job 4: p_{40}, t_{40}, p_{41}, t_{41}, p_{43}, t_{43}, p_{40}.

Since each machine is assumed to process only one job at a time, other circuits are added which determine the sequencing of the jobs on the corresponding machines:

Machine 1: m_{11}, t_{11}, m_{13}, t_{31}, m_{14}, t_{41}, m_{11};

Machine 2: m_{21}, t_{12}, m_{22}, t_{22}, m_{21};

Machine 3: m_{31}, t_{13}, m_{32}, t_{23}, m_{33}, t_{33}, m_{34}, t_{43}, m_{31}.

They are marked with a token that represents the availability of the machine to process a job.

Let us suppose that only average values of the processing times associated with the machines have been estimated, as follows: $s_{11} = 1$; $s_{12} = 3$; $s_{13} = 3$; $s_{23} = 1$; $s_{22} = 2$; $s_{31} = 2$; $s_{33} = 1$; $s_{41} = 2$; $s_{43} = 1$. Transitions t_{10}, t_{20}, t_{30}, and t_{40} are immediate (time duration equal to zero), since they account for the loading of the job into the system.

The bounds presented in chapter 3 can be computed for this model. The lower bound for the mean cycle time is the optimum value of the problem (LPP3), $\Gamma^{min} = 9$, which is the cycle time for the slowest elementary circuit: m_{21}, t_{12}, p_{13}, t_{13}, m_{32}, t_{23}, p_{22}, t_{22}, m_{21}.

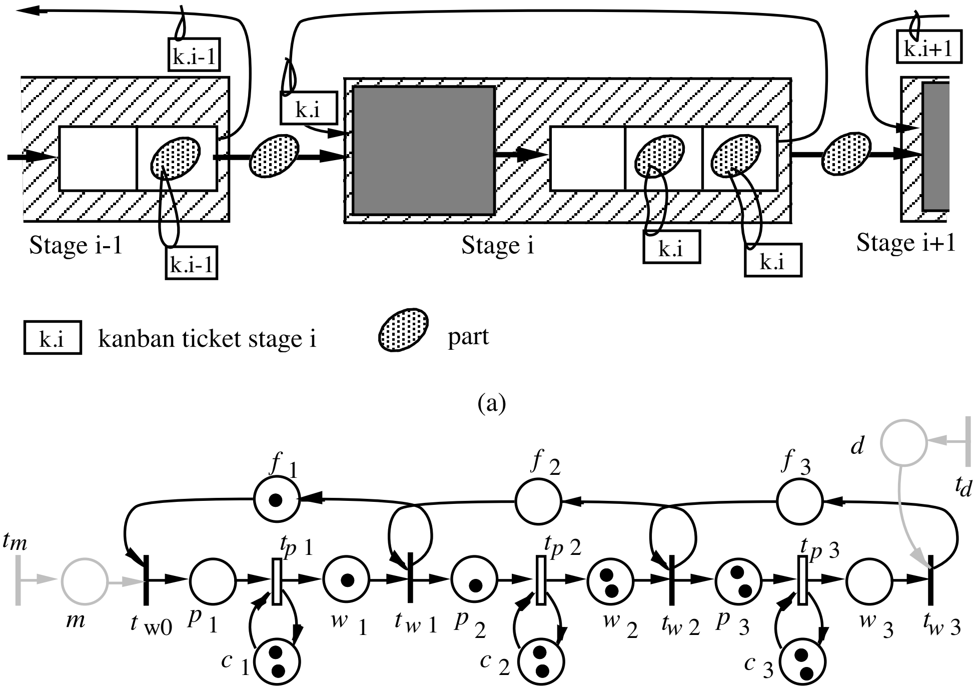
The upper bound for the mean cycle time which follows from theorem 3.2.2 is $\Gamma^{max} = 16$ (in this case, the sum of the average service times, because $L(t) = 1$ for all transition t).

The lower bound for the mean cycle time can be reached, for example, if deterministic timing is assumed (null coefficient of variation for the random variables which define the timing of transitions). On the other hand, the mean cycle time tends to the value of the upper bound if random variables with variances tending to infinity are conveniently selected (as in the proof of theorem 3.2.2) for the timing of transitions. If exponentially distributed random variables (coefficients of variation equal one) are associated with transitions, the actual value for the mean cycle time is $\Gamma = 9.985$, which is quite close to the lower bound (reached with coefficients of variation equal zero).

7.2.2 A kanban system

The *just-in-time* philosophy for the control of manufacturing systems consists of producing just the needed parts at each production stage and at just the right time. *Kanban* control is a way to implement a just-in-time manufacturing system.

A kanban is a ticket that accompanies a part through the several stages of the production system (see figure 7.13.a). When a part of a given stage is consumed by the succeeding stage, the ticket is sent back to trigger the production of a new part. The inventory of a given stage is controlled by the number of kanban tickets at this stage.



(a)

(b)

Figure 7.13: A kanban system and its marked graph representation.

In [DMFDD89], Petri nets have been shown to be well-adapted to provide a unified modelling of kanban systems. Most models encountered in literature can be easily represented by marked graph models.

The steady-state performance results presented in chapter 3 can be applied for analysing quantitatively these models. Without any assumption on the probability distributions associated with transitions, just using their mean values, reachable bounds for the measures of interest can be computed in polynomial time on the net structure.

Let us consider, as an example, the kanban system modelled with a marked graph of figure 7.13.b. We are interested on the computation of the average processing time of a whole part in steady-state, provided that both demands and materials exist for the continual production. In other words, we consider the computation of the mean cycle time of the subsystem in which places m and d have been deleted. Assume that mean values of random variables associated with transitions are: $s(t_{p1}) = 2$; $s(t_{p2}) = 5$; $s(t_{p3}) = 3$. Transitions t_{w0}, t_{w1}, t_{w2}, and t_{w3} are immediate. Infinite-server semantics is assumed for transitions, and this means that each one of the machines modelled with transitions t_{p1}, t_{p2}, and t_{p3} can process two parts simultaneously (if markings of places c_1, c_2, and c_3 equals 2).

