TESIS DE LA UNIVERSIDAD
DE ZARAGOZA

2015

55

María Estíbaliz Fraca Santamaría

Fluidization of Petri nets to improve the analysis of Discrete Event Systems

Departamento

Informática e Ingeniería de Sistemas

Director/es

Silva Suárez, Manuel
Júlvez Bueno, Jorge Emilio

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Servicio de Publicaciones

ISSN 2254-7606

Universidad Zaragoza
1542

Tesis Doctoral

FLUIDIZATION OF PETRI NETS TO IMPROVE THE ANALYSIS OF DISCRETE EVENT SYSTEMS

Autor

María Estíbaliz Fraca Santamaría

Director/es

Silva Suárez, Manuel
Júlvez Bueno, Jorge Emilio

UNIVERSIDAD DE ZARAGOZA
Informática e Ingeniería de Sistemas

2015

UNIVERSIDAD DE ZARAGOZA

PHD THESIS

PhD in Systems Engineering and Computer Science
Doctorado en Ingeniería de Sistemas e Informática

Fluidization of Petri nets to improve the analysis of Discrete Event Systems

Fluidificación de Redes de Petri para mejorar el análisis de Sistemas de Eventos Discretos

M. Estíbaliz Fraca Santamaría

Thesis Advisors: Manuel Silva Suárez
Jorge E. Júlvez Bueno

Departamento de Informática e Ingeniería de Sistemas (DIIS)
Universidad de Zaragoza

May 2015

A Laura Recalde
A Mónica Sanagustín

Mujeres sonrientes, trabajadoras, coherentes, luchadoras,
que disfrutasteis de vuestras cortas vidas en plenitud.
Habéis sido y seréis referencia para mí.

La investigación, el desarrollo de una profesión, el compromiso...,
son una aventura de darse a los demás, de ser un eslabón en la
historia de la humanidad, de sembrar y ser inspiración
para otros, que serán quienes recojan los frutos.

Agradecimientos - Acknowledgement

Quiero aprovechar este espacio para dar las gracias a todas las personas que de una manera o de otra habéis contribuido al desarrollo de esta tesis doctoral.

En especial, a mis directores de tesis, Jorge Júlvez y Manuel Silva. A Jorge, que me explicaste con ilusión tu primera idea sobre “estudiar un formalismo nuevo entre las redes de Petri continuas y las discretas, las HAPN”, y que durante estos casi cinco años has sido cómplice, paciente, apoyo, escucha y aliento. Siempre con una sonrisa, y ayudándome con cada línea de cada demostración, de cada idea, de cada ejemplo. A Manolo, que has estado ahí siempre atento, con tu capacidad de abstracción y tu memoria infinitas. Con cuidado de que no nos metiéramos en terrenos demasiado ásperos, o que no tuvieran la suficiente relevancia. Siempre planificando y pensando en qué hacer después. Recuerdo un folio de *future work* encabezado con “Estibaliz” que me diste en verano de 2011, y del cual he estado comprendiendo algunos de los conceptos hace poco. Gracias a los dos por permitirme desarrollar la tesis en el GISED, que es un grupo con prestigio internacional en la investigación de las Redes de Petri gracias a vosotros.

A Cristian Mahulea, que has estado siempre cercano, disponible, atento y la vez silencioso. A Levid, compañero de *Chapter book*, de viajes, de quedadas de tapas, de conversaciones sobre redes de Petri y sobre todo amigo; a Xu, my collegue of the Petri nets's Thesis during these years, we have shared supervisor, topic, and trips. A Renato, de quien siempre he admirado tu completa e interesante tesis doctoral, y aún hoy lo sigo haciendo. A José Manuel Colom, a José Merseguer, a Javier Campos.

A Laura Recalde, porque muchas de las ideas que aquí hemos desarrollado las tenías tú ya en mente; y porque la alegría e ilusión que dejaste a tu alrededor llegó hasta mí a través de los demás.

Gracias a mis compañeros del laboratorio L1.01. Habéis sido mucho más que unos compañeros con quienes compartir un café cada mañana. Primero a los que ya pasasteis, gracias a Carlos, a Javi, a Dorian, a Diego, a Rosario, a Nacho, a María José. Me hicisteis sentirme como en casa desde el primer día, a través de cafés, quedadas, conversaciones sobre la investigación, sobre el mundo, sobre la vida. Y por supuesto gracias a quienes aún estáis disfrutando de la etapa del doctorado, gracias a Raúl, a Sara, a Alejandro, a Javier, a Jorge (aunque seamos contemporáneos), a Dani (aunque seas del laboratorio aquél). Mucha suerte en el resto de esta aventura. Disfrutad de estos años de poder pegaros horas y horas con una demostración o experimento, de no tener horarios, de poder ser flexibles. No dejéis de tomar cafés mañaneros, y seguid aprovechando los mediodías para conversar, reír y dibujar en la pizarra. ¡Cuidad del cactus!. También a muchos otros doctorandos y visitantes del departamento: Yasir, Henry, Alejo, Marilú, Ricardo, Jesús, Alejandro, César, Marta, Víctor, José Manuel, Fernando, Edgar, Alejandro... Con todos recuerdo una conversación, unas tapas, un comentario de complicidad, unas risas... ¡Por el DIIS pasáis gente muy especial!

Thank you Serge Haddad for the research stay that I did in the LSV in ENS Cachan. The 3-4 months that I spent there were a great opportunity to know a very good research institution from France. I remember specially the hours spent on your office, probing properties and finding efficient procedures on your board. I think we worked hard, and I am also glad of the scientific results that we achieve. Thank you also to the great people that I met there: Aina, César, Hernán, Christoph, Mahsa, Aiswairia, Martín... I found very good friends in Cachan.

Je voudrais aussi remercier à Dimitri Lefebvre, qui m'a ouvert les portes du Havre, pour le temps et l'énergie qu'il ma consacré. Les chercheurs du GREAH s'occupant aussi d'aspects plus appliqués que ceux de notre laboratoire, mon séjour m'a permis de m'approcher à la fluidité des réseaux de Petri d'un autre point de vue. Cette période à l'Université du Havre a été aussi très intense du côté humain. Un grand bonjour à Marwa Taleb, Edouard Lecrerq, Bruno, Eirini, Santana et Mario.

I do not want to forget to other colleagues which are also "freaks of the Petri nets", with which I have shared inspiration through research conversations in several events: Anna-Lena Meyer, Enrique Aguayo, Berenice Gudiño, Manuel Navarro, José Luis Malacara, Carla Seatzu, Maria Paola Cabasino, Alessandro Giua, Beniamino Guida.

Quiero dar las gracias a mi familia, Blanca, Elisardo y David. De vosotros he aprendido a ser optimista, a mirar el lado bueno de las cosas, a adaptarme a los cambios, a tener curiosidad y ganas de aprender, y a combinar los planes a corto plazo con los objetivos a largo plazo. Quiero recordar a mi tío Javier, y a mi yaya Esther, que también fallecieron durante el desarrollo de esta tesis.

También quiero dar las gracias al resto de mis amigos, primero por animarme a empezar el doctorado, y después por animarme y comprenderme. Gracias a Adrián, Alberto, Almudena, Álvaro, Anna, Bernar, Cristina, David, Emma, Erika, Chaby, Chema, Fernando, Ismael, Javi, Luismi, María, Maribel, Marieta, Mónica, Myriam, Nacho, Noemí, Óscar, Pablo, Raúl, Rubén, Sandra, Saúl, Silvia, Vanesa ...

Finalmente gracias (¿por qué no?) a las instituciones que me han permitido embarcarme en este viaje, en el que he podido ganar mi primer salario, ahorrar, emanciparme, hacer dos inspiradoras estancias de investigación, y viajar a cuatro congresos (incluída una increible semana con nuestros colegas de las RdP en Guadalajara, México) y una *summer school*. Gracias principalmente a la beca predoctoral B174/11 del Gobierno de Aragón. Gracias también al Proyecto Europeo "*DISC, Distributed Supervisory Control of Complex Plants*", al proyecto CICYT-FEDER DPI2010-20413 del Gobierno de España, a la subvención IT/27 del Gobierno de Aragón al grupo de investigación GISED, y al Fondo Social Europeo.

Resumen

Las Redes de Petri (RdP) son un formalismo ampliamente aceptado para el modelado y análisis de Sistemas de Eventos Discretos (SED). Por ejemplo sistemas de manufactura, de logística, de tráfico, redes informáticas, servicios web, redes de comunicación, procesos bioquímicos, etc.

Como otros formalismos, las redes de Petri sufren del problema de la "explosión de estados", en el cual el número de estados crece explosivamente respecto de la carga del sistema, haciendo intratables algunas técnicas de análisis basadas en la enumeración de estados. La fluidificación de las redes de Petri trata de superar este problema, pasando de las RdP discretas (en las que los disparos de las transiciones y los marcados de los lugares son cantidades enteras no negativas) a las RdP continuas (en las que los disparos de las transiciones, y por lo tanto los marcados se definen en los reales).

Las RdP continuas disponen de técnicas de análisis más eficientes que las discretas. Sin embargo, como toda relajación, la fluidificación supone el detrimento de la fidelidad, dando lugar a la pérdida de propiedades cualitativas o cuantitativas de la red de Petri original.

El objetivo principal de esta tesis es mejorar el proceso de fluidificación de las RdP, obteniendo un formalismo continuo (o al menos parcialmente) que evite el problema de la explosión de estados, mientras aproxime adecuadamente la RdP discreta. Además, esta tesis considera no solo el proceso de fluidificación sino también el formalismo de las RdP continuas en sí mismo, estudiando la complejidad computacional de comprobar algunas propiedades.

En primer lugar, se establecen las diferencias que aparecen entre las RdP discretas y continuas, y se proponen algunas transformaciones sobre la red discreta que mejorarán la red continua resultante.

En segundo lugar, se examina el proceso de fluidificación de las RdP autónomas (i.e., sin ninguna interpretación temporal), y se establecen ciertas condiciones bajo las cuales la RdP continua preserva determinadas propiedades cualitativas de la RdP discreta: limitación, ausencia de bloqueos, vivacidad, etc.

En tercer lugar, se contribuye al estudio de la decidibilidad y la complejidad computacional de algunas propiedades comunes de la RdP continua autónoma.

En cuarto lugar, se considera el proceso de fluidificación de las RdP temporizadas. Se proponen algunas técnicas para preservar ciertas propiedades cuantitativas de las RdP discretas estocásticas por las RdP continuas temporizadas.

Por último, se propone un nuevo formalismo, en el cual el disparo de las transiciones se adapta a la carga del sistema, combinando disparos discretos y continuos, dando lugar a las Redes de Petri híbridas adaptativas. Las RdP híbridas adaptativas suponen un marco conceptual para la fluidificación parcial o total de las Redes de Petri, que engloba a las redes de Petri discretas, continuas e híbridas. En general, permite preservar propiedades de la RdP original, evitando el problema de la explosión de estados.

Contents

1 Introduction

In science and engineering disciplines, a *model* is an abstract representation of reality, in which some details have been omitted and only the relevant characteristics (with respect to an objective) have been preserved. *Formal models* provide mathematical and unambiguous methods and techniques for the modelling, design, analysis and verification of a lot of kinds of systems. A *formalism* states how a certain formal model can be built.

A family of formalisms can be considered as a *modelling paradigm*, a conceptual framework that allows to obtain several instances from some common concepts and principles. A given system can be modelled by distinct formalisms from the same modelling paradigm, obtaining some advantages such as coherence and economy.

The discrete state systems whose evolution is completely determined by the occurrence of some discrete events are denoted as *Discrete Event Systems* (DES). They appear in many fields, for instance in manufacturing, logistics, computer networks (web services, communication networks...), traffic systems, population dynamics, biochemical processes, etc. The discipline devoted to the modelling, analysis and verification of those systems is also denoted as DES. The research on DES has been approached from several disciplines, such as systems theory, computer science and operational research. The modelling and analysis of DES is considered in this thesis.

Petri Nets (PN) are one of the most widely used formalisms for DES. They constitute the modelling paradigm considered in this work, having powerful analysis and synthesis techniques and a direct graphical representation. PN consist of places (drawn as circles) which represent the state, and transitions (drawn as rectangles or bars) which represent the events among states. In contrast to other formalisms, the state of a system is a collection of *local* states, and it is represented with tokens contained in the places. The PN formalism has been especially used to model concurrent and dis tributed systems in which *concurrency* and *synchronization* relations appear [Pet81; Bra83; Sil85; DA10].

Different Petri net formalisms can be used for the diverse phases of the life-cycle of a system. For instance, the basic untimed model can be used to study basic relationships with non determinism, and a timed model can be used for performance evaluation.

Fluidization of Petri nets

Petri nets are a powerful formalism for the modelling and analysis of DES. However, some analysis techniques are inefficient for the analysis of highly populated systems. As other formalisms for DES, the analysis of PN suffers from the *state explosion problem*: the set of reachable states grows exponentially with respect to the initial population of the system, and hence the analysis techniques based on the exploration of the state space become intractable for high workloads.

A classical technique to overcome this difficulty is known as *fluidization*, which proposes to relax the integrality constraints of the original discrete model, and deals with a continuous (or hybrid) approximation. Such a relaxation aims at computationally more efficient analysis methods, at the price of losing some precision. It is the kind of relaxation which appears when a *Integer Programming Problem* (IPP) is relaxed to a *Linear Programming Problem* (LPP), which has more efficient analysis techniques.

In the field of PN, total fluidization of classical *Discrete Petri Nets* (DPN) results in *Continuous Petri Nets* (CPN). The idea of the fluidization of Petri nets was proposed at the net level (see [DA10] for a comprehensive view) in 1987 in the field of manufacturing systems by David and Alla in [DA87]. Developed in parallel, the fluidization at the level of the *state equation* was proposed at the same meeting (the 8th European Workshop on Application and Theory of PN, Zaragoza) by Silva and Colom (see [STC98]), focusing on the use of linear programming techniques to analyse the net systems.

Continuous PN are obtained by removing the integrality constraint of the firing of transitions, which means that the firing count vector and consequently the marking are no longer restricted to be in the naturals, but relaxed into non-negative real numbers [DA10; Sil+11].

If no time interpretation is chosen, untimed (or autonomous) continuous Petri nets are obtained. Moreover, as in discrete PN, distinct time interpretations can be considered. The most common approaches consider time associated to transitions, dealing to different firing semantics. *Timed Continuous PN* (TCPN) under *infinite server semantics* are one of the most used and accepted time interpretations [MRS09; Sil+11].

Continuous PN can be seen as a relaxation of a discrete PN system. Moreover, continuous PN have been also used to directly model and analyse systems in different application fields. In [Cab+10], a method based on CPN is proposed for the fault diagnosis of manufacturing systems that manage systems intractable with discrete Petri nets (for modelling of manufacturing systems see also [ZA90]). In [RL+10], the authors introduce a bottom-up modelling methodology based on CPN to represent a cell metabolism and solve a regulation control problem.

When not every transition of a discrete PN system is fluidified, *Hybrid Petri Nets* (HPN) [BGS01; DA10] are obtained, in which some transitions are continuous and the rest of them are discrete. Applications of HPN include the modelling of manufacturing systems [BGM00], the simulation of water distribution systems [GMLMA12] or the analysis of traffic in urban networks [Váz+10], among others.

Limitations of continuous PN

At first glance, the simple way in which the basic definitions of discrete models are extended to continuous ones may make us naively think that their behaviour will be similar. However, the behaviour of the continuous model can be completely different just because the integrality constraint has been dropped. In other words, not all DEDs can be satisfactorily fluidified.

In the context of PN, the relaxation from discrete PN to continuous PN obtains more tractable analysis techniques, at the price of loosing some fidelity to the discrete system, such as losing some system capabilities, e.g., mutual exclusion [DA10; Sil+11].

Considering untimed PN, some qualitative properties of the discrete PN may be lost by fluidization. In continuous PN, a transition can be fired in real amounts. Hence, some markings (or states) which were not reachable by the original discrete system become reachable by the continuous system. Because of these new firings that are possible and markings which become reachable, important properties of the original discrete model such as deadlock-freeness, liveness, reversibility, etc. are not preserved, in general, by the fluidization of untimed PN [RTS99].

When a time interpretation is considered (i.e., timed PN), some quantitative properties may also be lost by the fluidified system. These discrepancies between discrete *Stochastic Petri Nets* (SPN) and TCPN have been studied in the literature, and some techniques have been proposed to improve the accuracy of the original fluid approximation such as [LL12; VS12]. However, more research is needed to establish the conditions under which the fluid approximation is not good, and to propose approaches to improve such fluid approximation.

Objectives of the thesis

The general goal of this thesis is to contribute to the study and improvement of the fluidization of discrete PN, in order to overcome the limitations of continuous PN.

Considering both untimed and timed PN, one of the objectives of this thesis is the study of the fluidization processes, i.e., to establish qualitative and quantitative properties which are preserved by fluidization and under which conditions.

Another important objective is to improve the fluidization process, i.e., to reduces the differences between a discrete PN and its continuous counterpart. This improvement is specially important when “relatively” small populations are considered, neither very small ones (for which fluidization would not be needed) nor very large ones (in which fluidization gives reasonably good results). Some alternative fluidization techniques are proposed for this aim. They will offer a trade-off between the fidelity to the original discrete PN, and the avoidance of the state explosion problem. The obtained improvement will be bigger (in relative terms) when the marking of the system is small. The possible techniques include: modification of the structure of the discrete net before its fluidization, definition of new semantics or structures for the firing of transitions, and definition of a new formalism which proposes the partial fluidization of transitions. Partial fluidization is considered in the formalism of *Hybrid Adaptive Petri Nets* (HAPN), which proposes a partially hybrid PN in which transitions can be discrete or continuous depending on its workload.

Finally, it is also interesting to study not only the fluidization processes, but also the continuous PN formalism itself. Research on complexity issues provides a better understanding of the formalism. The objective is to improve the known lower and upper bounds of checking some properties in CPN [RTS99; JRS03; RHS10], and to show that a property (such as reachability set inclusion) which is undecidable for DPN can become decidable for CPN.

Contributions and publications

The main contributions obtained by this thesis for the analysis and improvement of the fluidization of PN are enumerated below:

- *Improvement of the original discrete system.* Some preliminary transformations of the discrete PN at the structure level to improve the fluidified system are suggested. The proposed techniques can be found in [FJS14a], and in a paper submitted to an indexed journal [FJS s].

- *Study of the fluidization of untimed PN.* The thesis establishes the conditions that a discrete PN has to fulfil to preserve some qualitative properties (such as boundedness, deadlock-freeness, liveness, etc) when it is fluidified. This work has been presented in a conference paper [FJS12], among the finalists for the "best student paper award", and an extended version has been published in a journal [FJS14a].

- *Determination of the computational complexity of checking some properties in untimed PN.* The upper bounds for some decision procedures have been improved, and novel lower bounds have been determined. These results have been published in [FH13], which obtained the "outstanding paper award" recognition, and in the journal paper [FH15].

- *Preservation of quantitative properties by the fluidization of discrete stochastic PN.* The conditions under which the throughput approximation is not good have been studied, and the fluidization process has been improved by means of a new semantics for the firing of transitions. Most of these results are included in the conference paper [FJS14b], also finalist for the "best student paper award". A generalization of the work has been submitted to an indexed journal [FJS s].

- *Definition and study of a conceptual framework for the partial fluidization of untimed PN, hybrid adaptive PN.* It is a generalization of discrete, continuous and hybrid PN, and it also allows adaptation to the workload of the system. An algorithm to characterize the reachability set of a net system in this formalism has been designed. And the preservation of certain properties of the discrete system has been studied. The definition and some preliminary properties preservation results were published in the conference paper [Fra+11]. And the computation of its reachability graph and reachability set are included in the journal paper [FJS15].

Document organization

The document is organized in four parts. The first of them presents some definitions and previous concepts of discrete PN and it proposes some basic transformations over the original discrete system to improve the fluidified one. Part II studies the fluidization process of untimed PN: preservation of untimed properties of the discrete system, and complexity analysis of some properties in continuous PN. The fluidization of timed PN is considered in Part III, which focuses on the approximation of the throughput of discrete systems. Finally, the formalism of HAPN and some techniques for its reachability analysis are proposed in Part IV. The objectives of each part and each chapter are described below:

- **Part I. Previous concepts and transformations of the discrete system**
  In this part, the discrete PN formalism is considered before its fluidization. First, some concepts and definitions are provided. Then, some transformations of the original discrete system are proposed. These transformations do not modify the behaviour of the discrete PN system, but they can improve the fluidization process.

  - **Chapter 2. Previous concepts on discrete Petri net systems**
    This chapter presents the formal definitions and concepts related to discrete Petri nets. Untimed PN are defined first, as well as some reachability concepts and properties. Then, a time interpretation is considered, dealing with discrete stochastic Petri nets.
  - **Chapter 3. Previous transformations of the original discrete system**
    An interesting technique to improve the fluidization is to apply some previous transformations to the original PN system, preserving its discrete behaviour. When the transformed PN is fluidified, its behaviour is more faithful to the one of the original discrete PN. These previous transformations performed on the discrete PN system are described in this chapter.