For the initial marking depicted in figure 7.13.b, the lower bound for the cycle time (which is reached for deterministic timing) is $\Gamma^{min} = 2.5$, which is the inverse of the throughput of transition t_{p2}. This transition models the *bottleneck* machine of the system, and the *utilization* of this machine is 1 (always busy).

The problem of minimizing the resources in a given manufacturing system for obtaining the same upper bound on throughput of productivity can also be studied. Let us consider the dual problem of (LPP3):

$$\begin{array}{rll} \Gamma^{min} = & \text{minimize} & \gamma \\ & \text{subject to} & C \cdot z + \gamma M_0 \geq PRE \cdot \vec{s} \end{array} \qquad \text{(LPP20)}$$

Then, for a given cost function w for the amount of resources (e.g., the marking weighted with a cost vector W), the initial cost $w(M_0)$ can be minimized without increasing the lower bound Γ^{min} of the mean cycle time by solving the following problem:

$$\begin{array}{ll} \text{minimize} & w(M_0') = W^T \cdot M_0' \\ \text{subject to} & C \cdot z + \Gamma^{min} M_0' \geq PRE \cdot \vec{s} \\ & M_0' \leq M_0 \\ & M_0' \geq 0 \end{array} \tag{LPP21}$$

The restriction $M_0' \leq M_0$ is introduced because, otherwise, a reachable marking from M_0 (with less number of tokens that M_0) could have been obtained as optimum solution, when, in fact the amount of resources is the same for all reachable markings (and the bounds do not change considering any of them as initial marking, see property 3.1.2). This restriction can be deleted if consistency of W is assumed (i.e., $W^T \cdot C = 0$, then $w(M) = w(M_0)$, for all M reachable from M_0). In general, the optimum solution for this problem is non-integer, therefore classical techniques for finding the optimum integer solution could be applied [GN72]. In any case, the optimum value of the objective function (in the non-integer case) is a lower bound for the cost of resources for which a given throughput is obtained.

The problem of minimizing initial cost without increasing the upper bound Γ^{max} of the mean cycle time derived in theorem 3.2.2, can be also considered. In this case, due to the non-linear expression of this bound, only a partial minimization can be expressed in terms of a single linear programming problem. Taking into account (see property 2.1.14) that for strongly connected marked graphs the computation of $L(t_j)$ can be formulated in terms of the problem:

$$\begin{array}{rll} L(t_j) = & \text{maximize} & k \\ & \text{subject to} & M_0 + C \cdot \vec{\sigma} \geq kPRE[t_j] \end{array} \tag{LPP22}$$

We can consider the problem of minimizing the initial cost without decreasing none of the values of $L(t_j)$, as follows:

$$\begin{array}{ll} \text{minimize} & w(M_0') = W^T \cdot M_0' \\ \text{subject to} & M_0' + C \cdot \vec{\sigma} \geq L(t_j) PRE[t_j], \quad \forall t_j \in T \end{array} \tag{LPP23}$$

As for the problem (LPP21), integer programming techniques could be applied for assuring the integrality of the solutions.

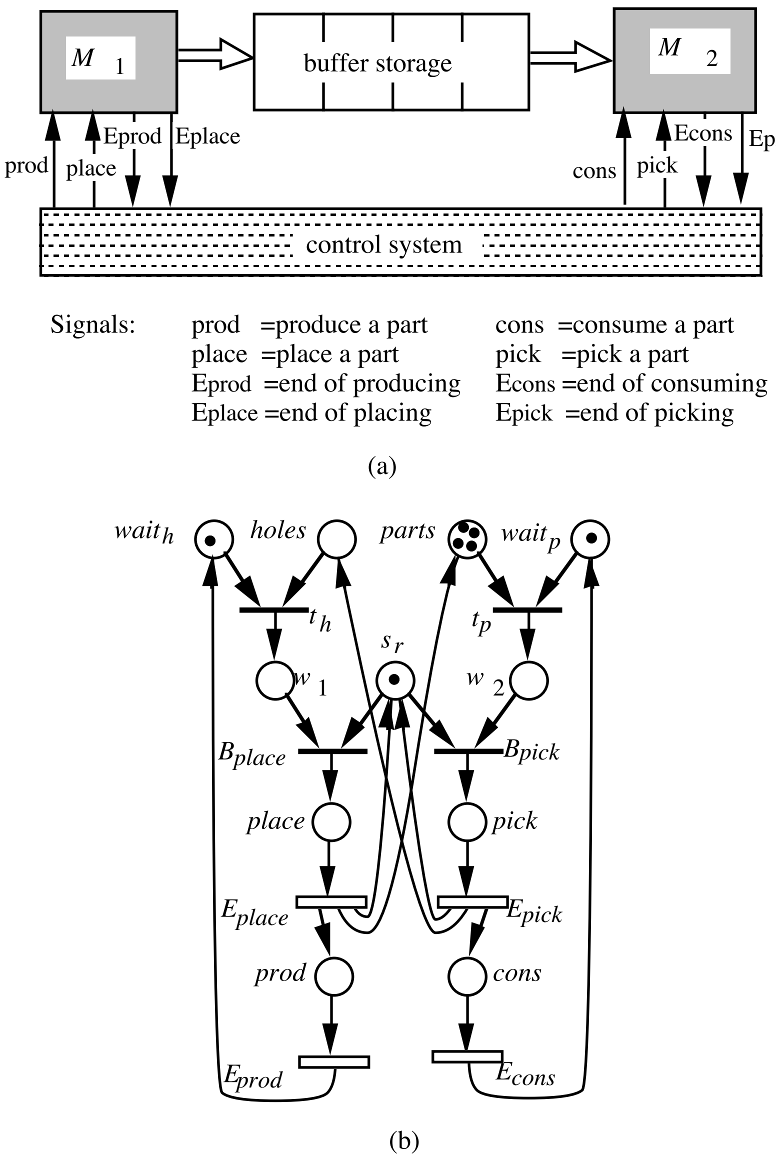
Comming back to the net depicted in figure 7.13.b, for the same number of kanban tickets at each stage, the problem of minimizing resources at places c_1, c_2, and c_3 (capacity of machines), preserving Γ^{min}, can be considered using (LPP21). The result is that the number of tokens at place c_1 can be reduced to 1 without modifying the bound for the mean cycle time.

On the other hand, if the optimization cost criterion consists of reducing as much tokens as possible (i.e., both kanban tickets and capacity of machines), the resolution of problem (LPP21) gives that the capacity of machine 1 (marking of place c_1) can be reduced to 1, and the number of kanbans at stages 1 and 2, can be reduced to 1 and 2, respectively (instead of 2 and 3, as is depicted in figure 7.13.b), without changing the bound for the mean cycle time. In fact the optimum real solution of the problem (LPP21) for $W = \vec{1}$, says that both the capacity of machine 1 and the number of kanbans at the first stage can be reduced to 0.8 units. For the capacity of machine 3 and for the number of kanbans at the third stage, the optimum value is 1.2. In this case, the optimum integer solution is just the excess round of the optimum real solution.

It can be pointed out that for the deterministic timing case the previous minimization of resources preserves the actual mean cycle time. This is not the case for general distribution timing (with non-null coefficient of variation). For example, for exponentially distributed timing of transitions, the actual mean cycle time for the initial marking depicted in figure 7.13.b) is $\Gamma = 2.674$. However, considering as initial marking the one which minimizes the resources (both capacities and kanbans) the actual mean cycle time is $\Gamma' = 3.290$ (i.e., greater than Γ). The result, $\Gamma = 2.674 < \Gamma' = 3.290$, is easily explained by the uncertainty introduced by the stochastic assumption (non-null coefficients of variation).

7.2.3 A producer-consumer system

Let us consider the problem of modelling and evaluating a *producer-consumer* system composed by two machines and a buffer storage, as depicted in figure 7.14.a [Sil85]. The machine M_1 produces parts that are placed at the buffer. The maximum capacity of the buffer is four



(a)

(b)

Figure 7.14: A producer-consumer system and its mono-T-semiflow net representation.

parts. The machine M_2 picks parts from the buffer for processing them. The control system for the production and consumption of parts is depicted in figure 7.14.b by means of a Petri net. Machines M_1 and M_2 cannot operate simultaneously with the buffer, i.e., the pick and place operations are in mutual exclusion (modelled with place s_r).

Obviously, the net in figure 7.14.b is not a marked graph. Transitions B_{place} and B_{pick} can be in an effective conflict. The net is mono-T-semiflow (the unique minimal T-semiflow is the vector with all components equal 1), and the results presented in section 5.1 can be applied. They allow to compute in polynomial time on the net size, performance bounds for the throughput of transitions in steady-state.

Let us suppose that transitions t_h, t_p, B_{place}, and B_{pick} are immediate and that the mean values of random variables associated with the rest of transitions are: $s(E_{place}) = 2$; $s(E_{pick}) = 3$; $s(E_{prod}) = 4$; and $s(E_{cons}) = 2$.

Problem (LPP15) of theorem 5.1.1 gives the following lower bound for the mean cycle time of the net:

$$\begin{aligned}\Gamma^{min} &= \max \{ \; s(E_{place}) + s(E_{pick}), \\ &\qquad\quad (s(E_{place}) + s(E_{pick}))/4, \\ &\qquad\quad s(E_{place}) + s(E_{prod}), \\ &\qquad\quad s(E_{pick}) + s(E_{cons}) \; \} = \\ &= 6\end{aligned} \tag{7.2}$$

As it is remarked in section 5.1, this bound is non-reachable, in general. However, in this case, the lower bound is equal to the actual cycle time for deterministic timing. In fact, if deterministic timing is considered, the buffer storage capacity (initial number of tokens at place *parts*) can be reduced to 1, without modifying the actual mean cycle time. This is because, in this particular case, there exist two different P-semiflows Y_1 and Y_2 (with $||Y_1|| = \{s_r, place, pick\}$ and $||Y_2|| = \{holes, parts, w_1, w_2, s_r, place, pick\}$) involving the same set of timed transitions (E_{place} and E_{pick}). Since $Y_1^T \cdot M_0 = 1$, then the number of tokens at place *parts* can be reduced to 1, and the same optimum value in problem (LPP15) is preserved.

As for the marked graph case, the minimization of tokens preserving the lower bound for the mean cycle time does not preserves the actual mean cycle time for general (non-deterministic) timing. For example,

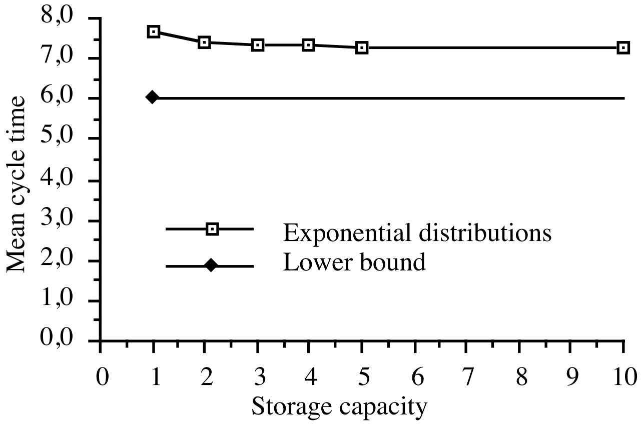


Figure 7.15: Mean cycle time for different storage capacities, under exponentially distributed timing.

considering exponentially distributed timing for the net in figure 7.14.b, the actual mean cycle time decreases if the initial number of tokens at place *parts* (capacity of the storage) increases. In fact, the increase of the cycle time is stopped when the capacity of storage makes insignificant the portion of time during which the machines are waiting for a hole at the buffer or for a part (see figure 7.15).