- **Part II. On the fluidization of untimed Petri nets**
  This part focuses on the fluidization of untimed PN. The preservation of homothetic properties by the continuous PN is established. Moreover, the computational complexity of the CPN formalism itself is examined.

  - **Chapter 4. Previous concepts on the fluidization of untimed Petri nets**
    This chapter presents the fluidization of untimed Petri nets. Some basic concepts about continuous Petri nets are included in this chapter, such as its definition, its reachability analysis and certain system properties.

- **Chapter 5. Preservation of homothetic properties by continuous Petri nets**
  This chapter studies the preservation of untimed properties such as boundedness, deadlock-freeness, liveness, etc. when a DPN system is fluidified. It states a sufficient condition (homothetic monotonicity) for property preservation. It also examines the preservation of deadlock-freeness in different net subclasses.
- **Chapter 6. Complexity analysis of continuous Petri net properties**
  In contrast with Chapter 4, in which the fluidization process is considered, this chapter considers the continuous PN formalism itself. In fact, it continues with the complexity analysis of deciding the considered properties (boundedness, deadlock-freeness, etc) in CPN.

○ **Part III. On the fluidization of timed Petri nets**
When a time interpretation is added to the CPN, *Timed Continuous PN* (TCPN) are obtained. The approximation of quantitative properties from discrete systems such as steady state throughput is investigated.

- **Chapter 7. Previous concepts on the fluidization of timed Petri nets**
  This chapter presents some formal definitions and concepts about fluidization of timed Petri nets. It presents the addition of a time interpretation to the untimed CPN. The throughput approximation of discrete stochastic Petri nets is considered.
- **Chapter 8. The "bound reaching problem"**
  In this chapter, the bound reaching problem is identified, a particular situation in which the differences between the discrete and the continuous behaviour are particularly acute. Moreover, a new semantics to fluidify the PN systems which suffer from such *bound reaching problem* is derived from *infinite servers* semantics.

○ **Part IV. Hybrid adaptive Petri nets**
In this part, an alternative fluidization formalism is proposed, the *Hybrid Adaptive Petri Nets* (HAPN). It is a general formalism that includes discrete, continuous and hybrid PN, and it allows a partial fluidization of transitions. Each transition has an associated threshold, such that it behaves as continuous if its enabling degree is above the threshold, and otherwise it behaves as discrete.

- **Chapter 9. Hybrid adaptive Petri nets: definition and deadlock-freeness preservation**
  This first chapter dedicated to HAPN presents the definition of the untimed formalism. Moreover, reachability set inclusion and preservation of deadlock- freeness of DPN systems by HAPN is studied.
- **Chapter 10. Reachability analysis of Hybrid Adaptive Petri nets**
  This chapter characterizes the reachability property of HAPN. A method to obtain

the reachability graph and reachability space of HAPN is proposed. Due to the fact that HAPN includes HPN, the method can be also applied to HPN.

- **Chapter 11. Conclusions and perspectives**
  Finally, Chapter 11 summarizes the conclusions of the thesis and identifies some research lines which still remain open.

Part I

Previous concepts and transformations of the discrete system

2

Previous concepts on discrete Petri net systems

"*We [the Moderns] are like dwarves perched on the shoulders of giants [the Ancients], and thus we are able to see more and farther than the latter*".
Bernard of Chartres

This chapter presents the basic concepts related to the discrete Petri Nets (PN) considered in this thesis. First, untimed models are introduced. The formal definition is presented first, then basic reachability concepts are introduced. Some properties of systems which are used in the rest of the document are also defined: synchronic properties such as boundedness, behavioural properties such as *deadlock-freeness*, *liveness*, *home state* and *reversibility*.

Moreover, *Linear Enabling Functions*, which were proposed to express the enabling of a transition with a single linear function, are defined. The analysis of (homothetic) deadlock-freeness by using them is proposed, as well as its implementation by means of the *representative* places.

PN with time are also introduced. Among the different time interpretations of discrete PN, stochastic PN have been considered. The chapter starts which some basic notations which will be used along the thesis.

Notations

$\mathbb{N}$ (resp. $\mathbb{Q}$, $\mathbb{R}$) is the set of non negative integers (resp. rational, real numbers). Given a set of numbers E, $E_{\geq 0}$ (resp. $E_{>0}$) denotes the subset of non negative (resp. positive) numbers of E. Given an $E \times F$ matrix $\boldsymbol{A}$ with E and F sets of indices, $E' \subseteq E$ and $F' \subseteq F$, the $E' \times F'$ submatrix $\boldsymbol{A}[E', F']$ denotes the restriction of $\boldsymbol{A}$ to rows indexed by E' and columns indexed by F'. The support of a vector $\boldsymbol{v} \in \mathbb{R}^E$, denoted $\|\boldsymbol{v}\|$, is defined by $\|\boldsymbol{v}\| \stackrel{\text{def}}{=} \{e \in E \mid \boldsymbol{v}[e] \neq 0\}$. $\boldsymbol{0}$ denotes the null vector. It is written $\boldsymbol{v} \geq \boldsymbol{w}$ when $\boldsymbol{v}$ is componentwise greater or equal than $\boldsymbol{w}$ and $\boldsymbol{v} \gneq \boldsymbol{w}$ when $\boldsymbol{v} \geq \boldsymbol{w}$ and $\boldsymbol{v} \neq \boldsymbol{w}$. It is written $\boldsymbol{v} > \boldsymbol{w}$ when $\boldsymbol{v}$ is componentwise strictly greater than $\boldsymbol{w}$.

2.1 Untimed Petri nets

Some concepts on untimed discrete PN are presented in this section. First, the concepts related to the PN structure and discrete PN systems are defined. Then, some reachability concepts related to the formalism are presented. Finally, classical properties are defined for discrete PN. Some of the considered properties are synchronic properties, which include boundedness and B-fairness among others; and other behavioural properties such as deadlock-freeness, liveness or reversibility. It is assumed that the reader is familiar with Petri nets (see [Mur89; DiC+93] for a gentle introduction).

2.1.1 Petri net systems

The Petri net structure is denoted as $\mathcal{N}$:

Definition 2.1 *A PN is a tuple $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}\rangle$ where $P = \{p_1, p_2, ..., p_n\}$ and $T = \{t_1, t_2, ..., t_m\}$ are disjoint and finite sets of places and transitions, and $\boldsymbol{Pre}$, $\boldsymbol{Post}$ are $|P| \times |T|$ sized, natural valued, incidence matrices.*

Given a node $v \in P \cup T$, its *preset*, ${}^\bullet v$, is defined as the set of its input nodes. Its *postset*, $v^\bullet$, is defined as the set of its output nodes. These definitions can be naturally extended to sets of nodes.

The *reverse* net of $\mathcal{N}$ is defined as $\mathcal{N}^{-1} = \langle P, T, \boldsymbol{Post}, \boldsymbol{Pre}\rangle$, in which places and transitions coincide, and arcs are inverted.

Given a Petri net and a marking, the discrete Petri net system is defined:

Definition 2.2 *A discrete PN system is a tuple $\langle \mathcal{N}, \boldsymbol{m}_0\rangle_D$ where $\mathcal{N}$ is the structure and $\boldsymbol{m}_0 \in \mathbb{N}^{|P|}$ is the initial marking.*

In discrete PN systems, a transition t is *enabled* at $\boldsymbol{m}$ if for every $p \in {}^\bullet t$, $m[p] \geq Pre[p, t]$. An enabled transition t can be fired in any amount $\alpha \in \mathbb{N}$ such that $0 < \alpha \leq enab(t, \boldsymbol{m})$, where the discrete enabling degree is defined as:

$$enab(t, \boldsymbol{m}) = \min_{p \in {}^\bullet t} \left\lfloor \frac{m[p]}{Pre[p, t]} \right\rfloor \tag{2.1}$$

The firing of t in an amount equal to 1 leads to a new marking $\boldsymbol{m}'$, and it is denoted as $\boldsymbol{m}\xrightarrow{t}\boldsymbol{m}'$. It holds $\boldsymbol{m}' = \boldsymbol{m} + \boldsymbol{C}[P,t]$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the token flow matrix (incidence matrix if $\mathcal{N}$ is self-loop free) and $\boldsymbol{C}[P,t]$ denotes the column t of the matrix $\boldsymbol{C}$. The state (or fundamental) equation, $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C}\cdot\boldsymbol{\sigma}$, summarizes the way the marking evolves; where $\boldsymbol{\sigma}$ is the firing count vector associated to the fired sequence σ.

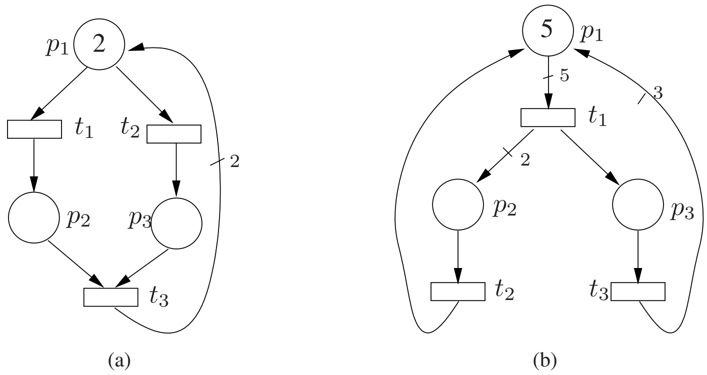


Figure 2.1: Two very basic PN systems.

Right ($\boldsymbol{C}\cdot\boldsymbol{x} = \boldsymbol{0}$) and left ($\boldsymbol{y}^T\cdot\boldsymbol{C} = \boldsymbol{0}$) natural annullers of the token flow matrix are called T- and P-*semiflows*, respectively. A semiflow is *minimal* when its support is not a proper superset of the support of any other semiflow, and the greatest common divisor of its elements is one. If $\exists\boldsymbol{y} > \boldsymbol{0}$ s.t. $\boldsymbol{y}^T\cdot\boldsymbol{C} = \boldsymbol{0}$, then the net is *conservative*, and if $\exists\boldsymbol{x} > \boldsymbol{0}$ s.t. $\boldsymbol{C}\cdot\boldsymbol{x} = \boldsymbol{0}$, then the net is *consistent*.

Three different concepts are defined by the P-*semiflows*:

Definition 2.3

- *The P-semiflow (a vector). The vector $\boldsymbol{y}$ which holds $\boldsymbol{y}^T\cdot\boldsymbol{C} = \boldsymbol{0}$.*
- *The token conservation law or marking invariant (an equation). If $\exists\boldsymbol{y}\gneq\boldsymbol{0}$ then, by the state equation, it holds that given an arbitrary $\boldsymbol{m}_0$, $\boldsymbol{y}^T\cdot\boldsymbol{m}_0 = \boldsymbol{y}^T\cdot\boldsymbol{m}$ for every reachable marking $\boldsymbol{m}$.*
- *The conservative component (a net). The P-subnet generated by the support of a P-semiflow is called conservative component, meaning that it is a part of the net that conserves its weighted token content.*

For example, the PN in Fig. 2.1(a) has a P-semiflow equal to $\boldsymbol{y} = (1,1,1)$. Given $\boldsymbol{m}_0 = (2,0,0)$, it holds that $m[p_1]+m[p_2]+m[p_3] = 2$ for every reachable marking. Moreover, the PN in Fig. 2.1(b) has a P-semiflow equal to $\boldsymbol{y} = (1,1,3)$, i.e., $m[p_1]+m[p_2]+3\cdot m[p_3] = 5$ (for the initial marking $\boldsymbol{m}_0 = (5,0,0)$). Both nets are conservative.

Dually, the T-*semiflows* also define three different concepts:

Definition 2.4

- *The T-semiflow (a vector). The vector* $\boldsymbol{x}$ *which holds* $\boldsymbol{C} \cdot \boldsymbol{x} = 0$.
- *T-semiflows are firing count vectors that identify potentially cyclic behaviours in the system. If* $\exists \boldsymbol{x} \gneq \boldsymbol{0}$ *s.t.* $\boldsymbol{x}$ *is a T-semiflow and* $\boldsymbol{x}$ *is fireable from* $\boldsymbol{m}$ *then, by the state equation,* $\boldsymbol{m} \xrightarrow{\sigma} \boldsymbol{m}$ *with* σ *being a firing sequence whose firing count vector is equal to* $\boldsymbol{x}$.
- *The consistent component (a net). The T-subnet generated by the support of a T-semiflow.*

For example, the PN in Fig. 2.1(a) has a T-semiflow equal to $\boldsymbol{x} = (1, 1, 1)$, which identifies the cyclic behaviour $t_1 t_2 t_3$ which returns the marking to the initial state. Moreover, the PN in Fig. 2.1(b) has a T-semiflow equal to $\boldsymbol{x} = (1, 2, 1)$, i.e., the initial marking is recovered when $t_1 t_2 t_2 t_3$ are fired. Both nets are consistent.

A set of places Θ is a *trap* if: (a) $\Theta^\bullet \subseteq {}^\bullet\Theta$; and (b) for each place $p \in \Theta$, the firing of any $t \in {}^\bullet p$ enables at least one $t \in p^\bullet$. Condition (b) is always true in ordinary PN. A marked trap will always remain marked in discrete systems. Analogously, a set of places Σ is a *siphon* if ${}^\bullet\Sigma \subseteq \Sigma^\bullet$. An empty siphon will always remain empty.

2.1.2 Reachability and implicit places

The set of all the reachable markings of a discrete system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is denoted as reachability set, $RS_D(\mathcal{N}, \boldsymbol{m}_0)$.

Definition 2.5 $RS_D(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} \mid \exists \sigma = t_{\gamma_1} \dots t_{\gamma_k} \; such\ that\ \boldsymbol{m}_0 \xrightarrow{t_{\gamma_1}} \boldsymbol{m}_1 \xrightarrow{t_{\gamma_2}} \boldsymbol{m}_2 \cdots \xrightarrow{t_{\gamma_k}} \boldsymbol{m}_k = \boldsymbol{m}\}$.

We introduce below the *linearized reachability set*, defined as the set of all vectors $\boldsymbol{m}$ for which a $\boldsymbol{\sigma}$ exists s.t. the state equation is fulfilled. It is defined both in the natural and in the real numbers.

Definition 2.6

- $LRS_D(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} \in \mathbb{N}^{|P|} \mid \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma},\ where\ \boldsymbol{m} \in \mathbb{N}^{|P|}, \boldsymbol{\sigma} \in \mathbb{N}^{|T|}\}$.
- $LRS_C(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} \in \mathbb{R}^{|P|} \mid \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma},\ where\ \boldsymbol{m} \in \mathbb{R}^{|P|}, \boldsymbol{\sigma} \in \mathbb{R}^{|T|}\}$.

A *spurious* marking $\boldsymbol{m}_s \in \mathbb{R}^{|P|}$ of a discrete system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is a solution of the state equation, i.e., $\boldsymbol{m}_s \in LRS_C(\mathcal{N}, \boldsymbol{m}_0)$ which is not reachable in the discrete system, i.e., $\boldsymbol{m}_s \notin RS_D(\mathcal{N}, \boldsymbol{m}_0)$.

Given two systems with the same set of places, the reachability set inclusion property can be defined.

Definition 2.7 ***Reachability set inclusion***
Given systems $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ and $\langle \mathcal{N}', \boldsymbol{m}'_0 \rangle_D$ with $P = P'$, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is reachable included in $\langle \mathcal{N}', \boldsymbol{m}'_0 \rangle_D$ if $RS_D(\mathcal{N}, \boldsymbol{m}_0) \subseteq RS_D(\mathcal{N}', \boldsymbol{m}'_0)$.

A place p is said to be an *implicit place* if it does not constrain the behaviour of the discrete system.

Definition 2.8 *Given a PN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$:*

- *A place p is* implicit *in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ if it is never the unique place that prevents the firing of a transition.*
- *A place p is* structural implicit *in $\mathcal{N}$ if there exists $\boldsymbol{m}_0$ for which p is implicit.*

A characterization of the structural implicit places is given by [STC98]:

Proposition 2.9 *Let $\mathcal{N} = \langle P \cup \{p\}, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle_D$. Place p is structurally implicit if:*

1. *A $\boldsymbol{y} \geq \boldsymbol{0}$ exists such that $\boldsymbol{C}[p,T] \geq \boldsymbol{y}^T \cdot \boldsymbol{C}[P,T]$*
2. *No $\boldsymbol{x} \geq \boldsymbol{0}$ exists such that $\boldsymbol{C}[P,T] \cdot \boldsymbol{x} \geq \boldsymbol{0}$ and $\boldsymbol{C}[p,T] \cdot \boldsymbol{x} < \boldsymbol{0}$*

A sufficient condition for place p to be implicit in $\langle \mathcal{N}, \boldsymbol{m_0} \rangle_D$ is that $m_0[p] \geq z$, with z defined as follows:

$$
\begin{array}{rll}
z = & min & \boldsymbol{y} \cdot \boldsymbol{m}_0 \\
& s.t. & \boldsymbol{y}^T \cdot \boldsymbol{C}(P,T) \geq \boldsymbol{C}(p,T)) \\
& & \boldsymbol{y}^T \cdot \boldsymbol{Pre}(P, p^\bullet) \geq \boldsymbol{Pre}(p, p^\bullet) \\
& & \boldsymbol{y} \geq 0
\end{array}
\tag{2.2}
$$

The minimal initial marking from which a structural implicit place p becomes implicit can be efficiently computed from the initial markings of the rest of the places.

Proposition 2.10 *[STC98] Let $\mathcal{N} = \langle P \cup \{p\}, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle_D$. Place p is implicit if $m_0[p]$ is greater than or equal to the optimal value of the following* Linear Programming Problem *(LPP):*

$$
\begin{array}{ll}
min & \boldsymbol{y}^T \cdot \boldsymbol{m}_0[P] + \nu \\
s.t. & \boldsymbol{y}^T \cdot \boldsymbol{C}[P,T] \leq \boldsymbol{C}[p,T] \\
& \boldsymbol{y}^T \cdot \boldsymbol{Pre}[P, p^\bullet] + \nu \cdot \boldsymbol{1} \geq \boldsymbol{Pre}[p, p^\bullet] \\
& \boldsymbol{y} \geq \boldsymbol{0}
\end{array}
\tag{2.3}
$$

We can also consider the *concurrent implicit* places, which preserve not only the transition firing sequences, but also the steps [STC98; GVC99]. Its characterization is given by the following proposition.

Proposition 2.11 *[GVC99] Let $\mathcal{N} = \langle P \cup \{p\}, T, \boldsymbol{Pre}, \boldsymbol{Post}\rangle_D$. Assuming there exists a marking solution of the state equation $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C}[P,T] \cdot \boldsymbol{\sigma}$ enabling at least one transition $t \in p^\bullet$, then p is a* concurrent implicit *place if $\exists \boldsymbol{y}, \boldsymbol{z}, \nu$ such that:*

$$\begin{aligned}
&\boldsymbol{y}^T \cdot \boldsymbol{C}[P,T] \leq \boldsymbol{C}[p,T]\\
&\boldsymbol{z}^T \cdot \boldsymbol{Pre}[P,p^\bullet] + \nu \cdot \mathbf{1}^T \geq \boldsymbol{Pre}[p,p^\bullet]\\
&\boldsymbol{y}^T \cdot \boldsymbol{m}_0 + \nu < m_0[p] + 1\\
&\boldsymbol{y} \geq \boldsymbol{z} \geq \mathbf{0}, \nu \leq 0
\end{aligned}$$

2.1.3 System properties

Some common properties of PN systems are presented in this section for discrete PN. They are divided in two general groups: synchronic properties, and deadlock-freeness and liveness properties.

Synchronic properties

Synchrony theory [Sil87; SC88] is a branch of the Net Theory that studies transition firing dependences (synchronic properties). Considering PN systems, these dependences are qualitatively characterized as *synchronic relations*.The *synchronic properties* describe some behavioural properties of PN systems in terms of *quantitative assertions*. Some of the synchronic properties are *lead*, *distance*, *places bounds*, *places mutual exclusion*, *B-fairness*, etc, which may be interesting for the analysis of manufacturing systems.

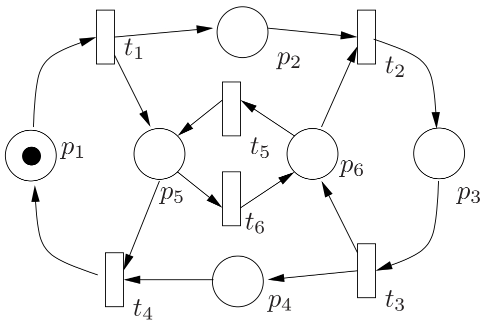


Figure 2.2: PN system to show the fireability of its T-semiflows: $\boldsymbol{x}_1 = (1,1,1,1,0,0)$, $\boldsymbol{x}_2 = (0,0,0,0,1,1)$. $\boldsymbol{x}_2$ can be fired in isolation but $\boldsymbol{x}_1$ can not be fired without firing $\boldsymbol{x}_2$.