Conclusions

Timed and stochastic Petri nets have been proposed as a unified model including several extensions of queueing networks with synchronization primitives that have appeared in the literature. This work is a starting point for an efficient performance evaluation of timed and stochastic large Petri net models.

From among the original basic concepts that have been introduced in this work, we remark the definition of weak ergodicity, that allows the estimation of long run performance also in the case of deterministic models. Other interesting original concepts in the framework of Petri nets wich generalize the classical ones of enabling and liveness of transitions are the enabling and liveness bounds. They have shown their significance in the computation of bounds for the steady-state performance of the model. Therefore, they provide a good example of possible interleaving between qualitative and quantitative analysis for timed and stochastic Petri nets. The structural counterpart of enabling bound allows an efficient computation of liveness bound of transitions for some net subclasses such as marked graphs and live and bounded free choice nets.

The intimate relationship between qualitative and quantitative aspects of Petri nets is stressed in this text by the introduction of a new mixed classification criterion. The computability of visit ratios for transitions (a classic performance analysis concept) from different net parameters, such as the structure, the routing rates at conflicts, the initial marking, and the timing of transitions, has propitiated the definition of new interesting net subclasses as well as the identification of other well-known families.

One of our primary goals was to try to deeply bridge two active fields: qualitative theory of Petri nets and stochastic models (stochastic nets and extensions of queueing networks) theory. The benefits have been for both the qualitative and quantitative understanding of such models. From the qualitative point of view, some fundamental new results have been obtained. We remark the appearance of original results about the rank of the incidence matrix of structually live and structurally bounded nets and, in particular, of free choice nets. Several key results in the structure theory of these nets appear as corollaries of the rank theorem. As a by-product of the proof of that theorem, an interesting property about the preservation of structural liveness and structural boundedness of a net after the addition of some particular local schedulers has been shown. We remark also the derivation of interesting qualitative results for totally open deterministic systems of sequential processes such as, for instance, a reversibility characterization.

From the quantitative (performance analysis) point of view, fast algorithms (of polynomial complexity) allow to compute bounds for the throughput of several important classes of synchronized queueing networks. The upper bound on throughput for strongly connected marked graphs was first proposed by Ramchandani in 1974, and then re-discovered and/or re-interpreted by many others, in the framework of the study of the exact performance of timed Petri nets with deterministic timing. The contributions given here in this sense are three: an alternative reformulation in terms of linear programming problems; the proof that deterministic case represents an upper bound in performance independently of the probability distribution also in the framework of stochastic Petri nets; the proof that the upper bound is reachable not only by deterministic but also by stochastic models, with arbitrary values of coefficients of variation.

The presented lower bounds on throughput are new results. The lower bound on throughput consisting of the inverse of the sum of the average service times of all transitions divided by their respective liveness bounds reduces to the trivial sequentialization of all transitions in the case of 1–bounded nets, but has been shown to be reachable for strongly connected marked graphs with some service probability distribution when the coefficient of variation increases.

Some interesting performance monotonicity and reversibility properties, as well as a polynomial complexity liveness characterization of marked graphs have been derived.

The extension to the case of non-strongly connected marked graphs, which has been considered in the literature of deterministically timed nets as straightforward, is less trivial than one can perceive at first glance. We have derived an algorithm for the computation of exact measures for the performance of non-strongly connected marked graphs, from the knowledge of the throughput of their isolated strongly connected components.

Good results have been also obtained for live and bounded free choice nets. A direct application of the same algebraic techniques than for strongly connected marked graphs does not lead in this case to so accurate throughput upper bounds. Nevertheless, with the introduction of more information from the structure and, in particular, using the multisets of circuits of the net (extension from a graph theory point of view of the techniques used for strongly connected marked graphs), reachable upper bounds for the throughput of transitions have been derived, in the case of 1–bounded nets. The lower bounds for the throughput introduced for strongly connected marked graphs have been shown to be also reachable for live and bounded free choice nets.

Concerning the extensions to other nets, the results vary from one class to another. In the case of mono-T-semiflow nets, which are structurally characterized, it has not been possible to derive reachable throughput bounds. On the other hand, for bounded ordinary persistent nets, which are behaviourally characterized, reachable upper bounds for the throughput of transitions have been obtained using an extension of a technique originally proposed by Ramchandani for 1–bounded persistent nets. Specially satisfactory are the results obtained for totally open deterministic systems of Markovian sequential processes. For these nets, both the ergodicity characterization and the computation of exact steady-state performance measures can be achieved in polynomial time on the net size.

Other performance measures, different from throughput, are interesting for the quantitative analysis of distributed systems. As an example, we derive bounds for the queue lengths at stations and for the average response times, from throughput bounds using some information from the net structure and classical fundamental queueing theory laws.

New ideas for the improvement of bounds using some more information from the form of the distribution functions defining the service of transitions have been presented. If Coxian timing is assumed, the exact performance of the "almost isolated" P-components or "embedded queueing networks" (with limited server semantics) of a structurally live and structurally bounded net (which can be computed efficiently because those components are product form queueing networks) can be used in order to improve the throughput upper bounds.

The derived results have been applied for the performance evaluation of several examples taken from literature in the fields of distributed computing systems and manufacturing systems. We have tried to convince the reader that "in nature" there exist some interesting and non-trivial examples that satisfy the restrictions that we have imposed in this work.