Let us illustrate the fireability of T-semiflows by means of a simple example. Consider the PN system in Fig. 2.2, with initial marking $\boldsymbol{m}_0 = (1,0,0,0,0,0)$. It has two T-semiflows: $\boldsymbol{x}_1 = (1,1,1,1,0,0)$ and $\boldsymbol{x}_2 = (0,0,0,0,1,1)$. If the PN system is considered as discrete, then $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ cannot be fired in isolation (i.e., any sequence $\sigma = [t_1(t_6t_5)^*t_6t_2t_3(t_5t_6)^*t_5t_4]^*$ is fireable).

A synchronic property that holds in a given PN system from a certain initial marking is said to be *behavioural*, while a synchronic property which holds for every initial marking is called *structural*.

Boundedness and B-fairness are considered in this work, as representative of synchronic properties.

Definition 2.12 ***Boundedness (B).***

- *A place p is bounded if $\exists b \in \mathbb{R}_{>0}$ such that for all $\boldsymbol{m} \in RS_D(\mathcal{N}, \boldsymbol{m}_0)$, $m[p] \leq b$.*
- *A system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is bounded if every $p \in P$ is bounded.*

The structural bound of a place p, $SB(p)$, can be computed as $SB(p) = max\{m[p]| \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \boldsymbol{m}, \boldsymbol{\sigma} \geq 0\}$.

Definition 2.13 ***B-Fairness (BF).***

- *Two transitions t, t' are in B-fair relation if $\exists b \in \mathbb{R}_{>0}$ such that for all $\boldsymbol{m} \in RS_D(\mathcal{N}, \boldsymbol{m}_0)$, for every finite (infinite) firing sequence σ fireable from $\boldsymbol{m}$, it holds that if $\boldsymbol{\sigma}(t) = 0$ then $\boldsymbol{\sigma}(t') \leq b$; and if $\boldsymbol{\sigma}(t') = 0$ then $\boldsymbol{\sigma}(t) \leq b$.*
- *A system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is B-fair if every pair of transitions $t, t' \in T$ is in B-fair relation.*

A net $\mathcal{N}$ is *structurally* bounded (resp. *structurally* B-fair) if $\forall \boldsymbol{m}_0$, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is bounded (resp. B-fair).

The structural bound of a place p, and the structural enabling bound of a transition t are respectively defined as:

$$SB(p) = \lfloor max\{m[p] \mid \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \boldsymbol{m}, \boldsymbol{\sigma} \geq 0\} \rfloor$$

$$SEB(t) = \lfloor max\{e \mid \forall p \in {}^{\bullet}t, e \leq \frac{m[p]}{Pre[p,t]}, \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \boldsymbol{m}, \boldsymbol{\sigma} \geq 0\} \rfloor$$

Consider the PN system in Fig. 2.3. The structural bound of places p_3 and p_4 is $SB(p_3) = SB(p_4) = 2$. However, both places cannot have two tokens at the same marking, and the structural enabling bound of t_5 is $SEB(t_5) = 1$.

Deadlock-freeness and liveness properties

Some well known properties in discrete systems [Mur89; DiC+93], and often required for real systems, are recalled below.

Definition 2.14 ***Deadlock-Freeness (DF).*** *A system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is deadlock-free if $\forall \boldsymbol{m} \in RS_D(\mathcal{N}, \boldsymbol{m}_0)$, $\exists t \in T$ such that t is enabled at $\boldsymbol{m}$.*

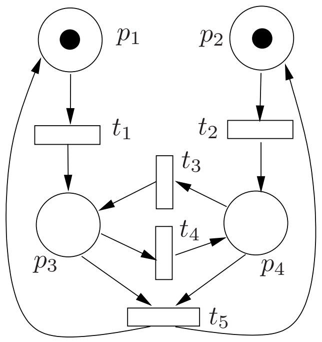


Figure 2.3: A PN net system. The structural enabling bound of t_5 is equal to 1.

For example, the PN system in Fig. 2.1(b) is deadlock-free, while the one in Fig. 2.1(a) is not deadlock-free, because it can reach deadlock marking $\boldsymbol{m}_d = (0, 2, 0)$.

Definition 2.15 ***Liveness (L)***. *A system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is live if for every transition t and for every marking $\boldsymbol{m} \in RS_D(\mathcal{N}, \boldsymbol{m}_0)$ there exists $\boldsymbol{m}' \in RS_D(\mathcal{N}, \boldsymbol{m}_0)$ such that t is enabled at $\boldsymbol{m}'$.*

A home state is a marking that can be reached whatever the current state. This property can express for instance that recovering from faults is always possible.

Definition 2.16 ***Home state***. *A marking $\boldsymbol{m}_h$ is a home state if $\forall \boldsymbol{m}' \in RS_D(\mathcal{N}, \boldsymbol{m}_0)$, $\boldsymbol{m}_h \in RS_D(\mathcal{N}, \boldsymbol{m}')$.*

When $\boldsymbol{m}_0$ is a home state, it is said that $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ is reversible.

Definition 2.17 ***Reversibility (R)***. *A system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is reversible if for any marking $\boldsymbol{m} \in RS_D(\mathcal{N}, \boldsymbol{m}_0)$ it holds that $\boldsymbol{m}_0 \in RS_D(\mathcal{N}, \boldsymbol{m}_0)$.*

A net $\mathcal{N}$ is *structurally* deadlock-free (resp. *structurally* live; resp. *structurally* reversible) if $\exists \boldsymbol{m}_0$ s.t. $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ is deadlock-free (resp. live; resp. reversible).

2.1.4 Petri net system subclasses

PN subclasses are usually defined by imposing some constraints on the structure of the net system. Some of them are defined below.

Definition 2.18 *(Petri net subclasses)*

- *Ordinary Petri nets are PN in which arc weights are equal to 1, i.e.* $\boldsymbol{Pre}, \boldsymbol{Post} \in \{0,1\}^{|P|x|T|}$.
- *State machines (SM) are ordinary Petri nets where each transition has one input and one output place, i.e.,* $\forall t, |^{\bullet}t| = |t^{\bullet}| = 1$.
- *Marked graphs (MG) [Com+71] are ordinary Petri nets where each place has one input and one output transition, i.e.,* $\forall p, |^{\bullet}p| = |p^{\bullet}| = 1$.
- *Join free (JF) nets are Petri nets in which each transition has at most one input place, i.e.,* $\forall t, |^{\bullet}t| \leq 1$.
- *Choice free (CF) nets [TCS97] are Petri nets in which each place has at most one output transition, i.e.,* $\forall p, |p^{\bullet}| \leq 1$.
- *Free choice (FC) nets [Hac72] are ordinary Petri nets in which conflicts are always equal, i.e.,* $\forall t, t'$*, if* $^{\bullet}t \cap {}^{\bullet}t' = \emptyset$*, then* $^{\bullet}t = {}^{\bullet}t'$.
- *Equal Conflict (EQ) nets [TS96] are Petri nets in which conflicts are always equal, i.e.,* $\forall t, t'$*, if* $^{\bullet}t \cap {}^{\bullet}t' = \emptyset$*, then* $\boldsymbol{Pre}[P,t] = \boldsymbol{Pre}[P,t']$.

Mono-T-Semiflow (MTS) nets [CCS91] are conservative Petri nets which have a unique T-semiflow whose support contains all the transitions.

Considering not only the PN structure, but also the initial marking, DSSP systems are defined. They model relations in which certain *modules* (modelled by state machines) cooperate through buffers (modelled by places which connect the *modules*). For example, in the DSSP in Fig. 2.4 (a), three state machines are connected with buffers b_1, b_2, b_3. They are more general than Equal Conflict nets, and have some characteristics in common. A property of DSSP is that all behavioural conflicts in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ are behaviourally equal: If t, t' are in conflict and both are enabled, then $Pre[P,t] = Pre[P,t']$. They are defined as follows.

Definition 2.19 *A PN system* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ *is a DSSP[RTS98], with* $\mathcal{N} = \langle P, T, Pre, Post \rangle$ *where* P *is the disjoint union of* $P_1, \ldots, P_n$ *and* B*, and* T *is the disjoint union of* $T_1, \ldots, T_n$*, and the following holds:*

1. *For every* $i \in \{1..n\}$*, let* $\mathcal{N}_i = \langle P_i, T_i, Pre[P_i, T_i], Post[P_i, T_i] \rangle$*. Then,* $\langle \mathcal{N}_i, \boldsymbol{m}_0[P_i] \rangle$ *is a live and safe state machine.*
2. *For every* $i, j \in \{1..n\}$*, if* $i \neq j$ *then* $Pre[P_i, T_j] = Post[P_i, T_j] = \mathbf{0}$.
3. *For each buffer* $b \in B$*:*
   (a) $dest(b) \in \{1..n\}$ *exists s.t.* $b^{\bullet} \subseteq T_{dest(b)}$.
   (b) *If* $t, t' \in p^{\bullet}$*, where* $p \in P_{dest(p)}$*, then Pre[b,t] = Pre[b,t'].*

A DSSP marking *is a marking for a DSSP net which respects the monomarkedness of the state machines (in continuous systems, the sum of all the tokens of each SM is equal to 1).*

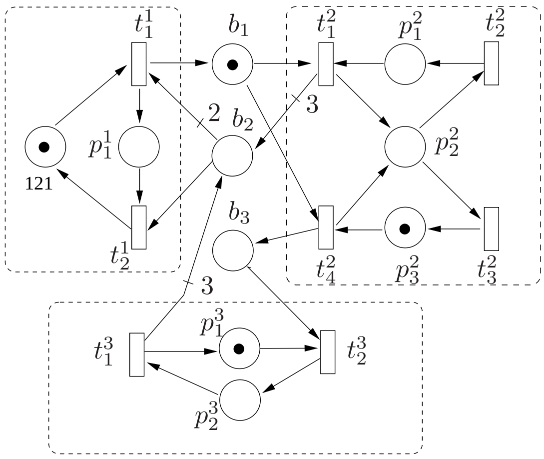


Figure 2.4: A consistent DSSP system

2.2 Linear Enabling Functions

Linear Enabling Functions (LEF) were introduced for discrete PN systems in [TCS93; BCS94], to characterize the enabling of transitions such that ${}^\bullet t_i > 1$ by a single linear expression.

The concept of linear enabling functions is presented in Section 2.2.1. Then, the LEF are used to check homothetic DF of a discrete PN system (Section 2.2.2). Finally, the addition of *representative* places for *join* transitions is presented in Section 2.2.3. These representative places will be used in Chapter 8 to apply the ρ-$semantics$ proposed there to *join* transitions.

2.2.1 The concept of Linear Enabling Functions

The purpose of LEF is to represent the enabling of a transition by a single linear expression.

In order to define LEF, transitions are classified in four Classes (obtained from the classification in five classes proposed in [BC94], from which the last two classes have been fused):

- **Class 1.** Transitions with a single input place: $|{}^\bullet t| = 1$.
- **Class 2.** Transitions with several input places, and for all of them its SB is equal to the weight of the arc to the transition: $\forall p \in {}^\bullet t$, $SB(p) = Pre[p, t_i]$.
- **Class 3.** Transitions with several input places, and all of them except one, denoted as pi, fulfil that its SB is equal to the weight of the arc to the transition: $\forall p \in {}^\bullet t \setminus \{\pi\}$, $SB(p) = Pre[p, t_i]$.
- **Class 4.** Transitions that do not belong to precedent classes. It is, that $SB(p) > Pre[p, t_i]$ for more than one of its input places p.

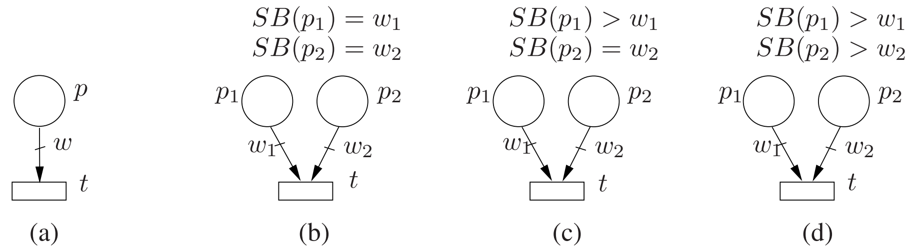


Figure 2.5: Classification of transition t related to the definition of LEFs. (a) Class 1, (b) Class 2, (c) Class 3, and (d) Class4.

The enabling degree of a transition t of Class 1, Class 2 or Class 3 can be represented with a single LEF [BCS94], as it is summarized below.

- **Class 1.** Transition t is enabled when $\underset{p\in{}^\bullet t}{m}[p] \geq Pre[p,t]$.
- **Class 2.** Transition t is enabled when $\sum_{p\in{}^\bullet t} m[p] \geq \sum_{p\in{}^\bullet t} Pre[p,t]$.
- **Class 3.** Transition t is enabled when $m[\pi] + SB(\pi)\cdot \sum_{p\in\{{}^\bullet t_i\setminus\pi\}} m[p] \geq Pre[\pi,t_i] + SB(\pi)\cdot \sum_{p\in\{{}^\bullet t_i\setminus\pi\}} Pre[p,t_i]$.
- **Class 4.** Its enabling cannot be directly represented by a single LEF, and some previous transformations of the PN should be done (see [BCS94]). One of the possible transformations is presented in Fig. 2.6.

2.2.2 Checking homothetic deadlock-freeness

The aim of this section is to characterize homothetic deadlock-freeness for structurally bounded Petri nets, by means of the basic definition of deadlock and the use of LEF. A PN system $\langle\mathcal{N},\boldsymbol{m}_0\rangle_D$ is homothetic deadlock-free if it is deadlock-free for $\boldsymbol{m}_0$ and for any proportional initial marking $\boldsymbol{m}'_0 = k\cdot\boldsymbol{m}_0$, for every k in the naturals.

A first method to check homothetic DF of $\langle\mathcal{N},\boldsymbol{m}_0\rangle_D$ can be to check monotonic DF, which implies homothetic DF. Monotonic DF can be checked with the siphon-trap property [Bra83; JC03]: If every siphon of $\mathcal{N}$ contains a marked trap which is marked at $\boldsymbol{m}_0$, then $\langle\mathcal{N},\boldsymbol{m}_0\rangle_D$ is monotonic DF (with some marking restrictions in the case of non-ordinary PN). However, checking this property is NP-complete, even for ordinary nets [OWW10].

In this section, a linear technique is presented to provide a sufficient condition for homothetic DF. For this purpose, a technique for the study of DF in discrete PN systems considered in [STC98] is recalled. It will allow us to analyse not only DF of a given system $\langle\mathcal{N},\boldsymbol{m}_0\rangle_D$, but also homothetic DF (for any scaled initial marking $k\cdot\boldsymbol{m}_0$).

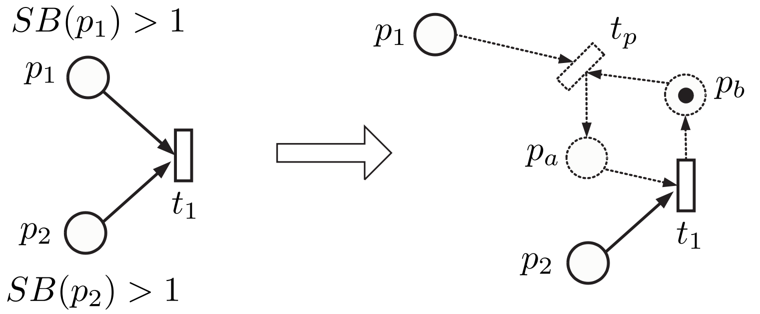


Figure 2.6: Transformation performed over a transition of Class 4. Places p_a and p_b are added, with $SB(p_a) = SB(p_b) = 1$, and the firing sequences are preserved. After the transformation, t become a transition of Class 3 because only p_2 holds $SB(p_2) > Pre[p_2, t]$, and the added transition t_p is also of Class 3, because only one of its input places (p_1) holds $SB(p_1) > Pre[p_1, t]$.

The following general sufficient condition for DF, based on the state equation, exploits the definition: "a deadlock corresponds to a marking in which no transition is fireable".

Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ be a PN system. If there does not exist any solution $(\boldsymbol{m}, \boldsymbol{\sigma})$ to the following system, then $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is deadlock-free.

$$\begin{array}{l} \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma} \\ \boldsymbol{m} \geq \boldsymbol{0}, \boldsymbol{\sigma} \geq \boldsymbol{0}, \\ \bigvee_{p \in {}^\bullet t} m[p] \leq Pre[p, t] - 1, \ \forall t \in T \end{array} \tag{2.4}$$

Nevertheless, notice that the system above contains $|T|$ "complex conditions" (one for each transition) which are non linear, due to the "$\vee$" connective. Thus, (2.4) can be handled by solving independently a set of $\prod_{t \in T} |{}^\bullet t|$ systems of linear inequalities, a quantity that grows exponentially: the number of linear systems is multiplied by $|{}^\bullet t|$ for each *join* transition.

Let us illustrate the key idea with the example in Fig. 2.7(a). Initially, the system that characterizes the sufficient condition for DF is: If there does not exist a solution to the following system, then the net system is DF (thus $2^3 = 8$ linear systems should be explored):

$$\begin{array}{ll} \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, & \\ \boldsymbol{m} \geq \boldsymbol{0}, \boldsymbol{\sigma} \geq \boldsymbol{0}, & \\ (m[p_1] = 0 \vee m[p_2] = 0), & \{t_1 \textit{ is not enabled}\} \\ (m[p_1] = 0 \vee m[p_3] = 0), & \{t_2 \textit{ is not enabled}\} \\ (m[p_4] = 0 \vee m[p_5] = 0) & \{t_3 \textit{ is not enabled}\} \end{array} \tag{2.5}$$

In [STC98], some transformation rules are considered in order to reduce the number of systems generated by (2.4), using LEF. Furthermore, in Theorem 34 of [STC98], it was proved that the system (2.4) can be rewritten as a single system of linear inequalities for every structurally bounded PN system.

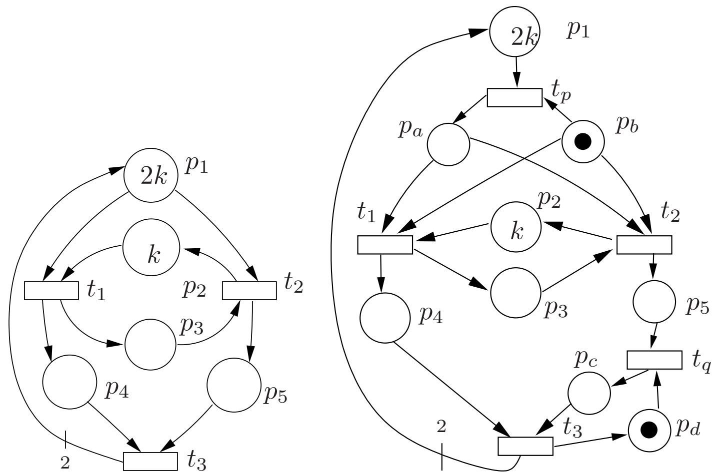


Figure 2.7: (a) PN system; (b) the same PN system, where t_p, t_q, p_a, p_b, p_c and p_d have been added in order to study DF with a single linear system.

First, the PN is transformed to a PN in which every transition has at most one input place whose SB is larger than the weight of its input arc. Then, the following rule is applied, which is analogous to the enabledness function for transitions of Class 3.

Reduction rule. Let t be a transition in Class 3 (i.e., ${}^\bullet t = \pi \cup \{p'\}$, where $SB(p) \leq Pre[p,t]$ for every $p \in \pi$. Then, the set of integer solutions of (2.5) is preserved if the disabledness condition corresponding to t is replaced by the following one:

$$SB(p') \cdot \sum_{p \in \pi} m[p] + m[p'] \leq SB(p') \cdot Pre[p,t] + Pre[p',t] - 1$$

The PN system (Fig. 2.7(a)) is first transformed to the one in Fig. 2.7(b), a transformation that preserves the reachable sequences (hence,it preserves DF): place p_1 is transformed to "p_1, t_p, p_a, p_b". Then, a term related to transition t_p needs to be added in order to express a deadlock: $m[p_1] = 0 \vee m[p_b] = 0$. Now, the rule can be applied to t_1 (with $p' = p_1$ and $\pi = \{p_b\}$) because $SB(p_1) = 2 \cdot k$ and $SB(p_b) = 1$. The term is transformed to $SB(p_1) \cdot m[p_b] + m[p_1] \leq SB(p_1) \cdot Pre[p_b,t] + Pre[p_1,t] - 1$, i.e., $m[p_1] + 2 \cdot k \cdot m[p_b] \leq 2k$. Moreover, $SB(p_a) = 1$ and $SB(p_2) = k$. Hence, non enabledness of t_1, $(m[p_1] = 0 \vee m[p_2] = 0)$, is reduced to $k \cdot m[p_a] + m[p_2] \leq k$. The other terms are analogously reduced.

The resulting system is:

$$
\begin{array}{ll}
\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, & \\
\boldsymbol{m} \geq \boldsymbol{0}, \boldsymbol{\sigma} \geq \boldsymbol{0}, & \\
k \cdot m[p_a] + m[p_2] \leq k, & \{t_1 \; is \; not \; enabled\} \\
k \cdot m[p_a] + m[p_3] \leq k, & \{t_2 \; is \; not \; enabled\} \\
\frac{3}{2}k \cdot m[p_c] + m[p_4] \leq \frac{3}{2}k, & \{t_3 \; is \; not \; enabled\} \\
m[p_1] + 2k \cdot m[p_b] \leq 2k, & \{t_p \; is \; not \; enabled\} \\
k \cdot m[p_d] + m[p_5] \leq k & \{t_q \; is \; not \; enabled\}
\end{array}
\tag{2.6}
$$

In this example, (2.6) has no solution, so the PN system is DF. In general, if a solution exists, it can be either a reachable deadlock or a *spurious* marking of the discrete net system (a *killing spurious* marking). In other words, system (2.6) only provides a sufficient condition for DF of discrete PN systems (semidecision).

If (2.6) has not real solutions, then the system is *homothetically* DF. In this example, system (2.6) obtained for the PN in Fig. 2.7(b), has no solution in the real domain. Consequently, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically DF.

2.2.3 Representative places for join transitions

The LEF of a transition of Class 2 or Class 3 can be represented in the PN system with an implicit place which is a *representative* place of the enabling of the discrete transition. Notice that transitions of Class 1 has a unique input place, hence it can be considered as its representative place. Transitions of Class 4 may also require a previous transformation to Classes 2 or 3.

The *representative* place of a transition t_i is defined as a linear combination of the input places of the transition, and it is added to the system as explained below:

- **Class 2.** The representative place r_i of a transition t_i is built as a linear combination of the places in ${}^\bullet t_i$, which represents the LEF for t_i. Place r_i corresponds to the linear combination of every place $p \in {}^\bullet t_i$, and it is built as:

$$
\boldsymbol{C}[r_i, T] = \sum_{p \in {}^\bullet t_i} \boldsymbol{C}[p, T] \tag{2.7}
$$

  And its initial marking is obtained as:

$$
m_0[r_i] = \sum_{p \in {}^\bullet t_i} m_0[p] \tag{2.8}
$$

- **Class 3.** The representative place r_i of a transition t_i is a linear combination of the places in ${}^\bullet t_i$, which represents the LEF for t_i. Place r_i is built as:

$$
\boldsymbol{C}[r_i, T] = \boldsymbol{C}[\pi, T] + SB(\pi) \cdot \sum_{p \in \{{}^\bullet t_i \setminus \pi\}} \boldsymbol{C}[p, T] \tag{2.9}
$$

And its initial marking is obtained as:

$$m_0[r_i] = m_0[\pi] + SB(\pi) \cdot \sum_{p \in \{^\bullet t_i \setminus \pi\}} m_0[p] \tag{2.10}$$

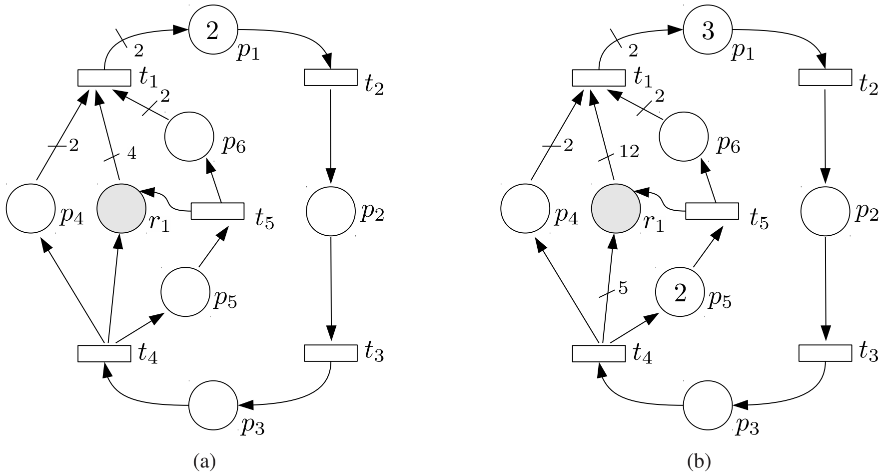


Figure 2.8: Without the grey places, PN system with two different initial markings: (a) $\boldsymbol{m}_0 = (2,0,0,0,0,0)$ and (b) $\boldsymbol{m}'_0 = (3,0,0,0,2,0)$. The addition of the representative places r_1 drawn in grey colour preserves their firing sequences as discrete PN systems.

Consider the PN in Fig. 2.8(a) without the grey places with $\boldsymbol{m}_0 = (2,0,0,0,0,0)$. Transition t_1 belongs to Class 2, because $SB(p_4) = 2 = Pre[p_4, t_1]$, and $SB(p_6) = 2 = Pre[p_6, t_1]$. Hence, the enabling of t_1 can be represented by $m[p_4] + m[p_6] \geq Pre[p_4, t_1] + Pre[p_6, t_1)$, i.e., $m[p_4] + m[p_6] \geq 4$. The *representative* place r_1 has been added, which is created as the addition of places p_4 and p_6, i.e., $p_4 + p_6$.

Consider now $\boldsymbol{m}'_0 = (3,0,0,0,2,0)$ (Fig. 2.8(b), without the grey places). In this case, transition t_1 belongs to Class 3. Considering $\pi = p_6$, it holds $SB(p_6) = 5 > Pre[p_6, t_1] = 2$, and all the other input places of t_1, i.e., $^\bullet t_1 \setminus \{\pi\} = \{p_4\}$, hold the equality, $SB(p_4) = Pre[p_4, t_1] = 2$. The enabling of transition t_1 in the discrete PN system can be represented by the following condition: $m[p_6] + SB(p_6) \cdot m[p_4] \geq Pre[p_6, t_1] + SB(p_6) \cdot Pre[p_4, t_1]$. Representative place r_1 has been created as $p6 + SB(p_6) \cdot p_4$, which corresponds to $p_6 + 5 \cdot p_4$ (see the grey places r_1 in Figs.2.8(a) and (b)).

The added place r_i is constructed as a linear combination of the places of the original PN system (both in Class 2 and Class 3). Consequently, its marking is also a linear combination of markings in $^\bullet t_i \setminus \{r_i\}$, and it will be never the one constraining the enabling degree of t_i. Hence, by construction, it is implicit in the discrete system.

Proposition 2.20 *The representative place* r_i *is implicit in the obtained system* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$.

Once the implicit place r_1 has been added, places p_4 and p_6 become implicit in the discrete net system and they could be removed.

2.3 Timed Petri nets

By introducing time to the model, timed PN are obtained. A simple and broadly used way to introduce time to PN is to assume that time is associated to transitions, which is addressed here. Other methods consist in adding time to the places, the arcs, etc. Stochastic PN (see [Mol82; AM+95]) are presented below as a time interpretation of discrete PN. Fluidization of stochastic PN will be considered in Part III in this thesis.

2.3.1 Stochastic Petri nets

A Markovian *Stochastic Petri Net* (SPN) system is a discrete PN system in which the transitions fire at independent exponentially distributed random time delays, and conflicts are solved with a race policy (given two transitions having a common input place, the one having the lowest associated delay will fire).

The firing time of each transition is characterized by its firing rate, which corresponds to the average delay associated to each server in the corresponding transition. A SPN is defined as follows:

Definition 2.21 *A SPN is a tuple* $\langle \mathcal{N}, \boldsymbol{m}_0, \boldsymbol{\lambda} \rangle$, *where* $\mathcal{N}$ *is the PN,* $\boldsymbol{m}_0 \in \mathbb{N}^{|P|}$ *is the initial marking, and* $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$ *is the vector of rates associated to the transitions.*

In this work, *infinite server semantics* (ISS) is assumed for all transitions. The system evolves as a jump Markov process where the firing time of an enabled transition t_i, at a given marking $\boldsymbol{m}$, is given by a random variable which follows an exponentially distributed function with parameter $\lambda_i \cdot enab(t_i, \boldsymbol{m})$. The reachability graph of a SPN system is isomorphous to a Markov Chain [Mol82].

The average marking vector, $\overline{\boldsymbol{m}}$, in an ergodic [AM+95] SPN system is defined as follows [FN89]:

$$\overline{m}[p] \underset{AS}{=} \lim_{\tau \to \infty} \frac{1}{\tau} \int_0^{\tau} m[p]_u \, \mathrm{d}u \tag{2.11}$$

where $m[p]_u$ is the marking of place p at time u and the notation $\underset{AS}{=}$ means *equal almost surely*.

Similarly, the steady-state throughput, $\chi_{SPN}(t)$, in an ergodic SPN is defined as [FN89]:

$$\chi_{SPN}(t) \underset{AS}{=} \lim_{\tau \to \infty} \frac{\boldsymbol{\sigma}[t]_\tau}{\tau} \tag{2.12}$$

where $\boldsymbol{\sigma}[t]_\tau$ is the firing count of transition t at time τ.

3 Previous transformations of the original discrete system

"*Hope for the best, prepare for the worst*".
Chris Bradford

This chapter proposes some basic transformations of the original discrete PN system with the aim of obtaining a better approximation when it is fluidified.

The transformations are based on the addition of some places which are implicit in the discrete PN system, i.e., they modify the LRS_C of the PN system but not the RS_D. Consequently, they may modify the behaviour of the continuous system, making it more faithful to the discrete one, as it is illustrated in Part III. These techniques consider the structure of the net ($\mathcal{N}$) and the initial marking of the system ($\boldsymbol{m}_0$), so they can be performed over a given PN system, but not to the net structure itself. These transformations add some *cutting* places which are *implicit* in the discrete system [STC98] but they modify the behaviour of the continuous one.

Introduction

When a discrete PN system is fluidified, some solutions of the LRS_C which are not reachable in the discrete system become reachable in the continuous [SR02]. In order to obtain a better approximation of the original PN system when the system will be fluidified, some basic transformations of the net are presented to remove those markings from the LRS_C, to make them also not reachable in the continuous system, which is specially interesting if they are deadlock markings. This chapter proposes the addition of implicit places to remove some *spurious* behaviour which may appear in the fluidified system.

There can be two kinds of markings which are not reachable in the discrete system but belong to its LRS_C: integer markings, which were *spurious* in the discrete system [STC98]; or non integer markings, which cannot be reachable in the discrete, but it is also interesting to avoid them if they are deadlocks or vertex of the polytope defined by LRS_C.

Those integer *spurious* markings which are due to the emptying of a trap can be removed by a classical technique from the literature. Some results from [STC96] (later applied to continuous PN in [Sil+11]) about the addition of implicit places which remove such spurious deadlock markings are recalled in Section 3.1.

"Dual" in some sense to the well-known technique to avoid empty traps recalled in Section 3.1, a new one is introduced in Section 3.2 with the same purpose, but requiring an a priori knowledge of the non-reachability of the marking being investigated.

Considering non integer solutions, some classical works aim to remove the non-integer vertices of a polytope, such as the Gomory-Chvàtal cuts. Given a real polytope, they *cut* the markings outside the *integer hull* of the polytope [Bal+96]. This method could be used to remove undesired non integer spurious markings. However, these techniques can be not very efficient in the general case [Cor12; Dun11]. Gomory cuts are tractable for a well chosen set of equations, but finding a good family of cuts is still an open problem [Cor12].

In this work, we propose to implement some *cuts* on the polytope *ad hoc* for a given PN system, particularly considering the structure of the net. The cut is obtained with an *implicit place* which limits the maximal marking of each place to the higher possible integer. These implicit places modify the $LRS_C(\mathcal{N}, \boldsymbol{m}_0)$, however, they do not modify the $RS_D(\mathcal{N}, \boldsymbol{m}_0)$.

We propose to implement some *cuts* on the polytope, considering the Petri net structure. These *cuts* aim to avoid a spurious marking, and they are obtained by means of an *implicit* places which force a marking relation. We propose three different kinds of *implicit* places to avoid such non-integer vertices of the polytope: *rational vertex cutting* places avoid those vertices which are non-integer; *marking truncation* places are a particular case of those places but more efficient; *enabling truncation* places do not modify the set of reachable markings, but they can modify the firing amounts of the transitions.

Moreover, the *enabling truncation* places introduced in Section 3.5 do not modify the reachable markings, but they avoid some *spurious* behaviour. They limit the firing of a given transition by truncating the marking to the highest possible integer, which can improve the approximation of a temporized PN system. These transformations often obtain a significant improvement, and they require low computational costs.

3.1 Places to avoid the emptying of a trap

A technique developed for removing spurious solutions which are due to the emptying of an initially marked trap is considered in [STC98] for discrete PN systems, and applied to continuous PN systems in [Sil+11]. This kind of spurious solutions, which could be reached in the continuous system by firing an infinite firing sequence, are removed by adding some places which are *implicit* in the discrete model.

It is well known that a marked trap cannot be emptied in a discrete PN. However, traps may be emptied in continuous PN in the limit, considering infinite firing sequences (lim-reachability)[RTS99]. If a solution of the state equation $\boldsymbol{m}$ does not mark a trap which was marked at $\boldsymbol{m}_0$, then it is *spurious* in the discrete system. The technique consists in using a trap generator [ECS93], what allows the checking of implicit places in polynomial time (a sufficient condition). Then, a monitor *implicit* place is added to the system which adds an invariant (a p-semiflow) which forces the trap to remain marked.

A *spurious* marking in the discrete system is a solution of the state equation (also of the system (2.6)) which is not reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ (see Section 2.1.2). Let us remark that this technique removes markings that for sure are spurious in the discrete system, because it considers traps being emptied, which is not possible in a discrete system.

An example with this type of spurious deadlocks is the PN in Fig. 3.1(a) without considering the grey places, with initial marking $\boldsymbol{m}_0$ = (2, 2, 0). This PN system is live as discrete. However, its LRS_C contains three *spurious* deadlock markings (see the *reachability graph* of the net system in Fig. 3.1(b), where the shaded markings correspond to spurious solutions). Consider $\boldsymbol{m}_d = (0, 4, 0)$, it corresponds to $\boldsymbol{\sigma} = (0, 0, 2)$, which is not a fireable sequence.

The existence of a trap that is initially marked but can be emptied in the limit can be characterized with a set of linear inequations, which consists in a *trap generator* and the expression that an initially marked trap becomes empty. Let us define $\mathbf{Pre}_\Theta$ and $\mathbf{Post}_\Theta$ as $|P| \times |T|$ sized matrices such that:

- $\mathbf{Pre}_\Theta[p, t] = 1$ if $\mathbf{Pre}[p, t] > 0$, $\mathbf{Pre}_\Theta[p, t] = 0$ otherwise
- $\mathbf{Post}_\Theta[p, t] = |{}^\bullet t|$ if $\mathbf{Post}[p, t] > 0$, $\mathbf{Post}_\Theta[p, t] = 0$ otherwise.

Equations $\{\boldsymbol{y}^T \cdot \boldsymbol{C}_\Theta \geq \mathbf{0},\ \boldsymbol{y} \geq \mathbf{0}\}$ where $\boldsymbol{C}_\Theta = \mathbf{Post}_\Theta - \mathbf{Pre}_\Theta$ define a generator of traps (Θ is a trap iff $\exists \boldsymbol{y} \geq \mathbf{0}$ such that $\Theta = \|\boldsymbol{y}\|$, $\boldsymbol{y}^T \cdot \boldsymbol{C}_\Theta \geq \mathbf{0}$) [ECS93; STC98]. Hence, given $\boldsymbol{m}$ a solution of the state equation, we can check in polynomial time a sufficient condition for being spurious:

Proposition 3.1 *Given* $\boldsymbol{m} \in \mathbb{N}_{\geq 0}^{|P|}$ $(\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma},\ \ \boldsymbol{m}, \boldsymbol{\sigma} \geq 0)$, *if*

- $\boldsymbol{y}^T \cdot \boldsymbol{C}_\Theta \geq \mathbf{0}, \boldsymbol{y} \geq \mathbf{0}$, {*trap generator*}
- $\boldsymbol{y}^T \cdot \boldsymbol{m}_0 \geq 1$, {*initially marked trap*}
- $\boldsymbol{y}^T \cdot \boldsymbol{m} = 0$, {*trap empty at* $\boldsymbol{m}$}

has solution, then $\boldsymbol{m}$ *is a* spurious *solution in the discrete PN system.*

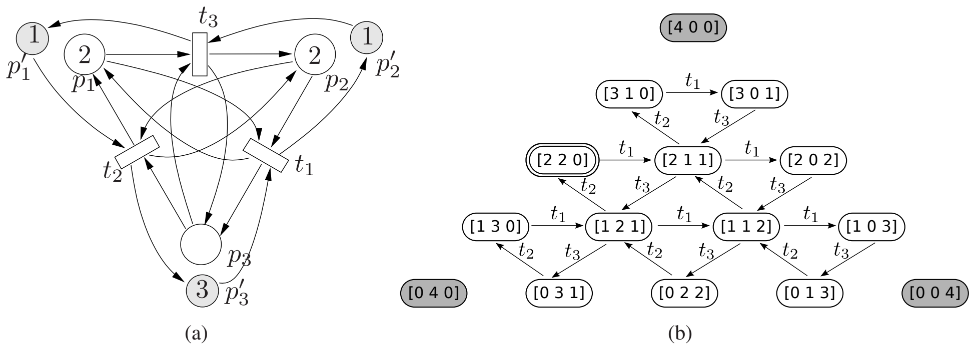


Figure 3.1: (a) Without the grey places, deadlock-free discrete PN system with three *spurious* deadlocks; and (b), its reachability graph as discrete, where the shaded markings correspond to spurious solutions, all isolated deadlocks.

Firstly proposed for discrete PN [STC98], the technique presented in [STC98; Sil+11] removes this kind of spurious deadlocks by adding some *implicit* places in the discrete model. Let us consider the trap $\Theta_1 = \{p_1, p_3\}$ in the net in Fig. 3.1(a). Since it is initially marked, in the discrete model its marking must satisfy $m[p_1] + m[p_3] \geq 1$. Considering the token conservation low obtained from the P-semiflow, $m[p_1] + m[p_2] + m[p_3] = 4$, it leads to $m[p_2] \leq 3$. This last inequality can be forced by adding a slack variable to the system, i.e., a *cutting implicit place* p_2' (shown in Fig.3.1(a)), such that $m[p_2] + m[p_2'] = 3$. The initial marking of p_2' can be simply set as $m_0[p_2'] = 3 - m_0[p_2] = 1$. By adding p_2', the spurious deadlock marking $\boldsymbol{m}_d = (0, 4, 0)$, in which trap $\Theta_1 = \{p_1, p_3\}$ is empty, is removed. Similarly, by adding p_1', p_3', the spurious markings which empty traps $\Theta_2 = \{p_2, p_3\}$ and $\Theta_3 = \{p_1, p_2\}$ are also removed.

It is interesting to remark that, by removing spurious deadlocks (in fact, any spurious marking), the approximation of the performance of the discrete net system, provided by the timed relaxation, is also improved (see Chapter 7). This is true even if the deadlock is not reached in the timed continuous model. In any case, removing spurious solutions represents an improvement of the fluidization, being specially important when those solutions are deadlocks or non-live steady states. Moreover, the added implicit places are also useful for the improvement of structure based performance bounds for discrete net systems [CCS92].

3.2 Places to avoid the emptying of a siphon

Here, we propose another technique to remove such spurious solutions which can be reached by firing even a *finite* firing sequence in the corresponding continuous PN system.

Dual in some sense, they would be removed by preventing certain siphons from being emptied, with the addition of some places which are *implicit* in the discrete model. An important difference with respect to the previous technique is that some of the *deadlock* markings that now can be removed may be reachable by the original discrete system if it was not deadlock-free. Consequently, this technique should be applied to remove solutions that *we know* that are spurious. The first step is to identify the existence of a siphon that is initially marked but it is emptied in a marking $\boldsymbol{m}_d$ (which is a solution of the state equation). It can be characterized by a set of linear inequalities, which consists in a *siphon generator* [ECS93] and the expression that an initially marked siphon becomes empty.

Let us define $\mathbf{Pre}_\Sigma$ and $\mathbf{Post}_\Sigma$ as $|P| \times |T|$ sized matrices such that:

- $Pre_\Sigma[p,t] = |t^\bullet|$ if $Pre[p,t] > 0$, $Pre_\Sigma[p,t] = 0$ otherwise
- $Post_\Sigma[p,t] = 1$ if $Post[p,t] > 0$, $Post_\Sigma[p,t] = 0$ otherwise.