In what concerns the future work, related with qualitative aspects of Petri nets, we should mention that a new structurally defined subclass, that of nets with freely related T-semiflows, or FRT-nets, has been introduced in this work whose behavioural properties must be more deeply studied. These nets, that from a performance analysis point of view stand out by the efficient computability of their visit ratios, include the interesting class of free choice nets with the addition of monitors non-disturbing decisions, with a huge panorama of applications in the fields of distributed computer systems, communications, and manufacturing.

Further work must be done in the computation of tight bounds for Coxian timing of transitions. A first step has been taken in this direction concerning the upper bound on throughput, mainly for live and bounded free choice nets. Work is in progress for the improvement of the throughput lower bounds presented here.

Once an efficient computation of performance measures has been obtained, a next step is the solution of optimization problems. Some ideas have been introduced in this text, for the kanban system example, about the utilization of mathematical programming techniques for the achievement of minimization of resources preserving the performance bounds. In any case, more effort should be dedicated to this field.

Finally, we want to stress the fact that the theoretical results presented in this text, being easier not only to compute but also to understand and to interpret than classical “exact” ones, can have a substantial impact on the application of performance evaluation techniques in the early design phases of complex distributed systems.

Bibliography

[AMBB$^+$89] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Transactions on Software Engineering*, 15(7):832–846, July 1989.

[AMBC84] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(2):93–122, May 1984.

[AMBC86] M. Ajmone Marsan, G. Balbo, and G. Conte. *Performance Models of Multiprocessor Systems.* MIT Press, Cambridge, USA, 1986.

[AMBCC87a] M. Ajmone Marsan, G. Balbo, G. Chiola, and G. Conte. Generalized stochastic Petri nets revisited: Random switches and priorities. In *Proceedings of the International Workshop on Petri Nets and Performance Models*, pages 44–53, Madison, WI, USA, August 1987. IEEE-CS Press.

[AMBCC87b] M. Ajmone Marsan, G. Balbo, G. Chiola, and G. Conte. Modeling the software architecture of a prototype parallel machine. In *Proceedings of the 1987 SIGMETRICS Conference*, Banff, Alberta, Canada, May 1987. ACM.

[AMBCD86] M. Ajmone Marsan, G. Balbo, G. Chiola, and S. Donatelli. On the product-form solution of a class of multiple-bus multiprocessor system models. *Journal of Systems and Software*, 6(1,2):117–124, May 1986.

[BB80] S. C. Bruell and G. Balbo. *Computational Algorithms for Closed Queueing Networks*. Elsevier Science Publishers B.V. (North Holland), New York, 1980.

[BBW89] F. Baccelli, N. Bambos, and J. Walrand. Flow analysis of stochastic marked graphs. In *Proceedings of the IEEE Conference on Decision and Control*, 1989.

[BCMP75] F. Baskett, K. M. Chandy, R. R. Muntz, and F. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, April 1975.

[Bes87] E. Best. Structure theory of Petri nets: The free choice hiatus. In W. Brawer, W. Reisig, and G. Rozenberg, editors, *Advances in Petri Nets'86 - Part I*, volume 254 of *LNCS*, pages 168–205. Springer-Verlag, Bad Honnef, Germany, February 1987.

[BG85] S. C. Bruell and S. Ghanta. Throughput bounds for generalized stochastic Petri net models. In *Proceedings of the International Workshop on Timed Petri Nets*, pages 250–261, Torino, Italy, July 1985. IEEE-CS Press.

[BM89] F. Baccelli and A. Makowski. Queueing models for systems with synchronization constraints. *Proceedings of the IEEE*, 77(1):138–161, January 1989.

[Bra83] G. W. Brams. *Réseaux de Petri: Théorie et Pratique. T.1. théorie et analyse.* Masson, Paris, 1983. In French.

[BT81] A. Bertoni and M. Torelli. Probabilistic Petri nets and semi-Markov systems. In *Proceedings of the 2^{nd} European Workshop on Petri Nets*, pages 59–78, Bad Honnef, Germany, September 1981.

[Buz73] J. P. Buzen. Computational algorithms for closed queueing networks with exponential servers. *Communications of the ACM*, 16(9):527–531, September 1973.

[BV84] E. Best and K. Voss. Free choice systems have home states. *Acta Informatica*, 21:89–100, 1984.

[CCCS89] J. Campos, G. Chiola, J. M. Colom, and M. Silva. Tight polynomial bounds for steady-state performance of marked graphs. In *Proceedings of the 3^{rd} International Workshop on Petri Nets and Performance Models*, pages 200–209, Kyoto, Japan, December 1989. IEEE-CS Press.

[CCCS90] J. Campos, G. Chiola, J. M. Colom, and M. Silva. Properties and performance bounds for timed marked graphs. Technical report, Dpto. de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Spain, July 1990.

[CCS89] J. Campos, G. Chiola, and M. Silva. Properties and steady-state performance bounds for Petri nets with unique repetitive firing count vector. In *Proceedings of the 3^{rd} International Workshop on Petri Nets and Performance Models*, pages 210–220, Kyoto, Japan, December 1989. IEEE-CS Press.

[CCS90a] J. Campos, G. Chiola, and M. Silva. Properties and performance bounds for closed free choice synchronized monoclass queueing networks. Research Report GISI-RR-90-2, Dpto. de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Spain, January 1990.

[CCS90b] J. Campos, J. M. Colom, and M. Silva. Improving throughput upper bounds for synchronized queueing networks. Technical report, Dpto. de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Spain, June 1990.

[CCS90c] J. Campos, J. M. Colom, and M. Silva. Performance evaluation of repetitive automated manufacturing systems. In *Proceedings of the Rensselaer's Second International Conference on Computer Integrated Manufacturing*, pages 74–81, Rensselaer Polytechnic Institute, Troy, New York, May 1990. IEEE-CS Press.

[CCS90d] J. M. Colom, J. Campos, and M. Silva. On liveness analysis through linear algebraic techniques. In *Proceedings of Design Methods Based on Nets, ESPRIT Basic Research Action 3148, W.G.3*, Paris, France, June 1990. Deliverables covering the period June 1989 to June 1990.

[CCS91] J. Campos, G. Chiola, and M. Silva. Ergodicity and throughput bounds of Petri nets with unique consistent firing count vector. *IEEE Transactions on Software Engineering*, February 1991. To appear.

[Cha72] K. M. Chandy. The analysis and solutions for general queueing networks. In *Proceedings of the Sixth Anual Princeton Conference on Information Sciences and Systems*, pages 224–228, Princeton, NJ, USA, March 1972.

[CHEP71] F. Commoner, A. Holt, S. Even, and A. Pnueli. Marked directed graphs. *Journal of Computer and System Science*, 5(5):511–523, October 1971.

[Chi87] G. Chiola. A graphical Petri net tool for performance analysis. In *Proceedings of the 3^{rd} International Workshop on Modeling Techniques and Performance Evaluation*, Paris, France, March 1987. AFCET.

[CHW75] K. M. Chandy, U. Herzog, and L. S. Woo. Parametric analysis of queueing networks. *IBM Journal of Res. Develop*, 19(1):36–42, January 1975.

[Cia89] G. Ciardo. *Analysis of Large Stochastic Petri Net Models.* PhD thesis, Department of Computer Science, Duke University, Durham, NC, 1989.

[CMQV89] G. Cohen, P. Moller, J. P. Quadrat, and M. Viot. Algebraic tools for the performance evaluation of discrete event systems. *Proceedings of the IEEE*, 77(1):39–58, January 1989.

[Cou77] P. J. Courtois. *Decomposability: Queueing and Computer System Applications*. Academic Press, New York, 1977.

[Cox55] D. R. Cox. A use of complex probabilities in the theory of stochastic processes. *Proceedings of the Cambridge Philosophical Society*, 51(2):313–319, April 1955.

[CS89a] J. Campos and M. Silva. Steady-state performance evaluation of totally open systems of Markovian sequential processes. In M. Cosnard and C. Girault, editors, *Decentralized Systems*, pages 427–438. North-Holland, Amsterdam, 1990.

[CS89b] J. M. Colom and M. Silva. Convex geometry and semiflows in P/T nets. A comparative study of algorithms for computation of minimal p-semiflows. In *Proceedings of the* 10^{th} *International Conference on Application and Theory of Petri Nets*, pages 74–95, Bonn, Germany, June 1989.

[CS89c] J. M. Colom and M. Silva. Improving the linearly based characterization of P/T nets. In *Proceedings of the* 10^{th} *International Conference on Application and Theory of Petri Nets*, pages 52–73, Bonn, Germany, June 1989.

[DA84] M. Diaz and P. Azema. Petri net based models for the specification and validation of protocols. In G. Rozenberg, H. Genrich, and G. Roucairol, editors, *Advances in Petri Nets 1984*, volume 188 of *LNCS*, pages 101–121. Springer-Verlag, Berlin, Germany, 1984.

[DB78] P. J. Denning and J. P. Buzen. The operational analysis of queueing network models. *ACM Computing Surveys*, 10(3):225–261, September 1978.

[Deo74] N. Deo. *Graph Theory with Applications to Engineering and Computer Science.* Prentice-Hall, Englewood Cliffs, NJ, USA, 1974.

[DLT90] Y. Dallery, Z. Liu, and D. Towsley. Equivalence, reversibility and symmetry properties in fork/join queueing networks with blocking. Technical report, MASI 90-32, University Paris 6, 4 Place Jussieu, Paris, France, June 1990.

[DMFDD89] M. Di Mascolo, M. Y. Frein, Y. Dallery, and R. David. Modeling of kanban systems using Petri nets. In K. Stecke and R. Suri, editors, *Proceedings of the* 3^{rd} *ORSA/TIMS Conference on Flexible Manufacturing Systems*, pages 307–312. Elsevier Science Publishers B.V. (North Holland), 1989.

[Erl09] A. K. Erlang. The theory of probabilities and telephone conversations. *Nyt Tidsskrift Matematik*, 20:33–39, 1909.

[ES83] D. L. Eager and K. C. Sevcik. Performance bound hierarchies for queueing networks. *ACM Transactions on Computer Systems*, 1(2):99–115, May 1983.

[ES86] D. L. Eager and K. C. Sevcik. Bound hierarchies for multiple-class queueing networks. *Journal of the ACM*, 33(1):179–206, January 1986.

[ES90] J. Esparza and M. Silva. On analysis and synthesis of free choice systems. Technical report, GISI-RR-90-10, Dpto. de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Spain, June 1990.

[Esp90] J. Esparza. *Structure Theory of Free Choice Nets.* PhD thesis, Dpto. de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Zaragoza, Spain, June 1990. Research Report GISI-90-03.

[FN85a] G. Florin and S. Natkin. Les réseaux de Petri stochastiques. *Technique et Science Informatiques*, 4(1):143–160, February 1985. In French.

[FN85b] G. Florin and S. Natkin. Les réseaux de Petri stochastiques, 1985. Thesis de Doctorat d'Etat, Université Pierre et Marie Curie, Paris (in French).

[FN86] G. Florin and S. Natkin. One-place unbounded stochastic Petri nets: Ergodicity criteria and steady-state solutions. *Journal of Systems and Software*, 6(1,2):103–115, May 1986.

[FN89a] G. Florin and S. Natkin. Matrix product form solution for closed synchronized queuing networks. In *Proceedings of the 3^{rd} International Workshop on Petri Nets and Performance Models*, pages 29–37, Kyoto, Japan, December 1989. IEEE-CS Press.