Equations $\{\boldsymbol{y}^T \cdot \boldsymbol{C}_\Sigma \geq \mathbf{0},\ \boldsymbol{y} \geq \mathbf{0}\}$ where $\boldsymbol{C}_\Sigma = \mathbf{Post}_\Sigma - \mathbf{Pre}_\Sigma$ define a generator of siphons (Σ is a siphon iff $\exists \boldsymbol{y} \geq \mathbf{0}$ such that $\Sigma = \|\boldsymbol{y}\|$, $\boldsymbol{y}^T \cdot \boldsymbol{C}_\Sigma \geq \mathbf{0}$, analogous to the generator of traps [ECS93; STC98]). Hence, given $\boldsymbol{m}_d$ a solution of the state equation reported as a deadlock, we can check in polynomial time if a minimal siphon marked at $\boldsymbol{m}_0$ is unmarked at $\boldsymbol{m}_d$, and report which is such siphon.

Proposition 3.2 *Given* $\boldsymbol{m}_d \in \mathbb{R}^{|P|}_{\geq 0}$ ($\boldsymbol{m}_d = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, $\boldsymbol{m}_d$, $\boldsymbol{\sigma} \geq \mathbf{0}$), *if*

- $\boldsymbol{y}^T \cdot \boldsymbol{C}_\Sigma \geq \mathbf{0}, \boldsymbol{y} \geq \mathbf{0},$ {*siphon generator*}
- $\boldsymbol{y}^T \cdot \boldsymbol{m}_0 \geq 1,$ {*initially marked siphon*}
- $\boldsymbol{y}^T \cdot \boldsymbol{m} = 0,$ {*siphon empty at* $\boldsymbol{m}_d$}

has solution, then a siphon Σ *marked at* $\boldsymbol{m}_0$ *is emptied at* $\boldsymbol{m}_d$.

Once a siphon which has been emptied is identified, it can be forced to remain marked by the addition of a place which is *implicit* in the discrete model.

Notice that the added places are *implicit* in the discrete PN system, however they make the system (lim)-live as continuous. Let us apply the method to a second example.

Consider the system in Fig. 3.2, which reaches the continuous deadlock $\boldsymbol{m}_d$, where $m_d[r_3] = m_d[q_3] = m_d[p_3] = 1$, and $m_d[q_i] = m_d[p_i] = m_d[r_i] = 0\ \forall i \in \{0,1,2,4\}$. This deadlock occurs due to the fact that siphon $\Sigma_1 = \{q_1, q_4, p_1, p_4, r_1, r_2\}$ has been emptied (notice that once a siphon is emptied, it is never marked again).

The technique presented here adds a monitor place to prevent this siphon from being emptied. In order to keep Σ_1 marked in the continuous net system, the following inequality should be forced: $m[q_1]+m[q_4]+m[p_1]+m[p_4]+m[r_1]+m[r_2] \geq 1$. It can be forced by the addition of a slack variable u_1, i.e., a *cutting* implicit place, to force the following invariant relation: $m[q_1] + m[q_4] + m[p_1] + m[p_4] + m[r_1] + m[r_2] - m[u_1] = 1$. Since $m[u_1] \geq 0$, siphon Σ_1 will remain marked. Place u_1 is shown in the dotted part in Fig. 3.2, it has s_4 and t_4 as input transitions and s_3 and t_3 as output transitions and its initial marking can be calculated from the invariant: $m_0[u_1] = m_0[q_1] + m_0[q_4] + m_0[p_1] + m_0[p_4] + m_0[r_1] + m_0[r_2] - 1 = 0+0+0+0+1+1-1 = 1$.

The added place u_1 is *implicit* in the discrete PN system and it removes the marking $\boldsymbol{m}_d$ from the LRS_C.

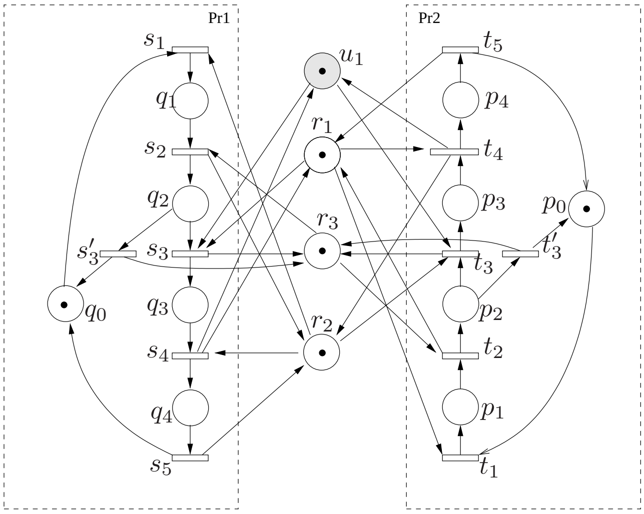


Figure 3.2: Without the dotted place u_1 and the related arcs, it is a deadlock-free discrete system. It deadlocks as continuous, reaching $\boldsymbol{m}_d = (0,0,0,1,0,0,0,0,1,0,0,0,1)$, which was a *spurious* deadlock of the state equation ([GV99], p. 111).

3.3 Rational vertex cutting places

The aim of this technique is to *cut* the non-integer vertices of the polytope of LRS_C.

Consider the example in Fig. 3.6(a). It has four reachable markings as discrete: $\boldsymbol{m}_0 = (1,0,1,0,0)$, $\boldsymbol{m}_1 = (0,1,0,3,0)$, $\boldsymbol{m}_2 = (0,0,1,0,3)$, and $\boldsymbol{m}_3 = (0,0,1,1,1)$. The polytope of the markings contained in its LRS_C is the convex set defined by the those markings as vertices and $(0,0,1,1.5,0)$ as other vertex. Some of the markings of the $LRS(\mathcal{N}, \boldsymbol{m}_0)$ are outside the "integer hull", such as $\boldsymbol{m}_d = (0,0,1,1.5,0)$, which moreover it is a vertex of the polytope. Marking $\boldsymbol{m}_d$ is not reachable by the discrete PN because $\boldsymbol{m}_d \notin \mathbb{N}^{|P|}$.

Because it is a vertex and it is not reachable, in the nearest vertices, at least one of the places which are empty at $\boldsymbol{m}_d$ (i.e., p_1, p_2 and p_5) should be marked (with a marking equal or greater than 1, in the discrete model), otherwise it would not be a vertex. Hence, we can assure that the following inequality holds for every discrete marking: $m[p_1] + m[p_2] + m[p_5] \geq 1$. This inequality can be forced by the addition of a place v which is *implicit* in the discrete (but not in the continuous) PN: $m[p_1] + m[p_2] + m[p_5] - m[v] = 1$. From this equation, place v is defined as $\boldsymbol{C}[v,T] = \boldsymbol{C}[p_1,T] + \boldsymbol{C}[p_2,T] + \boldsymbol{C}[p_5,T]$, and $m_0[v] = m_0[p_1] + m_0[p_2] + m_0[p_5] - 1$, as depicted in Fig. 3.3.

Place v, here denoted as *rational vertex cutting* place, adds the invariant $m[p_1] + 2 \cdot m[p_3] + m[p_4] + m[v] = 3$ to the net, and $\boldsymbol{m}_d$ becomes not reachable in the continuous system.

Given a non-integer vertex $\boldsymbol{m}_v$, a place v can be added to force the following relation:

$\sum_{i|m_v[p_i]=0} m[p_i] \geq 1$. The obtained *rational vertex cutting* place v is implicit in the discrete PN and it cuts the marking $\boldsymbol{m}_v$ in the continuous system. Enumerating all the vertices of the polytope is computationally costly [AF92]. However, checking if a given solution $\boldsymbol{m}_v$ is a vertex of a polytope can be done in polynomial time, as discussed below.

A vertex $\boldsymbol{m}_v$ of the polytope defined by $LRS(\mathcal{N}, \boldsymbol{m}_0)$ is a solution which is not a linear combination of other solutions in $LRS(\mathcal{N}, \boldsymbol{m}_0)$, i.e. $\nexists\ \boldsymbol{m}_1, \boldsymbol{m}_2 \in LRS(\mathcal{N}, \boldsymbol{m}_0)$ with $\boldsymbol{m}_1 \neq \boldsymbol{m}_2$ s.t. $\boldsymbol{m}_v = \alpha \boldsymbol{m}_1 + \beta \boldsymbol{m}_2$ for arbitrary $\alpha, \beta > 0$. Hence, any vertex is a minimal support vector:

Proposition 3.3 *A solution $\boldsymbol{m}_v \in LRS(\mathcal{N}, \boldsymbol{m}_0)$ is a vertex of $LRS(\mathcal{N}, \boldsymbol{m}_0) \Rightarrow \boldsymbol{m}_v$ is a minimal support solution in $LRS(\mathcal{N}, \boldsymbol{m}_0)$*

The reverse is not true if $\mathcal{N}$ is not conservative. For example, consider the PN system in Fig. 3.3. Every marking is minimal support, but here only $\boldsymbol{m}_0 = (1)$ is a vertex. A polynomial time method is presented below to check if a given solution of the state equation is a vertex.

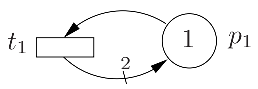


Figure 3.3: Non consistent, non conservative PN system.

Proposition 3.4 *A solution $\boldsymbol{m}_v \in LRS(\mathcal{N}, \boldsymbol{m}_0)$ is a vertex iff $\forall p_i \in P$, v_i is equal to 0, where v_i is defined by the following maximization problem:*

$$\begin{aligned} v_i = \quad & max \quad (m_v[p_i] - m_1[p_i]) \\ & s.t. \quad \boldsymbol{m}_1 = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}_1, \boldsymbol{m}_1, \boldsymbol{\sigma}_1 \geq \boldsymbol{0} \\ & \qquad\ \ \boldsymbol{m}_2 = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}_2, \boldsymbol{m}_2, \boldsymbol{\sigma}_2 \geq \boldsymbol{0} \\ & \qquad\ \ \boldsymbol{m}_v = 0.5\boldsymbol{m}_1 + 0.5\boldsymbol{m}_2 \end{aligned} \tag{3.1}$$

Proof. ($\Rightarrow$) Assume $\boldsymbol{m}_v$ is a vertex. By definition, $\nexists \boldsymbol{m}_a, \boldsymbol{m}_b$ s.t. $\boldsymbol{m}_a \neq \boldsymbol{m}_b$ and $\boldsymbol{m}_v = 0.5 \cdot \boldsymbol{m}_1 + 0.5 \cdot \boldsymbol{m}_2$. Hence, the maximization problem (3.1) finds that $\boldsymbol{m}_1 = \boldsymbol{m}_2 = \boldsymbol{m}_v$ is the optimal solution.
($\Leftarrow$) Assume $\boldsymbol{m}_v$ is not a vertex. Then, there exists two interchangeable solutions $\boldsymbol{m}_1, \boldsymbol{m}_2 \in LRS(\mathcal{N}, \boldsymbol{m}_0)$ s.t. $\boldsymbol{m}_1 \neq \boldsymbol{m}_2$ and $\boldsymbol{m}_v = 0.5 \cdot \boldsymbol{m}_1 + 0.5 \cdot \boldsymbol{m}_2$. Hence, there exists at least a place p_i s.t. $\boldsymbol{m}_1[p_i] < \boldsymbol{m}_v[p_i] < \boldsymbol{m}_2[p_i]$. Therefore, $v_i > 0$. ■

Proposition 3.4 can be checked in polynomial time, by solving a set of $|P|$ LPP, from which the first phase of the classical simplex approach is common, because only the objective function changes.

If $\mathcal{N}$ is consistent, the number of variables and restrictions can be reduced, by replacing $\boldsymbol{m}_k = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}_k$ by $B \cdot \boldsymbol{m}_k = B \cdot \boldsymbol{m}_0$, for $k = \{1, 2\}$, where B is a basis of $p\text{-}flows$ of $\mathcal{N}$ [STC98]. Then, v_i is defined as:

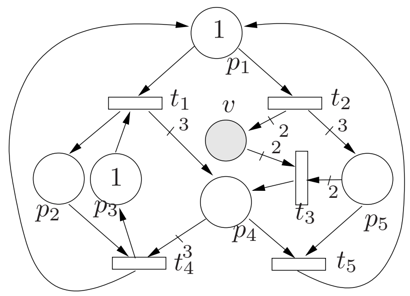


Figure 3.4: Petri net system. The *rational vertex cutting* place drawn in grey colour cuts the undesired *spurious* deadlock $\boldsymbol{m}_d = (0, 0, 1, 1.5, 0)$, which was not cut by any *marking truncation* place.

$$\begin{array}{rll} v_i = & max & (m_v[p_i] - m_1[p_i]) \\ & s.t. & B \cdot \boldsymbol{m}_1 = B \cdot \boldsymbol{m}_0, \boldsymbol{m}_1 \geq \mathbf{0} \\ & & B \cdot \boldsymbol{m}_2 = B \cdot \boldsymbol{m}_0, \boldsymbol{m}_2 \geq \mathbf{0} \\ & & \boldsymbol{m}_v = 0.5 \cdot \boldsymbol{m}_1 + 0.5 \cdot \boldsymbol{m}_2 \end{array} \quad (3.2)$$

3.4 Marking truncation places

A particular case of the *rational vertex cutting* place is presented here. The *marking truncation* places can be efficiently computed and added. Given that a marking of a discrete PN system belongs to $\mathbb{N}^{|P|}$, a place p_j will never have more tokens than its *structural enabling bound* in the naturals, $SB(p)$. However, this bound can be overlooked by the continuous system.

The addition of an implicit place q_j is proposed, which is a complementary place of each p_j (i.e., $Pre[q_j, T] = Post[p_j, T]$ and $Post[q_j, T] = Pre[p_j, T]$) which truncates the marking of p_j to the nearest integer, and consequently, undesired markings are avoided. The initial marking of such complementary place is: $m_0[q_j] = SB(p_j) - m_0[p_j]$ (with $SB(p_j)$ defined in the naturals).

For example, consider the PN in Fig. 3.5. The *marking truncation* place q_3 is added, which is *complementary* to p_3 and whose initial marking is $\boldsymbol{m}_0[q_3] = \lfloor 1.5 \rfloor - 0 = 1$. It is implicit in the discrete system, but it modifies the behaviour of the continuous one. It avoids to have more than 1 token in p_3, hence it avoids some spurious markings of the original system such as $\boldsymbol{m}_3 = (0, 1, 1.5, 0)$ (which is a continuous deadlock), in which $m_3[p_3] = 1.5$ was bigger than $SB(p_3) = 1$. Places q_1, q_2 and q_4 would not modify the continuous system, so they have not been added.

An interesting question is whether these *cuts* obtain the "convex hull" of the polytope defined by the reachable markings of $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ as the Gomory-Chvàtal cuts would do.

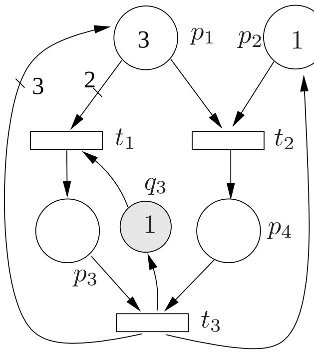
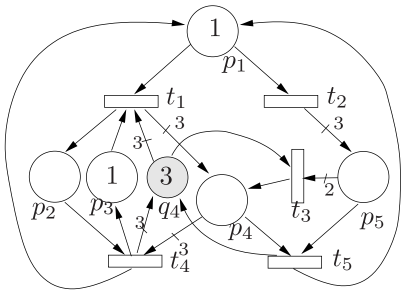


Figure 3.5: Petri net system. The *marking truncation* place q_3 drawn in grey colour is implicit in the discrete system, but removes the undesired markings from its LRS_C.



(a)

Figure 3.6: Petri net system. The *marking truncation* place drawn in grey colour does not *cut* the *spurious* deadlock marking $\boldsymbol{m}_d = (0, 0, 1, 1.5, 0)$, which is a vertex of the reachable polytope.

However, this technique is less powerful that the *rational vertex cutting* places. For example, consider again the PN system in Fig. 3.6(a). The *marking truncation* place q_4 is added to limit that of p_4. However, $\boldsymbol{m}_d$ is still reachable by the continuous PN (the rest of the places q_j would not avoid it either). Hence, the *marking truncation* places do not always obtain the "integer hull" of the polytope defined by the LRS_C.

3.5 Enabling truncation places

Finally, the *enabling truncation* places presented here do not modify the reachability set of the continuous PN system, but they limit its flow at a given marking if time is considered. They can have an effect over the throughput even if $SB(p)$ is integer (case in which the

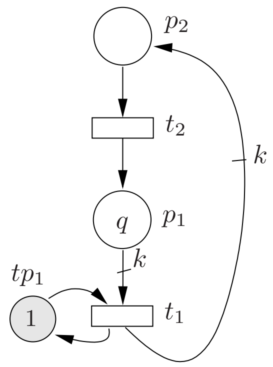


Figure 3.7: Petri net system. Considering $k \leq q \leq 2 \cdot k$, the *enabling truncation* place drawn in grey colour is implicit as discrete but it can modify the throughput of the continuous system (see Section 7.2.1).

marking truncation place would have no effect).

Analogously to the marking of a place, a transition t_i will never be fired an amount higher than its *structural enabling bound*: $SEB(t_i)$. The addition of an implicit place tp_i is proposed, to truncate the maximal possible firing of the transition to the highest possible integer. Such highest integer corresponds to $SEB(t_i)$.

Consider the PN in Fig. 3.5 with $k = 3$, $q = 5$ and $\boldsymbol{\lambda} = (1, 5)$. The SEB of t_1 is $SEB(t_1) = \lfloor 1.67 \rfloor = 1$. Hence, an *enabling truncation* place tp_1 with initial marking equal to 1 is added (see place tp_1, drawn in grey colour Fig. 3.5).

The added place does not modify the LRS_C of the PN, but it will have an effect over the behaviour of the TCPN when the system will be fluidified (see Chapter 7).

3.6 Conclusions

In this chapter, some basic transformations to be done over a discrete PN system before its fluidization have been proposed. These transformations consider both the structure of the net and its initial marking, and they are based on the addition of some places which are implicit in the discrete system, which cut some *vertex* (or sets of markings on its surface) of the potential *polytope*. The aim of most of these techniques is to remove from the state equation the markings which are not reachable by the discrete PN system.

Some of those *non reachable* markings are the solutions of the state equation in which a previously marked trap is empty. A technique to remove those markings was proposed in the literature for discrete PN [STC98], and recalled for discrete PN. Analogously, a technique to avoid markings in which a siphon is emptied has been proposed here. The weakness of this technique is that it does not assure a priori that the removed marking was spurious, it has to be checked in advance. These two techniques can eliminate both integer or non integer solutions from the state equation.

Due to the fact that $RS_D(\mathcal{N}, \boldsymbol{m}_0) \subseteq LRS_C(\mathcal{N}, \boldsymbol{m}_0)$, if there is a vertex of the state equation in the real numbers which is not natural, this marking is not reachable either. That vertex be also removed from the state equation by the addition of the *rational vertex cutting* places. A particular case of those markings are those solutions in which the marking of a place p is bigger than its natural structural bound $SB(p)$. These markings can be removed from the state equation by means of the *marking truncation* places proposed here.

Finally, the addition of other *implicit* places, called *enabling truncation* places, is proposed which do not alter the set of reachable markings, but they can modify the flow of the continuous PN in some situations.

Part II

On the fluidization of untimed Petri nets

4

Previous concepts on the fluidization of untimed Petri nets

"*Everything must be made as simple as possible. But not simpler*".
Albert Einstein

This chapter introduces the concepts related to the fluidization of untimed PN. The formal definition of continuous PN is presented first. Then, basic reachability concepts are introduced, and also some properties of systems which are used in the rest of the document are defined: synchronic properties such as boundedness and other behavioural properties such as *deadlock-freeness*, *liveness*, *home state* and *reversibility*.

Introduction

Some concepts of untimed PN are presented in this section. First, the concepts related to the PN structure and discrete and continuous PN systems are defined. Then, some reachability concepts related to both formalisms are presented. Finally, classical properties from discrete PN are defined for continuous PN. They are presented in two groups: synchronic properties, which include boundedness and B-fairness among others; and other behavioural properties such as deadlock-freeness, liveness or reversibility.

4.1 Continuous Petri net systems

The main difference between discrete and continuous PN is in the firing amounts and consequently in the marking, which in *discrete* PN are restricted to be in the naturals, while in *continuous* PN are relaxed into the non-negative real numbers [DA10; Sil+11]. Thus, a continuous PN system is understood as a relaxation of a discrete one.