[FN89b] G. Florin and S. Natkin. Necessary and sufficient ergodicity condition for open synchronized queueing networks. *IEEE Transactions on Software Engineering*, 15(4):367–380, April 1989.

[GN67] W. J. Gordon and G. F. Newell. Closed queueing systems with exponential servers. *Operations Research*, 15:254–265, 1967.

[GN72] R. S. Garfinkel and G. L. Nemhauser. *Integer Programming.* John Wiley & Sons, 1972.

[GP87] E. Gelenbe and G. Pujolle. *Introduction to Queuing Networks.* John Wiley & Sons, 1987.

[Hac72] M. H. T. Hack. Analysis of production schemata by Petri nets. M. S. Thesis , TR-94, M.I.T.,Boston, USA, 1972.

[Hil88] H. P. Hillion. Timed Petri nets and application to multistage production systems. In *Proceedings of the* 9^{th} *European Workshop on Applications and Theory of Petri Nets*, pages 164–182, Venice, Italy, June 1988.

[HL84] P. Heidelberger and S. S. Lavenberg. Computer performance evaluation methodology. *IEEE Transactions on Computers*, 33(12):1195–1220, December 1984.

[HP89] H. P. Hillion and J. M. Proth. Performance evaluation of job-shop systems using timed event-graphs. *IEEE Transactions on Automatic Control*, 34(1):3–9, January 1989.

[HT83] P. Heidelberger and K. S. Trivedi. Analytic queueing models for programs with internal concurrency. *IEEE Transactions on Computers*, 32:73–82, January 1983.

[HV85] M. A. Holliday and M. K. Vernon. A generalized timed Petri net model for performance analysis. In *Proceedings of the International Workshop on Timed Petri Nets*, pages 181–190, Torino, Italy, July 1985. IEEE-CS Press.

[IA89] S. M. R. Islam and H. H. Ammar. On bounds for token probabilities in a class of generalized stochastic Petri nets. In *Proceedings of the* 3^{rd} *International Workshop on Petri Nets and Performance Models*, pages 221–227, Kyoto, Japan, December 1989. IEEE-CS Press.

[Jac63] J. R. Jackson. Jobshop-like queueing systems. *Management Science*, 10(1):131–142, October 1963.

[JLL77] N. Jones, L.H. Landweber, and Y. Lien. Complexity of some problems in Petri nets. *Theoretical Computer Science*, 4:277–299, 1977.

[Kar84] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[KBB86] K. M. Kavi, B. P. Buckles, and U. N. Bhat. A formal definition of dataflow graph models. *IEEE Transactions on Computers*, 35(11):940–948, November 1986.

[KBB87] K. M. Kavi, B. P. Buckles, and U. N. Bhat. Isomorphisms between Petri nets and dataflow graphs. *IEEE Transactions on Software Engineering*, 13(10):1127–1134, October 1987.

[Kel76a] T.W. Keller. *Computer System Models with Passive Resources*. PhD thesis, University of Texas at Austin, Austin, TX, USA, 1976.

[Kel76b] F. P. Kelly. Networks of queues. *Advances on Applied Probability*, 8:416–432, 1976.

[Kle75] L. Kleinrock. *Queueing Systems Volume I: Theory*. John Wiley & Sons, New York, NY, USA, 1975.

[Kle76] L. Kleinrock. *Queueing Systems Volume II: Computer Applications*. John Wiley & Sons, New York, NY, USA, 1976.

[Kri84] J. Kriz. Throughput bounds for closed queueing networks. *Performance Evaluation*, 4:1–10, 1984.

[Lau87] K. Lautenbach. Linear algebraic calculation of deadlocks and traps. In K. Voss, H. Genrich, and G. Rozenberg, editors, *Concurrency and Nets*, pages 315–336. Springer-Verlag, Berlin, 1987.

[Lav89] S. S. Lavenberg. A perspective on queueing models of computer performance. *Performance Evaluation*, 10:53–76, 1989.

[LB86] J.Y. Le Boudec. A BCMP extension to multiserver stations with concurrent classes of customers. In *Proceedings of PERFORMANCE'86 and ACM SIGMETRICS*, Raleigh, NC, USA, May 1986.

[Lit61] J. D. C. Little. A proof of the queueing formula $L = \lambda W$. *Operations Research*, 9:383–387, 1961.

[LR78] L. H. Landweber and E. L. Robertson. Properties of conflict-free and persistent Petri nets. *Journal of the ACM*, 25(3):352–364, April 1978.

[LR87] A. A. Lazar and T. G. Robertazzi. Markovian Petri net protocols with product form solution. In *Proceedings of the Conference on Computer Communications*, pages 1054–1062, Washington, DC, USA, 1987. IEEE-CS Press.

[LZGS84] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik. *Quantitative System Performance.* Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1984.

[Mag84] J. Magott. Performance evaluation of concurrent systems using Petri nets. *Information Processing Letters*, 18:7–13, 1984.

[Mai87] D. Mailles. *Files d'Attente Descriptives pour la Modelisation de la Synchronisation dans les Systemes Informatiques.* PhD thesis, Laboratoire MASI, Univ. P. et M. Curie, Paris, France, September 1987. Technical Report 202 (in French).

[MB86] M. Minoux and G. Bartnik. *Graphes, Algorithmes, Logiciels.* Dunod Informatique, Paris, France, 1986.

[Mol81] M.K. Molloy. *On the Integration of Delay and Throughput Measures in Distributed Processing Models.* PhD thesis, UCLA, Los Angeles, CA, USA, 1981.

[Mol82] M. K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Transaction on Computers*, 31(9):913–917, September 1982.