Definition 4.1 *A continuous PN system is a tuple* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ *where* $\mathcal{N}$ *is the structure and* $\boldsymbol{m}_0 \in \mathbb{R}^{|P|}_{\geq 0}$ *is the initial marking.*

In continuous systems, a transition t is *enabled* at $\boldsymbol{m}$ if for every $p \in {}^\bullet t$, $m[p] > 0$. It can be fired in any amount $\alpha \in \mathbb{R}$ such that $0 < \alpha \leq enab(t, \boldsymbol{m})$. The *enabling degree* of a transition t at $\boldsymbol{m}$ measures the maximal amount in which the transition can be fired in a single occurrence, i.e., $enab(t, \boldsymbol{m}) = \min_{p \in {}^\bullet t}\{\frac{m[p]}{Pre[p,t]}\}$.

For example, considering the PN in Fig. 4.1(a) with the initial marking $\boldsymbol{m}_0 = (1, 1)$, transition t_1 can be fired an amount α from $\boldsymbol{m}_0$, with $0 < \alpha \leq 0.5$. Moreover, transition t_2 can be fired an amount α from $\boldsymbol{m}_0$, with $0 < \alpha \leq 1$.

Analogous to discrete PN systems, the firing of t in a certain amount α leads to a new marking $\boldsymbol{m}'$, and it is denoted as $\boldsymbol{m} \xrightarrow{\alpha t} \boldsymbol{m}'$. It holds $\boldsymbol{m}' = \boldsymbol{m} + \alpha \cdot \boldsymbol{C}[P, t]$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the token flow matrix (incidence matrix if $\mathcal{N}$ is self-loop free) and $\boldsymbol{C}[P, t]$ denotes the column t of the matrix $\boldsymbol{C}$. The state (or fundamental) equation, $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, summarizes the way the marking evolves; where $\boldsymbol{\sigma}$ is the firing count vector associated to the fired sequence σ.

4.2 Reachability

In continuous PN systems, two sets of reachable markings are considered: one denoted as $RS_C(\mathcal{N}, \boldsymbol{m}_0)$, that contains all the markings that are reachable with *finite* firing sequences.

Definition 4.2 $RS_C(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} \mid \exists \sigma = \alpha_1 t_{\gamma_1} \ldots \alpha_k t_{\gamma_k}$ *s.t.* $\boldsymbol{m}_0 \xrightarrow{\alpha_1 t_{\gamma_1}} \boldsymbol{m}_1 \xrightarrow{\alpha_2 t_{\gamma_2}} \boldsymbol{m}_2 \cdots \xrightarrow{\alpha_k t_{\gamma_k}} \boldsymbol{m}_k = \boldsymbol{m}$ *where* $\alpha_i \in \mathbb{R}_{>0}$, $\forall i \in \{1..k\}\}$.

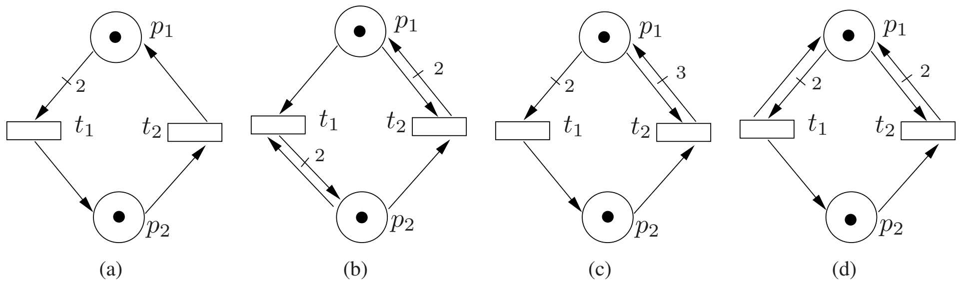


Figure 4.1: Four small PN systems with different behaviour as discrete and as continuous [RTS99].

An interesting property of the RS of CPN (different from the discrete RS) is that it is a convex set [RTS99]:

Definition 4.3 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a CPN system. Then, the set $RS_C(\mathcal{N}, \boldsymbol{m}_0)$ is convex, that is, if two markings $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ are reachable, then for any α s.t. $0 \leq \alpha \leq 1$, $\alpha \boldsymbol{m}_1 + (1 - \alpha)\boldsymbol{m}_2$ is also a reachable marking.*

Consider the PN system in Fig. 4.1(a). Both as discrete or as continuous, transition t_2 can be fired from $\boldsymbol{m}_0$. It can be fired an amount equal to 1: $\boldsymbol{m}_0 = (1,1) \xrightarrow{1t_2} \boldsymbol{m}_1 = (2,0)$. And then, transition t_1 can be fired: $\boldsymbol{m}_1 = (2,0) \xrightarrow{1t_1} \boldsymbol{m}_2 = (0,1)$, and transition t_2 can be fired again: $\boldsymbol{m}_2 = (0,1) \xrightarrow{1t_2} \boldsymbol{m}_3 = (1,0)$. Marking $\boldsymbol{m}_3$ is a deadlock in the discrete system. However, considered as continuous, transition t_1 can be fired in any amount α s.t. $0 < \alpha \leq enab(t_1, \boldsymbol{m}_3)$, i.e., between 0 and 0.5. It can be fired in the maximal possible amount as: $\boldsymbol{m}_3 = (1,0) \xrightarrow{0.5t_1} \boldsymbol{m}_4 = (0, 0.5)$. Transitions t_2 and t_1 can be successively fired from $\boldsymbol{m}_4$ as follows: $\boldsymbol{m}_4 = (0, 0.5) \xrightarrow{0.5t_2} \boldsymbol{m}_5 = (0.5, 0) \xrightarrow{0.25t_1} \boldsymbol{m}_6 = (0, 0.25) \xrightarrow{0.25t_2} \ldots$ The marking is divided by 2 each time that t_2 and t_1 are fired. However, both places are never emptied by a finite firing sequence.

However, considering an infinitely long firing sequence, the marking will approach $\boldsymbol{m}_d = (0,0)$, which is said to be reachable in the limit. The *lim-reachability* set, denoted as lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$, contains all the markings that are reachable either with a finite or with an *infinite* firing sequence.

Definition 4.4 lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} \mid \exists \sigma = \alpha_1 t_{\gamma_1} ... \alpha_i t_{\gamma_i} ...$ *s.t.* $\boldsymbol{m}_0 \xrightarrow{\alpha_1 t_{\gamma_1}} \boldsymbol{m}_1 \xrightarrow{\alpha_2 t_{\gamma_2}} \boldsymbol{m}_2 \ldots \boldsymbol{m}_{i-1} \xrightarrow{\alpha_i t_{\gamma_i}} \boldsymbol{m}_i \ldots$ *and* $lim_{i \to \infty} \boldsymbol{m}_i = \boldsymbol{m}$ *where* $\alpha_i \in \mathbb{R}_{>0}, \forall i > 0\}$.

In order to analyse and study the behaviour of systems, it is interesting to consider this lim-reachable marking, because it is the one to which the state of the system may converge. For any continuous system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$, the differences between $RS_C(\mathcal{N}, \boldsymbol{m}_0)$ and lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$ are just in the border points of their convex spaces. In fact, it holds that

$RS_C(\mathcal{N}, \boldsymbol{m}_0) \subseteq lim\text{-}RS_C(\mathcal{N}, \boldsymbol{m}_0)$. The closure of $RS_C(\mathcal{N}, \boldsymbol{m}_0)$ that is, all the points in $RS_C(\mathcal{N}, \boldsymbol{m}_0)$ plus the limit points of $RS_C(\mathcal{N}, \boldsymbol{m}_0)$, is equal to the closure of lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$ [JRS03].

Given that the firing amounts in a continuous system can be natural numbers, it trivially holds that $\text{RS}_D(\mathcal{N}, \boldsymbol{m}_0) \subseteq \text{RS}_C(\mathcal{N}, \boldsymbol{m}_0)$.

An immediate consequence of the definition of continuous firings is the following homothetic property [RTS99]:

Proposition 4.5 *If $\boldsymbol{m} \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$ then $\alpha \cdot \boldsymbol{m} \in RS_C(\mathcal{N}, \alpha \cdot \boldsymbol{m}_0)$, $\forall \alpha \in \mathbb{R}_{>0}$.*

Given two systems with the same set of places, the reachability set inclusion property can be defined.

Definition 4.6 ***reachability set inclusion***
Given systems $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ and $\langle \mathcal{N}', \boldsymbol{m}'_0 \rangle_C$ with $P = P'$, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is (lim-)reachable included in $\langle \mathcal{N}', \boldsymbol{m}'_0 \rangle_C$ if $RS_C(\mathcal{N}, \boldsymbol{m}_0) \subseteq RS_C(\mathcal{N}', \boldsymbol{m}'_0)$.

4.3 System properties

The properties which are recalled in Section 2.1.3 for discrete PN systems are redefined here for untimed continuous PN.

4.3.1 Synchronic properties

In contrast to the discrete PN systems, in which behavioural and structural synchronic properties are independent, in continuous PN systems both coincide over general conditions:

Property 4.7 *[SR02] Given a continuous PN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$, if every transition is fireable from $\boldsymbol{m}_0$ (i.e., if there does not exist empty siphon in $\boldsymbol{m}_0$) then behavioural and structural synchronic properties coincide.*

This result can be explained from the properties of continuous firings. According to the definition, any enabled transition in a continuous system can be fired in a sufficiently small quantity such that it does not become disabled. If every transition is fireable (equivalent to “there does not exist empty siphon in $\boldsymbol{m}_0$”, which can be verified in polynomial time [STC98]), then a strictly positive marking $\boldsymbol{m} > \boldsymbol{0}$ can be reached. From $\boldsymbol{m}$, the realizability of T-semiflows in isolation can be deduced [RTS99], therefore behavioural and structural *synchronic relations* [Sil87; SC88] coincide.

As an example, consider again the PN in Fig. 2.2 shown in Chapter 2. If the net system is considered as continuous, then enabled transitions can be fired in a small amount until a strictly positive marking is reached, for instance, by firing the sequence $\sigma = 0.4t_1 0.3t_6 0.2t_2 0.1t_3$, the system reaches a strictly positive marking $\boldsymbol{m}_1 = (0.6 0.2 0.1 0.1 0.1 0.1)$. From this marking $\boldsymbol{m}_1$, each of the two T-semiflows can be independently fired: for example, $\sigma_1 = 0.6t_1 0.6t_2 0.6t_3 0.6t_4$ and $\sigma_2 = 0.1t_5 0.1t_6$.

Definition 4.8 ***(lim-)boundedness (B).***

- *A place p is (lim-)bounded if $\exists b \in \mathbb{R}_{>0}$ such that for all $\boldsymbol{m} \in$ (lim-) $RS_C(\mathcal{N}, \boldsymbol{m}_0)$, $m[p] \leq b$.*
- *A system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is (lim-)bounded if every $p \in P$ is (lim-)bounded.*

The four PN examples in Fig. 4.1 are bounded and lim-bounded. For example, in the PN in Fig. 4.1(a), the bound of p_1 is $b = 2$, and the bound of p_2 is equal to $b = 1$.

Definition 4.9 ***(lim-)B-fairness (BF).***

- *Two transitions t, t' are in (lim-)B-fair relation if $\exists b \in \mathbb{R}_{>0}$ such that for all $\boldsymbol{m} \in$ (lim-)$RS_C(\mathcal{N}, \boldsymbol{m}_0)$, for every finite (infinite) firing sequence σ fireable from $\boldsymbol{m}$, it holds that if $\boldsymbol{\sigma}(t) = 0$ then $\boldsymbol{\sigma}(t') \leq b$; and if $\boldsymbol{\sigma}(t') = 0$ then $\boldsymbol{\sigma}(t) \leq b$.*
- *A system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is (lim-)B-fair if every pair of transitions $t, t' \in T$ is in (lim-)B-fair relation.*

The PN in Fig. 4.1(a) is B-fair, because the transitions cannot be fired in isolation.

A net $\mathcal{N}$ is *structurally* bounded (resp. *structurally* B-fair) if $\forall \boldsymbol{m}_0$, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is bounded (resp. B-fair).

4.3.2 Deadlock-freeness and liveness properties

The properties defined for DPN systems are redefined for CPN in this section, both considering reachability and $\lim$-reachability.

Definition 4.10 ***(lim-)deadlock-freeness (DF).*** *A system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is (lim-)deadlock-free if $\forall \boldsymbol{m} \in$ (lim-)$RS_C(\mathcal{N}, \boldsymbol{m}_0)$, $\exists t \in T$ such that t is enabled at $\boldsymbol{m}$.*

Definition 4.11 ***(lim-)liveness (L).*** *A system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is (lim-)live if for every transition t and for every marking $\boldsymbol{m} \in$ (lim-)$RS_C(\mathcal{N}, \boldsymbol{m}_0)$ there exists $\boldsymbol{m}' \in$ (lim-)$RS_C(\mathcal{N}, \boldsymbol{m})$ such that t is enabled at $\boldsymbol{m}'$.*

As said, deadlock-freeness and liveness properties of a discrete system are not always preserved by its continuous counterpart, as pointed out in [RTS99] through the examples in Fig. 4.1, in which deadlock-freeness and liveness coincide.

Their behaviour can be summarized as follows:

- Fig. 4.1(a) deadlocks as DPN. It is deadlock-free as CPN. It is not lim-deadlock-free as CPN.
- Fig. 4.1(b) deadlocks as DPN. It deadlocks as CPN. It is not lim-deadlock-free as CPN.
- Fig. 4.1(c) is deadlock-free as DPN. It deadlocks as CPN. It also reaches a deadlock considering reachability in the limit (lim-deadlock) as CPN.

- Fig. 4.1(d) is deadlock-free as DPN. It is deadlock-free as CPN. It is not lim-deadlock-free as CPN.

Home state and reversibility are defined below as those for DPN.

Definition 4.12 ***home state***. *A marking $\boldsymbol{m}_h$ is a (lim-) home state in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ if $\forall \boldsymbol{m}' \in$ $(lim\text{-})RS_C(\mathcal{N}, \boldsymbol{m}_0)$,*
$\boldsymbol{m}_h \in (lim\text{-})RS_C(\mathcal{N}, \boldsymbol{m}')$.

When $\boldsymbol{m}_0$ is a home state, it is said that $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is reversible:

Definition 4.13 ***(lim-)reversibility (R)***. *A system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is (lim-) reversible if for any marking $\boldsymbol{m} \in (lim\text{-})RS_C(\mathcal{N}, \boldsymbol{m}_0)$ it holds that $\boldsymbol{m}_0 \in (lim\text{-})RS_C(\mathcal{N}, \boldsymbol{m})$.*

Due to the fact that $RS_C(\mathcal{N}, \boldsymbol{m}_0) \subseteq$ lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$, a direct implication is that if $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is lim-bounded (resp. lim-B-fair, lim-deadlockfree), then it is also bounded (resp. B-fair, resp. deadlock-free) [RTS99].

A net $\mathcal{N}$ is *structurally* deadlock-free (resp. *structurally* live; resp. *structurally* reversible) if $\exists \boldsymbol{m}_0$ s.t. $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is deadlock-free (resp. live; resp. reversible).

All these properties about CPN are deeper investigated in the following chapters. First, the preservation of boundedness, deadlock-freeness, liveness and reachability of the discrete system by its continuous counterpart is studied in Chapter 5. Then, the CPN formalism itself is considered in Chapter 6, where the characterization and complexity bounds of checking these properties are recalled from the literature in some cases and it is developed in some others.

5

Preservation of homothetic properties by continuous Petri nets

“In isosceles triangles the angles at the base equal one another, and, if the equal straight lines are produced further, then the angles under the base equal one another”.
Thales of Miletus

Fluidization of Petri nets is the result of removing the integrality constraint in the firing of transitions. This relaxation may highly reduce the complexity of its analysis techniques but may not preserve important properties of the original system. This chapter aims at establishing conditions that an untimed discrete system must satisfy so that a given property is preserved by the continuous relaxation. These conditions are mainly based on the *marking homothetic behaviour* of the system. The focus is on logical properties as boundedness, B-fairness, deadlock-freeness, liveness and reversibility. Furthermore, testing homothetic monotonicity of boundedness and deadlock-freeness in the discrete systems is also studied, as well as the existence of spurious deadlocks in some system subclasses.

Introduction

At first glance, the simple way in which the basic definitions of discrete models are extended to continuous ones may make us naively think that their behaviour will be similar. However, the behaviour of the continuous model can be completely different just because the integrality constraint has been dropped. In other words, not all DEDS can be satisfactorily fluidified. Consider, for instance, the net system in Fig. 5.1 (a). If considered as discrete, the system is deadlock-free: from $\boldsymbol{m}_0$ = (3, 0), both t_2 and t_1 can be fired alternatively, and no deadlock can be reached. However, if considered as continuous, transition t_2 can be fired in an amount of 1.5 from $\boldsymbol{m}_0$, leading to a deadlock marking $\boldsymbol{m}_d$ = (0, 1.5).

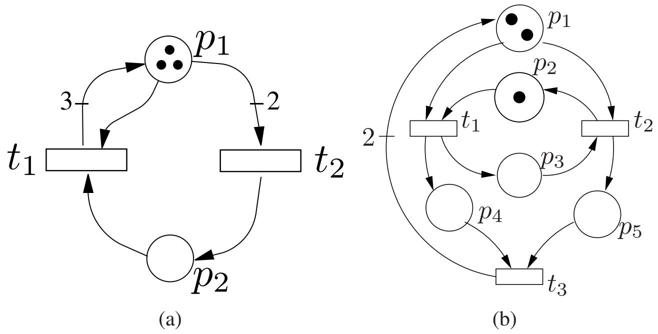


Figure 5.1: (a) Not homothetically deadlock-free PN system [RTS99] (b) homothetic deadlock-free PN system.

Notice that deadlock-freeness of the discrete system in Fig. 5.1 (a) highly depends on its initial marking. In fact, if the initial marking is doubled, i.e., if we consider $\boldsymbol{m}'_0$ = (6, 0), then the system deadlocks by firing t_2 an amount of 3.

Let us now consider the PN in Fig. 5.1 (b), which exhibits a different behaviour. Considered as discrete, it is deadlock-free for $\boldsymbol{m}_0$ = (2, 1, 0, 0, 0). Moreover, it is deadlock-free for any initial marking proportional to $\boldsymbol{m}_0$, i.e., $\boldsymbol{m}'_0 = k \cdot \boldsymbol{m}_0$, with $k \in \mathbb{N}$.

When the PN system is fluidified, i.e., the PN system in Fig. 5.1 (b) is considered as a continuous system, it preserves deadlock-freeness. We will exploit this idea to extract conditions for the preservation of properties.

The present work explores the kind of features that a discrete net system must exhibit so that a given property is preserved when it is fluidified. It focuses on classical properties as boundedness, B-fairness, deadlock-freeness, liveness and reversibility. The main ideas used here are: (a) the property of homothetic monotonicity of continuous firing sequences (needed for Lemma 5.4); (b) the fact that every real number could be approximated by a rational number (used in Lemma 5.5). Properties preservation is built over these two ideas. Furthermore, homothetic monotonicity of boundedness, B-fairness and deadlock-freeness properties in discrete Petri nets are studied, as well as property preservation for some subclasses. Some techniques to improve the fluidization are also considered, where the spurious deadlocks are *removed* with the addition of some *implicit* places.

This chapter is organized as follows. Section 5.1 introduces some concepts that will be used in the rest of the chapter, such as monotonicity and homothetic monotonicity. Section 5.2 sets the main results concerning homothetic properties in a discrete net system and its relations with the fluid counterpart. In Section 5.3, some results about homothetic boundedness and homothetic B-fairness of discrete PN are presented. Finally, preservation of properties for different net (system) subclasses is studied in Section 5.4 and some conclusions are summarized in Section 5.5.

5.1 Preliminary concepts and definitions

Some concepts used in this chapter are defined here. First, let us introduce an additional reachability set to be used in this work, the *rational reachability set* ($RS_Q(\mathcal{N}, \boldsymbol{m}_0)$): the set of markings that can be reached from $\boldsymbol{m}_0$ considering only firings in the set of rational numbers ($\mathbb{Q}$).

We will denote $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_Q$ the net system in which only rational amounts are fired by the transitions.

Definition 5.1 *$RS_Q(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} \mid \exists\, \sigma = \alpha_1 t_{\gamma_1} \ldots \alpha_k t_{\gamma_k}$ s.t. $\boldsymbol{m}_0 \xrightarrow{\alpha_1 t_{\gamma_1}} \boldsymbol{m}_1 \cdots \xrightarrow{\alpha_k t_{\gamma_k}} \boldsymbol{m}_k = \boldsymbol{m}$ where $\alpha_i \in \mathbb{Q}_{>0}, \forall i \in \{1..k\}\}$.*

The most important concept which is considered in this chapter is *marking homothetic monotonicity*. Here, *marking monotonicity* and *marking homothetic monotonicity* are defined below for Π. Along this chapter, let Π represent one of the following properties defined in Chapters 2 and 4: boundedness (B), B-fairness (BF), deadlock-freeness (DF), liveness (L) or reversibility (R).

In this work, we limit ourselves to the use of those concepts with respect to the marking. I.e., we consider *marking monotonicity* and *marking homothetic monotonicity*, in which the initial marking of the PN system is scaled. However, *monotonicity* with respect to other properties of interest are considered in other works, such as performance monotonicity w.r.t. (with respect to) the firing rates in timed PN systems [MRS09].

Definition 5.2 ***Monotonicity***. *Given a system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$, a property Π is* monotonic *w.r.t. $\boldsymbol{m}_0$ if:*
Π holds in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D \Longrightarrow \Pi$ holds in $\langle \mathcal{N}, \boldsymbol{m}'_0 \rangle_D$ for every $\boldsymbol{m}'_0 \geq \boldsymbol{m}_0$.

For example, in the PN system in Fig. 5.2, DF is not monotonic for $k = 1$, because it is deadlock-free for $k = 1$, $k = 3$, and $k \geq 5$; but it deadlocks for $k = 2$ and for $k = 4$. Moreover, it is live for $k \geq 6$.

Definition 5.3 ***Homothetic monotonicity***. *Given a system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$, a property Π is* homothetically monotonic *(for short,* homothetic*) w.r.t. $\boldsymbol{m}_0$ if:*
Π holds in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D \Longrightarrow \Pi$ holds in $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$, $\forall k \in \mathbb{N}^+$.

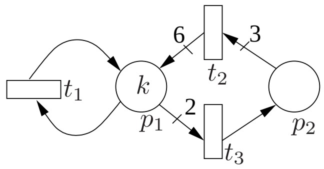


Figure 5.2: Non monotonic deadlock-free PN system

Homothetic monotonicity of DF can be illustrated with the example in Fig. 5.1 (b). The discrete net system is DF for $\boldsymbol{m}_0 = (2, 1, 0, 0, 0)$, and for any proportional initial marking $k \cdot \boldsymbol{m}_0$, i.e., it is homothetically DF. Nevertheless, the system is not *monotonically* DF for $\boldsymbol{m}_0$, for example, for $\boldsymbol{m}'_0 = (2, 2, 0, 0, 0)$ it deadlocks, where $\boldsymbol{m}'_0 \geq \boldsymbol{m}_0$.

Notice that monotonicity is more restrictive than homothetic monotonicity, i.e., if Π is monotonic then Π is also homothetically monotonic. Some classical results on studying monotonicity of certain properties such as liveness are the rank theorems [STC98], which give necessary or sufficient conditions from the structure of the net (with polynomial complexity), or the siphon-trap property [Bra83], which gives a necessary and sufficient condition for the behavioural property (with higher complexity).

5.2 Homothetic monotonicity and property preservation by fluidization

The aim of this section is to set certain conditions that a discrete PN system has to fulfil to preserve a certain property after being fluidified to a continuous PN system. It will be proved that, given a property Π which exhibits homothetic monotonicity in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$, Π is preserved by fluidization (i.e. in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$). We focus on the well-known properties defined in Chapter 4. First, two technical results (Lemmas 5.4 and 5.5) about reachability are presented.

5.2.1 Reachability

Based on the definition of $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_Q$ provided in the previous section, the following lemma states that for any marking $\boldsymbol{m}$ reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_Q$, there exists a $k \in \mathbb{N}$ such that a scaled marking $k \cdot \boldsymbol{m}$ is reachable in $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$.

Lemma 5.4 *Given a PN structure* $\mathcal{N}$ *and an initial marking* $\boldsymbol{m}_0 \in \mathbb{N}$, $\boldsymbol{m} \in RS_Q(\mathcal{N}, \boldsymbol{m}_0)$ $\Longrightarrow \exists k \in \mathbb{N}$ *such that* $k \cdot \boldsymbol{m} \in RS_D(\mathcal{N}, k \cdot \boldsymbol{m}_0)$.

Proof. Let us suppose $\boldsymbol{m} \in RS_Q(\mathcal{N}, \boldsymbol{m}_0)$, i.e., $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$, where $\sigma = \alpha_1 t_{\gamma_1} \cdots \alpha_n t_{\gamma_n}$, and $\alpha_i \in \mathbb{Q}$, $\forall i \in \{1 \cdots n\}$. Because each α_i is a rational amount, it can be considered as its irreducible fraction: $\alpha_i = \frac{n_i}{d_i}$.

We can multiply the rational sequence σ by the l.c.m. (least common multiple) of the denominators of the irreducible fractions, to obtain a sequence σ' in the naturals: $\sigma' = k \cdot \sigma$, where $k = l.c.m.(d_i \mid \frac{n_i}{d_i} = \alpha_i, \forall \alpha_i \in \sigma)$.
It holds $\alpha'_i \in \mathbb{N}$ for every α'_i in σ'. Because of the properties of the continuous PN (see Proposition 4.5), the initial marking ($\boldsymbol{m}_0$), the firing sequence (σ) and the resulting marking ($\boldsymbol{m}$) can be scaled by k in the continuous PN: $k \cdot \boldsymbol{m}_0 \xrightarrow{k \cdot \sigma} k \cdot \boldsymbol{m}$. Because it is a natural sequence fireable in the continuous PN, $\sigma' = k \cdot \sigma$ is also fireable from $k \cdot \boldsymbol{m}_0$ in the discrete system: $k \cdot \boldsymbol{m}_0 \xrightarrow{\sigma'} k \cdot \boldsymbol{m}$. ■

Now it is proved that, for any marking $\boldsymbol{m}$ reachable with a *real* firing sequence, another marking $\boldsymbol{m}'$ exists that is reachable with *rational* firings, such that it is as close to $\boldsymbol{m}$ as desired, and the set of empty places coincide.

Lemma 5.5 *For every $\sigma = \alpha_1 t_{\gamma_1} \ldots \alpha_i t_{\gamma_i}$, with $\alpha_j \in \mathbb{R}_{>0}$, $j \in 1..i$, s.t. $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$, with $\boldsymbol{m}_0 \in \mathbb{N}^{|P|}$, $\boldsymbol{m} \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$, and every ε, $\varepsilon' > 0$, there exists $\sigma' = \alpha'_1 t_{\gamma_1} \ldots \alpha'_i t_{\gamma_i}$ s.t. $\boldsymbol{m}_0 \xrightarrow{\sigma'} \boldsymbol{m}'$, with $\boldsymbol{m}' \in RS_Q(\mathcal{N}, \boldsymbol{m}_0)$ such that:*

- $||\boldsymbol{m}' - \boldsymbol{m}|| < \varepsilon$ *and*
- $m'[p] = 0 \Leftrightarrow m[p] = 0$
- $\forall j \leq i$, $|\alpha'_j - \alpha_j| < \varepsilon'$

Proof. Given $\sigma = \alpha_1 t_{\gamma_1} \ldots \alpha_i t_{\gamma_i}$, with $\alpha_j \in \mathbb{R}_{>0}, \forall j \in \{1..i\}$ such that $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$; then for any $\varepsilon, \varepsilon' > 0$, we will build the firing sequence $\sigma' = \alpha'_1 t_{\gamma_1} \ldots \alpha'_i t_{\gamma_i}$, $\alpha'_j \in \mathbb{Q}$, $\forall j \in \{1..i\}$, such that $\boldsymbol{m}_0 \xrightarrow{\sigma'} \boldsymbol{m}'$. Where $||\boldsymbol{m}' - \boldsymbol{m}|| < \varepsilon$, $|\alpha'_j - \alpha_j| < \varepsilon'$ and $(m'[p] = 0 \Leftrightarrow m[p] = 0)$. It will be proved by induction on the length of the sequence σ: $|\sigma| = i$.

- Base case ($|\sigma| = 1$).
  Let $\sigma = \alpha_1 t_{\gamma_1}$. Then, a $\alpha'_1 \in \mathbb{Q}$, has to be chosen. The firing of $\alpha_1 t_{\gamma_1}$ yields $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C}[P, t_{\gamma_1}]\alpha_1$; and the firing of a given $\alpha'_1 t_{\gamma_1}$ yields $\boldsymbol{m}' = \boldsymbol{m}_0 + \boldsymbol{C}[P, t_{\gamma_1}]\alpha'_1$. Subtracting both equations and considering its norm, we obtain $||\boldsymbol{m}' - \boldsymbol{m}||$ = $||\boldsymbol{C}[P, t_{\gamma_1}]$ $(\alpha'_1 - \alpha_1)||$. Since all the elements in $\boldsymbol{C}$ are finite numbers, a rational $\alpha'_1 \in \mathbb{Q}$ close enough to α_1 can be chosen to satisfy $||\boldsymbol{m}' - \boldsymbol{m}|| < \varepsilon$ and $|\alpha'_1 - \alpha_1| < \varepsilon'$. Moreover, since $\boldsymbol{m}_0 \in \mathbb{N}$, if the firing of α_1 emptied some places, then $\alpha_1 \in \mathbb{Q}$ and $\alpha'_1 = \alpha_1$ can be chosen. Otherwise (if no place has been emptied), then $\alpha'_1 \in \mathbb{Q}$ as close as desired to α_1 can be chosen that does not empty places.

- Inductive hypothesis ($|\sigma| = i$)
  Given $\sigma = \alpha_1 t_{\gamma_1} \ldots \alpha_i t_{\gamma_i}$, such that $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}_i$; there exists $\sigma' = \alpha'_1 t_{\gamma_1} \cdots \alpha'_i t_{\gamma_i}$, such that $\alpha'_j \in \mathbb{Q}, \forall j \in \{1..i\}$ and $\boldsymbol{m}_0 \xrightarrow{\sigma'} \boldsymbol{m}'_i$, where $||\boldsymbol{m}'_i - \boldsymbol{m}_i|| < \varepsilon$, $|\alpha'_i - \alpha_i| < \varepsilon'$ and $(m'_i[p] = 0 \Leftrightarrow m[p] = 0)$.

- Inductive step ($|\sigma| = i + 1$)
  Let consider the $i + 1$ firing. We can distinguish two cases:

  (a) The firing of $\alpha_{i+1} t_{\gamma_{i+1}}$ does not empty places in $^{\bullet}t_{\gamma_{i+1}}$. Then, it holds that $\boldsymbol{m} = \boldsymbol{m}_i + \boldsymbol{C}[P, t_{\gamma_{i+1}}]\alpha_{i+1}$, and $\boldsymbol{m}' = \boldsymbol{m}'_i + \boldsymbol{C}[P, t_{\gamma_{i+1}}]\alpha'_{i+1}$. Again, subtracting both equations and considering its norm, we obtain $||\boldsymbol{m}' - \boldsymbol{m}|| = ||(\boldsymbol{m}'_i - \boldsymbol{m}_i) + \boldsymbol{C}[P, t_{\gamma_{i+1}}] (\alpha'_{i+1} - \alpha_{i+1})||$.

  We have to force that $||\boldsymbol{m} - \boldsymbol{m}'|| < \varepsilon$ and $|\alpha'_{i+1} - \alpha_{i+1}| < \varepsilon'$. Given that $\boldsymbol{m}_i$ and $\boldsymbol{m}'_i$ fulfil the inductive hypothesis, the quantity $||\boldsymbol{m}'_i - \boldsymbol{m}_i||$ can be as small as desired. Moreover, since the elements of the matrix $\boldsymbol{C}$ are finite numbers, a rational α'_{i+1} close to α_{i+1} can be chosen such that $||\boldsymbol{C}[P, t_{\gamma_{i+1}}](\alpha'_{i+1} - \alpha_{i+1})||$ is as small as desired and no places in $^{\bullet}t_{\gamma_{i+1}}$ are emptied.

  (b) The firing of $\alpha_{i+1} t_{\gamma_{i+1}}$ empties places in $^{\bullet}t_{\gamma_{i+1}}$. Then, $\alpha'_{i+1} = enab(t_{\gamma_{i+1}}, \boldsymbol{m}'_i)$ is chosen, in order to empty the same input places. The amount α'_{i+1} is in $\mathbb{Q}$, because $\boldsymbol{m}'_i$ (and hence the enabling degree) is rational. Since $\boldsymbol{m}_i$ and $\boldsymbol{m}'_i$ fulfil the inductive hypothesis, they can be as close as desired. Thus, the firing of α'_{i+1} empties the same places than α_{i+1}, and $\boldsymbol{m}_{i+1}$ and $\boldsymbol{m}'_{i+1}$ can be as close as desired, as well as α_{i+1} and α'_{i+1}.

■

Lemmas 5.4 and 5.5, will help to prove some properties related to the preservation of B, BF; and DF, L, and R.

5.2.2 Synchronic properties: boundedness and B-fairness

Some properties are included in the general concept of synchronic properties [SC88], which are considered here. For a continuous PN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ in which every transition can be fired (i.e., there are no empty siphons at $\boldsymbol{m}_0$), *behavioural* and *structural synchronic relations* coincide, as noticed in [SR02]. Moreover, here it is proved (Propositions 5.6 and 5.7) that in any PN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$, a synchronic property Π_S (boundedness or B-fairness) is equivalent in the homothetic discrete system $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$ and in the continuous PN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$. The corresponding properties in the limit (i.e., with infinite sequences) are considered in Section 5.3. The proof is done for the two synchronic properties defined in Section 4.3.

Proposition 5.6 *$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically bounded $\Longleftrightarrow$ $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is bounded.*

Proof. ($\Longrightarrow$) Let us suppose the $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is unbounded, i.e., $\forall b \in \mathbb{R}^+ \; \exists p \in P \; \exists \boldsymbol{m} \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$ s.t. $m[p] > b$. If $\boldsymbol{m}$ is not in $RS_Q(\mathcal{N}, \boldsymbol{m}_0)$, but $m[p] > b$, because of Lemma 5.5, $\forall \varepsilon > 0$, $\exists \boldsymbol{m}'$ s.t. $||\boldsymbol{m}' - \boldsymbol{m}|| < \varepsilon$, so we can find another $\boldsymbol{m}' \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$ as near to $\boldsymbol{m}$ as desired, such that also $m'[p] > b$. And because of Lemma 5.4, if $\boldsymbol{m}' \in RS_Q(\mathcal{N}, \boldsymbol{m}_0)$, then $\exists k \in \mathbb{N}$ s.t. $k \cdot \boldsymbol{m}' \in RS_D(\mathcal{N}, k \cdot \boldsymbol{m}_0)$. Hence, in the discrete PN, $\forall b \in \mathbb{R}^+$, $\exists k \cdot \boldsymbol{m} \in RS_D(\mathcal{N}, k \cdot \boldsymbol{m}_0)$ s.t. $k \cdot m[p] > b$. Consequently, $\exists k \in \mathbb{N}$ s.t. the discrete system $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$ is unbounded.

($\Longleftarrow$) Let us suppose $\exists k \in \mathbb{N}$ s.t. the discrete system $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$ is unbounded. It means $\forall b \in \mathbb{R}^+$, $\exists p \in P\ \exists \boldsymbol{m} \in RS_D(\mathcal{N}, k \cdot \boldsymbol{m}_0)$ s.t. $b < m[p]$. If $\boldsymbol{m} \in RS_D(\mathcal{N}, k \cdot \boldsymbol{m}_0)$, then also $\boldsymbol{m} \in RS_C(\mathcal{N}, k \cdot \boldsymbol{m}_0)$. Because of Property 4.5, for each marking $\boldsymbol{m} \in RS_C(\mathcal{N}, k \cdot \boldsymbol{m}_0)$, the marking $\boldsymbol{m}' = \frac{\boldsymbol{m}}{k}$ is reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$. For every $c \in \mathbb{R}^+$ s.t. it holds, also for $c \cdot k$ it holds (since it holds for every real): $\exists p \in P\ \exists \boldsymbol{m}\ \in RS_D(\mathcal{N}, k \cdot \boldsymbol{m}_0)$ s.t. $m[p] > c \cdot k$. Consequently, for every real c, it holds that $\exists p \in P$ s.t. $m[p] > c \cdot k$. And it implies $m'[p] = \frac{m[p]}{k} > c$, where $m'[p] \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$. Hence, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is unbounded. ■

Using similar arguments, the following result considers B-fairness.

Proposition 5.7 *$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically B-fair $\Longleftrightarrow$ $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is B-fair.*

Proof. ($\Longrightarrow$) Let us suppose $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is not B-fair, then $\exists \boldsymbol{m} \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$, $\exists t, t'$ s.t. $\forall b \in \mathbb{R}^+$, $\exists \sigma$ fireable from $\boldsymbol{m}$ s.t. $\boldsymbol{\sigma}(t) = 0$ and $\boldsymbol{\sigma}(t') > b$. By Lemma 5.5, if $\boldsymbol{m}$ (or σ) is not in $\mathbb{Q}$, then there exists another $\boldsymbol{m}'$ (or other σ') in $\mathbb{Q}$ with the properties shown in the lemma. Then, by Lemma 5.4, $\exists k \in \mathbb{N}$ s.t. $k \cdot \boldsymbol{m}' \in RS_D(\mathcal{N}, k \cdot \boldsymbol{m}_0)$. And, by applying again Lemma 5.4, from $k \cdot \boldsymbol{m}'$ it is also possible to fire $\sigma'' = k \cdot \sigma'$ such that $\boldsymbol{\sigma}''(t) = 0$ and it makes $\boldsymbol{\sigma}''(t') > k \cdot b \geq b$, $\forall b \in \mathbb{R}^+$. Consequently, $\exists k \in \mathbb{N}$ s.t. the discrete system $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$ is not B-fair.

($\Longleftarrow$) Let us suppose $\exists k \in \mathbb{N}$ s.t. the discrete system $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$ is not B-fair. It means $\exists \boldsymbol{m} \in RS_D(\mathcal{N}, k \cdot \boldsymbol{m}_0)$, $\exists t, t'$ s.t. $\forall b \in \mathbb{R}^+$, $\exists \sigma$ fireable from $\boldsymbol{m}$ s.t. $\boldsymbol{\sigma}(t) = 0$ and $\boldsymbol{\sigma}(t') > b$. Due to the fact that $\boldsymbol{m} \in RS_D(\mathcal{N}, k \cdot \boldsymbol{m}_0)$, then a marking $\boldsymbol{m}' = \frac{1}{k}\boldsymbol{m}$ is reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$, from which $\sigma' = \frac{\sigma}{k}$ can be fired. It holds that $\boldsymbol{\sigma}'(t) = \frac{\boldsymbol{\sigma}(t)}{k} = 0$ and $\boldsymbol{\sigma}'(t') = \frac{\boldsymbol{\sigma}(t')}{k} > \frac{b}{k}$. This reasoning can be done for every $b \in \mathbb{R}^+$. Hence, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is not B-fair either. ■

5.2.3 Deadlock-freeness

Some results about the preservation of a homothetic property Π_L (deadlock-freeness, liveness and reversibility) when the system is fluidified are presented in the sequel. The results and their proofs are analogous for deadlock-freeness, liveness and reversibility. For a didactic purpose, in this section the results are explained for DF, and in the following section they are extended to liveness and reversibility.

As previously defined, DF in continuous PN only considers the markings that are reachable with *finite* firing sequences (reachability); while lim-DF considers also *infinite* firing sequences (lim-reachability). Both concepts will be considered here.

A technical result is presented first. It sets that, given a reachable deadlock marking $\boldsymbol{m}_d$ (the subscript $_d$ denotes *deadlock*) in $RS_C(\mathcal{N}, \boldsymbol{m}_0)$, either its firing sequence is in $\mathbb{Q}$ (so it is in $RS_Q(\mathcal{N}, \boldsymbol{m}_0)$) or it is in $\mathbb{R} \setminus \mathbb{Q}$ and then there exists another "close" deadlock that is in $\mathbb{Q}$ (also in $RS_Q(\mathcal{N}, \boldsymbol{m}_0)$). In summary, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is DF if and only if $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_Q$ is DF.

Lemma 5.8 *$\boldsymbol{m}_d \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$ is a deadlock $\Longleftrightarrow$ $\exists \boldsymbol{m}'_d \in RS_Q(\mathcal{N}, \boldsymbol{m}_0)$ s.t. it is a deadlock.*

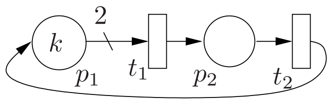


Figure 5.3: PN system which deadlocks as discrete for any k. It is deadlock-free as continuous, but it lim-deadlocks ($\boldsymbol{m}_d = (0,0)$ is lim-reachable).

Proof. ($\Longrightarrow$) Assume $\boldsymbol{m}_d \in RS_C(\mathcal{N}, \boldsymbol{m}_0) \smallsetminus RS_Q(\mathcal{N}, \boldsymbol{m}_0)$. Because of Lemma 5.5, $\forall \varepsilon > 0$ another $\boldsymbol{m}'_d \in RS_Q(\mathcal{N}, \boldsymbol{m}_0)$ exists such that $||\boldsymbol{m}'_d - \boldsymbol{m}_d|| < \varepsilon$ $\forall p$ s.t. $m_d[p] = 0$, also $m'_d[p] = 0$. Since $\forall t \in T$, t is not enabled in $\boldsymbol{m}_d$, then also $\forall t \in T$, t is not enabled in $\boldsymbol{m}'_d$. Hence, $\exists \boldsymbol{m}'_d \in RS_Q(\mathcal{N}, \boldsymbol{m}_0)$ that is a deadlock in the continuous system.