[Mol85] M.K. Molloy. Fast bounds for stochastic Petri nets. In *Proceedings of the International Workshop on Timed*

Petri Nets, pages 244–249, Torino, Italy, July 1985. IEEE-CS Press.

[MP70] J. J. Moder and C. R. Phillips. *Project Management with CPM and PERT.* Van Nostrand, New York, USA, 1970.

[MS82] J. Martínez and M. Silva. A simple and fast algorithm to obtain all invariants of a generalized Petri net. In C. Girault and W. Reisig, editors, *Application and Theory of Petri Nets*, volume 52 of *Informatik-Fachberichte*, pages 301–310, Berlin, Germany, 1982. Springer-Verlag.

[Mur77] T. Murata. Circuit theoretic analysis and synthesis of marked graphs. *IEEE Transactions on Circuits and Systems*, 24(7):400–405, July 1977.

[Mur83] K. G. Murty. *Linear Programming.* John Wiley & Sons, 1983.

[Mur85] T. Murata. Use of resource-time product concept to derive a performance measure of timed Petri nets. In *Proceedings 1985 Midwest Symposium Circuits and Systems*, Louisville, USA, August 1985.

[Mur89] T. Murata. Petri nets: properties, analysis, and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[Par66] R. J. Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, October 1966.

[Pet66] C.A. Petri. Communication with automata. Technical report, Rome Air Dev. Center, New York, NY, USA, 1966. Tech. Rep. RADC-TR-65-377.

[Pet81] J.L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice-Hall, Englewood Cliffs, NJ, USA, 1981.

[PNPM87] *Proceedings of the International Workshop on Petri Nets and Performance Models*, Madison, WI, USA, August 1987. IEEE-CS Press.

[PNPM89] *Proceedings of the 3^{rd} International Workshop on Petri Nets and Performance Models*, Kyoto, Japan, December 1989. IEEE-CS Press.

[Ram74] C. Ramchandani. *Analysis of Asynchronous Concurrent Systems by Petri Nets.* PhD thesis, MIT, Cambridge, MA, USA, February 1974.

[Rev84] D. Revuz. *Markov Chains.* North-Holland, Amsterdam, The Netherlands, 1984.

[RH80] C. V. Ramamoorthy and G. S. Ho. Performance evaluation of asynchronous concurrent systems using Petri nets. *IEEE Transactions on Software Engineering*, 6(5):440–449, September 1980.

[RK75] M. Reiser and H. Kobayashi. Queueing networks with multiple closed chains: Theory and computational algorithms. *IBM Journal of Res. Develop*, 19(3):283–294, May 1975.

[RL80] M. Reiser and S. S. Lavenberg. Mean value analysis of closed multichain queueing networks. *Journal of the ACM*, 27(2):313–322, April 1980.

[RL81] M. Reiser and S. S. Lavenberg. Corrigendum: Mean value analysis of closed multichain queueing networks. *Journal of the ACM*, 28(3):629, July 1981.

[Ros83] S. M. Ross. *Stochastic Processes.* John Wiley & Sons, New York, 1983.

[RP84] R. R. Razouk and C.V. Phelps. Performance analysis using timed Petri nets. In *Proceedings of the International Conference on Parallel Processing*, pages 126–129, August 1984.

[SB88] Y. Souissi and N. Beldiceanu. Deterministic systems of sequential processes; theory and tools. In F.H. Vogt, editor, *Concurrency 88*, volume 335 of *LNCS*, pages 380–400. Springer-Verlag, October 1988.

[SC88] M. Silva and J. M. Colom. On the computation of structural synchronic invariants in P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1988*, volume 340 of *LNCS*, pages 386–417. Springer-Verlag, Berlin, 1988.

[Sif78] J. Sifakis. Use of Petri nets for performance evaluation. *Acta Cybernetica*, 4(2):185–202, 1978.

[Sil85] M. Silva. *Las Redes de Petri en la Automática y la Informática.* Editorial AC, Madrid, 1985. In Spanish.

[Sil87] M. Silva. Towards a synchrony theory for P/T nets. In K. Voss, H. Genrich, and G. Rozenberg, editors, *Concurrency and Nets. Advances in Petri Nets*, pages 435–460. Springer-Verlag, Berlin, 1987.

[SMK82] C. H. Sauer, E. A. MacNair, and J. F. Kurose. The research queueing package: past, present, and future. In *Proceedings of the 1982 National Computer Conference.* AFIPS, 1982.

[Sri87] M. M. Srinivasan. Succesively improving bounds on performance measures for single class product form queueing networks. *IEEE Transactions on Computers*, 36:1107–1112, September 1987.

[Sur84] R. Suri. Generalized quick bounds for performance of queueing networks. *Computer Performance*, 5(2):116–120, June 1984.

[TPN85] *Proceedings of the International Workshop on Timed Petri Nets*, Torino, Italy, July 1985. IEEE-CS Press.

[TV84] P. S. Thiagarajan and K. Voss. A fresh look at free choice nets. *Information and Control*, 61(2):85–113, May 1984.

[Vog89] W. Vogler. Live and bounded free choice nets have home states. Report, Institut für Informatik, TU München, Germany, 1989.

[VZL87] M. Vernon, J. Zahorjan, and E. D. Lazowska. A comparison of performance Petri nets and queueing network models. In *Proceedings of the* 3^{rd} *International Workshop on Modelling Techniques and Performance Evaluation*, pages 181–192, Paris, France, March 1987. AFCET.

[ZSEG82] J. Zahorjan, K. C. Sevcik, D. L. Eager, and B. Galler. Balanced job bound analysis of queueing networks. *Communications of the ACM*, 25(2):134–141, February 1982.

[Zub85] W.M. Zuberek. Performance evaluation using timed Petri nets. In *Proceedings of the International Workshop on Timed Petri Nets*, pages 272–278, Torino, Italy, July 1985. IEEE-CS Press.