($\Longleftarrow$) It trivially holds: if $\exists \boldsymbol{m}_d \in RS_Q(\mathcal{N}, \boldsymbol{m}_0)$, then $\boldsymbol{m}_d$ is also reachable in $RS_C(\mathcal{N}, \boldsymbol{m}_0)$ and it is also a deadlock. ■

Let us now prove that, if a discrete PN is homothetically DF, it will also be DF as continuous.

Proposition 5.9 *$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically DF $\Longrightarrow$ $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is DF.*

Proof. Let us suppose $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ deadlocks. It means $\exists \boldsymbol{m} \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$ that is a deadlock. Because of Lemma 5.8, if $\boldsymbol{m}$ is a deadlock, then there exists $\boldsymbol{m}' \in RS_Q(\mathcal{N}, \boldsymbol{m}_0)$ that is a deadlock. Because of Lemma 5.4, $\exists k \in \mathbb{N}$ s. t. $\boldsymbol{m}'' = k \cdot \boldsymbol{m}'$, where $\boldsymbol{m}'' \in RS_D(\mathcal{N}, k \cdot \boldsymbol{m}_0)$. Since $\forall t \in T$, $\exists p \in {}^\bullet t, m'[p] = 0$, then also $\forall t \in T$, $\exists p \in {}^\bullet t, k \cdot \boldsymbol{m}''[p] = 0$, and consequently $\boldsymbol{m}''$ it is also deadlock: $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$ deadlocks. ■

Proposition 5.9 can be illustrated by the example in Fig. 5.1 (b). However, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is DF $\not\Longrightarrow$ $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically DF; as the PN in Fig. 5.3 shows. The net system is DF when considered as continuous (illustrated in [RTS99]); but $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$ deadlocks for every k when considered as discrete.

The previous results deal with DF. What happens if lim-DF is considered?

A continuous system which is lim-DF, could be homothetically DF as discrete. When this is not the case, a minimum value of k can be considered for homothetic monotonicity. We will denote that a property Π_L is *homothetic from* n if $\exists n \in \mathbb{N}$, s.t. $\forall k \geq n$, with $k \in \mathbb{N}$, Π_L holds in $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$. Now, the implication can be formulated (in "some sense" it is the inverse of Proposition 5.9).

Proposition 5.10 *$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is lim-DF $\Longrightarrow$ $\exists n \in \mathbb{N}$ s.t. $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically DF from n.*

Proof. Let us suppose $\forall n \in \mathbb{N}$ $\exists k \geq n$, $k \in \mathbb{N}$, such that the discrete system $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0 \rangle_D$ deadlocks. It means there exists an infinite ordered set $A = \{a_1, a_2, a_3 \dots\}$, such that $\forall a_i \in A$, $a_i < a_{i+1}$ and $\langle \mathcal{N}, a_i \cdot \boldsymbol{m}_0 \rangle_D$ deadlocks.

For each a_i for which it deadlocks, $\exists \boldsymbol{m}_d \in RS_D(\mathcal{N}, a_i \cdot \boldsymbol{m}_0)$ s.t. $\boldsymbol{m}_d$ is a deadlock. It holds that $\forall t \in T, \exists p \in {}^\bullet t, m_d[p] < Pre[p,t]$. Because of the definitions of continuous firings (Property 4.5), marking $\frac{\boldsymbol{m}_d}{a_i}$ is reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$.

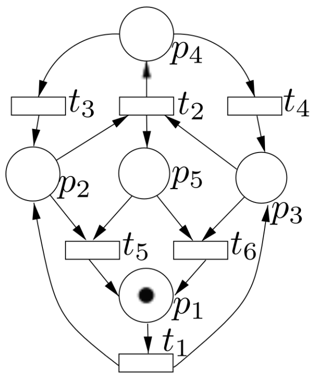


Figure 5.4: PN system which is live as discrete for any $k \cdot \boldsymbol{m}_0$, with $k \in \mathbb{N}^+$ (homothetically deadlock-free). It is deadlock-free as continuous, but not lim-deadlock-free ($\boldsymbol{m}_d = (0,0,0,0,2)$ is lim-reachable).

Given that a_i tends to infinite, making $a_i \rightarrow \infty$, then $\frac{m_d}{a_i}[p] \rightarrow 0$, and it will reach a deadlock in the limit. Consequently, the continuous $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is not lim-deadlockfree. ■

However, the reverse is not true (the stated Proposition 5.9 does not hold for lim-DF): $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically DF $\not\Longrightarrow$ $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is lim-DF. Given the net system in Fig. 5.4, it is homothetically DF for $\boldsymbol{m}_0$ =(1, 0, 0, 0, 0) considered as discrete. Even more, it can be easily proved that the discrete system is fully monotonic DF for $\boldsymbol{m}_0$ because every siphon contains a marked trap (see, for example, [Bra83]). However, when the net system is considered as continuous, the infinite firing sequence $\sigma = t_1\ t_2\ \frac{1}{2}t_3\ \frac{1}{2}t_4\ \frac{1}{2}t_2\ \frac{1}{4}t_3\ \frac{1}{4}t_4\ \frac{1}{4}t_2\ \frac{1}{8}t_3\ \frac{1}{8}t_4 \ldots$ can be fired, leading to the deadlock marking $\boldsymbol{m}_d = (0,0,0,0,2)$: the continuous system reaches a deadlock in the limit (as already noticed in [RTS99]).

Observe that $\boldsymbol{m}_d$ empties the trap and siphon $\{p_1, p_2, p_3, p_4\}$. Emptying a trap in a continuous net system can only be done considering an *infinitely* long firing sequence. A trap cannot be emptied in a discrete system, thus $\boldsymbol{m}_d$ is a *spurious* solution in the discrete net system; it is a deadlock, hence it is a *killing spurious* solution.

5.2.4 Liveness and reversibility

The lemmas and properties presented here are analogous to the ones presented for DF (Section 5.2.3); even the proofs are technically analogous. Figure 5.5 summarizes the relations among a certain property Π_L (DF, L or R), when considered in the discrete system (homothetically Π_L), in the continuous system (Π_L) and in the limit (lim-Π_L).

Analogously to Lemma 5.8:

Lemma 5.11 $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ *is* $\Pi_L \Longleftrightarrow \langle \mathcal{N}, \boldsymbol{m}_0 \rangle_Q$ *is* Π_L.

The proof of this lemma when Π_L = L is similar to the one of Lemma 5.8, but instead of considering a marking $\boldsymbol{m}_d$ which is a deadlock, a marking $\boldsymbol{m}_l$ from which $\exists t$ s.t. t cannot

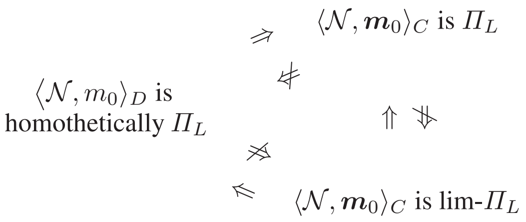


Figure 5.5: Relations w.r.t. a property $\Pi_L \in \{DF, L, R\}$.

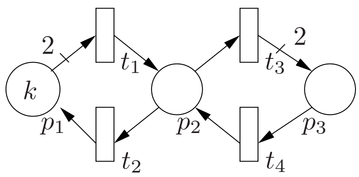


Figure 5.6: If $k \geq 2$, non reversible, non live discrete PN system. It is reversible and live as continuous.

be enabled from $\boldsymbol{m}_l$ must be considered. In the lemma when Π_L = R, a marking $\boldsymbol{m}_r$ from which the initial marking is not reachable must be considered.

Given these lemmas, a general result about preservation of a homothetic property by fluidization can be formulated (similar to Proposition 5.9).

Theorem 5.12 *$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically $\Pi_L \Longrightarrow \langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is Π_L.*

The proofs of the theorem for liveness (Π_L = L) and for reversibility (Π_L = R) would be similar to the proof of Proposition 5.9. The lemmas obtained from Lemma 5.11 for Π_L = L and Π_L = R would be also used in these proofs.

As an illustrative example, consider the Petri net example in Fig. 5.1 (b). It is homothetically live and homothetically reversible. Thus it preserves these properties when fluidified.

If the opposite implication is considered, again, even in the case of considering not *every* *k* but a *big enough k*, the implication is not true:

$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is $\Pi_L \not\Longrightarrow \exists n \in \mathbb{N}$, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically Π_L from n. Let us consider the example in Fig. 5.6. As continuous, it is live and reversible for $\boldsymbol{m}_0$ = (1, 0, 0): the marking can decrease by firing t_1 and t_2, but it can also be increased in the same amount by firing t_3 and t_4. However, if the net system is discrete, it is neither live nor reversible for $\boldsymbol{m}_0$ = (1, 0, 0), for any proportional initial marking $k \cdot \boldsymbol{m}_0$. This is true because for any value of k, transitions t_1 and t_2 can be fired until $m[p_1] < 2$. Then, no transition is enabled, it is deadlocked and the system is neither live nor reversible.

Let us now consider the properties in the limit, i.e., lim-Π_L. In this case, it holds (similar to Proposition 5.10):

Theorem 5.13 *$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is $\Pi_L \Longrightarrow \exists n \in \mathbb{N}$ s.t.$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically Π_L from n.*

The proofs of Theorem 5.13 for $\mathit{\Pi}_L$ = L and for $\mathit{\Pi}_L$ = R would be analogous to that of Proposition 5.10, in which also Lemma 5.4 would be used.

Analogously to lim-DF, the reverse is not true:

$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically $\mathit{\Pi}_L \not\Longrightarrow \langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is lim-$\mathit{\Pi}_L$.

Again, the PN system in Fig. 5.4 is live and reversible for $\boldsymbol{m}_0$=(1, 0, 0, 0, 0) if it is considered as discrete. However, when considered continuous, the infinite firing sequence σ = $t_1\ t_2\ \frac{1}{2}t_3\ \frac{1}{2}t_4\ \frac{1}{2}t_2\ \frac{1}{4}t_3\ \frac{1}{4}t_4\ \frac{1}{4}t_2\ \frac{1}{8}t_3\ \frac{1}{8}t_4 \ldots$ would reach the deadlock marking $\boldsymbol{m}_d$=(0, 0, 0, 0, 2) in the limit, so the system is not lim-live and not lim-reversible from $\boldsymbol{m}_d$.

5.3 Homothetic boundedness and homothetic B-fairness in discrete Petri nets

The aim of this brief section is to propose a characterization (necessary and sufficient condition) of homothetic boundedness and homothetic B-fairness for discrete PN systems.

By definition, if a net $\mathcal{N}$ is structurally bounded (structurally B-fair), then $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically bounded (homothetically B-fair).

However, the opposite is not true: $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically bounded $\not\Rightarrow \mathcal{N}$ is structurally bounded. For instance, if there is an empty siphon in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$, some transitions can never be fired. Thus, the system can be bounded and homothetically bounded for that $\boldsymbol{m}_0$, but $\mathcal{N}$ can be unbounded for a different initial marking.

Furthermore, if $\nexists$ empty siphon in $\boldsymbol{m}_0$ (a very reasonable condition for real systems), then every transition can be fired sooner or later from $\boldsymbol{m}_0$ or from a given $k \cdot \boldsymbol{m}_0$, and then homothetic boundedness implies structural boundedness. It is analogous when B-fairness is considered.

Theorem 5.14 *Let $\mathcal{N}$ be a net system and $\boldsymbol{m}_0$ be a marking in which every siphon of the net is marked. The following statements are equivalent:*

(1) $\mathcal{N}$ is structurally bounded
(2) $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically bounded
(3) $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is bounded
(4) $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is lim-bounded

Proof. (2) $\Leftrightarrow$ (3). Because of Proposition 5.6.
(1) $\Leftrightarrow$ (3). Proved in [RTS99].
(3) $\Leftrightarrow$ (4). Stated in [RTS99]. ■

Theorem 5.15 *Let $\mathcal{N}$ be a net system and $\boldsymbol{m}_0$ be a marking in which every siphon of the net is marked. The following statements are equivalent:*

(1) $\mathcal{N}$ is structurally B-fair
(2) $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is homothetically B-fair
(3) $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is B-fair
(4) $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is lim-B-fair

Proof. (2) $\Leftrightarrow$ (3). Because of Proposition 5.7.
(1) $\Leftrightarrow$ (3). Only (3) $\Rightarrow$ (1) needs to be proven. If every siphon is initially marked, a strictly positive marking can be reached, then every T-semiflow can be fired. From the fireability of the minimal T-semiflows, it is deduced that behavioural and structural relations coincide [RTS99]. Hence B-fairness in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is equivalent to structural B-fairness in $\mathcal{N}$.
(3) $\Leftrightarrow$ (4). Only (3) $\Rightarrow$ (4) needs to be proven. This result is analogous to the equivalence between boundedness and lim-boundedness. Every pair $t, t' \in T$ is in B-fair relation, i.e., for every $\boldsymbol{m} \in RS(\mathcal{N}, \boldsymbol{m}_0)$, for every finite firing sequence $\sigma_j = \alpha_1 t_{\gamma_1} ... \alpha_j t_{\gamma_j}$ of length j which is fireable from $\boldsymbol{m}$, it holds that $\exists b \in \mathbb{R}^+$ s.t. if $\boldsymbol{\sigma}_j(t) = 0$ then $\boldsymbol{\sigma}_j(t') \leq b$ and if $\boldsymbol{\sigma}_j(t') = 0$ then $\boldsymbol{\sigma}(t)_j \leq b$. Then, considering an infinitely long sequence, $\sigma = \alpha_1 t_{\gamma_1} \dots \alpha_j t_{\gamma_j} \alpha_{j+1} t_{\gamma_{j+1}} \dots$, it holds that for every finite subsequence of σ as long as desired it is also true. Hence, if $\boldsymbol{\sigma}(t) = 0$, then $\boldsymbol{\sigma}(t')$ converges to $\lim_{j \to \infty} \boldsymbol{\sigma}_j(t') \leq b$; and if $\boldsymbol{\sigma}(t') = 0$, then $\boldsymbol{\sigma}(t)$ converges to $\lim_{j \to \infty} \boldsymbol{\sigma}_j(t) \leq b$. Hence, t, t' are in lim-B-fair relation. ■

Existence of empty siphons at a given marking, structural boundedness, and structural B-fairness of a PN system can be checked in polynomial time (see [Sil+11; STC98; SM92]). Consequently, boundedness and B-fairness of a continuous system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ can be checked in polynomial time.

5.4 On the existence of spurious deadlocks in some Petri net system subclasses

The objective of this section is to study the existence of spurious deadlocks in some net system subclasses, what is highly related to deadlock-freeness and liveness preservation.

5.4.1 EQ

First, it is stated that live and bounded equal conflict net systems [TS96] do not have spurious deadlocks, and consequently the technique in Section 2.2.2 does not report *spurious* deadlocks (i.e., it gives a necessary and sufficient condition). Since equal conflict nets are a superclass of choice-free, weighted-T-systems and marked graphs, the results obtained for equal conflict also hold for these subclasses.

The following theorem states that EQ systems preserve lim-liveness (hence lim-deadlock-freeness, liveness, and deadlock-freeness) after fluidization.

Theorem 5.16 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ be a live and bounded EQ system. Then $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is lim-live and bounded.*

Proof. Liveness and boundedness of discrete EQ systems were studied in [TS96, (Theorem 30)]. If $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is live and bounded, then $\mathcal{N}$ is consistent, conservative, and certain condition on the rank of $\boldsymbol{C}$. Moreover, every p-semiflow of $\mathcal{N}$ is marked at $\boldsymbol{m}_0$. In [RTS99], it is proven for continuous that, given an EQ net $\mathcal{N}$ which is consistent, conservative, the same condition over the rank of $\boldsymbol{C}$, and $\boldsymbol{m}_0$ a marking in which every p-semiflow is marked, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is lim-live and bounded. ■

A direct implication of Theorem 5.16 is the absence of *killing spurious* solutions in the state equation in $\mathbb{R}^+$. Hence, the technique presented in Section 2.2.2 provides not only a sufficient but also necessary condition for deadlock-freeness. Alternatively, liveness preservation (considering finite firing sequences) can also be deduced from monotonicity of liveness in bounded discrete EQ systems (Theorem 15 in [TS96]): because it is monotonically live, then it is also homothetically live. Hence, it is live when fluidified (Theorem 5.12).

5.4.2 DSSP

Let us consider the preservation of liveness in DSSP in two steps: first, preservation of liveness considering finite firing sequences (parallel to directness of the $RS_D(\mathcal{N}, \boldsymbol{m}_0)$ in live DSSP) and then preservation of lim-liveness in *consistent* and live discrete DSSP (parallel to directness of its integer linerialized reachability set).

In [RTS98] it was proven that for live and consistent DSSP, there are no spurious deadlocks for discrete PN systems. This result also holds for live and consistent continuous DSSP, although the characteristics of continuous systems need to be considered in the proofs. Let us recall first a previous result from [RTS98] which also holds for continuous DSSP with some changes and considerations.

Lemma 5.17 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a DSSP which is live as discrete. Let $\boldsymbol{m}$ be a reachable marking, and $\Delta \boldsymbol{m} \geq 0$ where $\Delta \boldsymbol{m}[p_i] = 0\ \forall p_i \notin B$ and $\tilde{\boldsymbol{m}} = \boldsymbol{m} + \Delta \boldsymbol{m}$. Let $\tilde{\boldsymbol{m}} \xrightarrow{\alpha} \boldsymbol{m}'$, then β, γ exists s.t. $\tilde{\boldsymbol{m}} \xrightarrow{\alpha} \tilde{\boldsymbol{m}}' \xrightarrow{\beta} \tilde{\boldsymbol{m}}''$, and $\boldsymbol{m} \xrightarrow{\gamma} \tilde{\boldsymbol{m}}''$, where $\tilde{\boldsymbol{m}}'' = \boldsymbol{m}'' - \Delta \boldsymbol{m}$, and $\gamma = \alpha + \beta$. Moreover, if $\boldsymbol{m}' = \tilde{\boldsymbol{m}} - \Delta \boldsymbol{m} = \boldsymbol{m} + \boldsymbol{C} \cdot \alpha \in LRS_C(\mathcal{N}, \boldsymbol{m})$, then β is a finite sequence fireable from $\boldsymbol{m}$.*

Proof. Let δ be the shortest sequence fireable from $\boldsymbol{m}$ s.t. a transition $t \in \|\boldsymbol{\alpha}\|$ is immediately fired after δ an amount a_t, i.e. $\boldsymbol{m} \xrightarrow{\delta a_t t}$.

We will proof below that, even considering continuous markings in the places, it holds that $I(\|\boldsymbol{\delta}\|) \cap I(\|\boldsymbol{\alpha}\|) = \emptyset$.

Assume contrary. Let t_δ be the first transition in δ s.t. $I(t_\delta) \in I(\|\boldsymbol{\alpha}\|)$, and t_α the first transition in α belonging to that SM. Since the difference among $\boldsymbol{m}$ and $\tilde{\boldsymbol{m}}$ is only in the buffers, then t_δ is enabled after firing $\boldsymbol{m} \xrightarrow{\delta}$ iff t_δ is enabled after firing $\tilde{\boldsymbol{m}} \xrightarrow{\delta}$. Therefore, instead of t_δ, t_α could have being fired in δ. Contradiction with the minimality of δ.

Then, let us denote $\hat{t}$ the first transition of α from the same SM than t, which is fired a certain amount $a_t \hat{t}$. I.e., $\alpha = \alpha_a \hat{t} \alpha_b$, where $I(t) = I(\hat{t}) \notin I(\|\alpha_a\|)$.

Given that t, $\hat{t}$ are the first transitions from that SM enabled at δt (from $\boldsymbol{m}$),α (from $\tilde{\boldsymbol{m}}$), and the only difference between $\boldsymbol{m}$ and $\tilde{\boldsymbol{m}}$ is in the buffers, then $enab(\hat{t}, \boldsymbol{m}) > 0 \Leftrightarrow enab(\hat{t}, \tilde{\boldsymbol{m}}) > 0$. Given that $I(\|\boldsymbol{\delta}\| \cap I(\|\boldsymbol{\alpha}\|) = \emptyset$, $\hat{t}$ is also enabled after the firing $\tilde{\boldsymbol{m}} \xrightarrow{\delta}$.

However, it can be enabled an amount smaller than a_t. In that case, we fire it the maximal allowed amount. Select:

$$a_x = min\{enab(\hat{t}, \boldsymbol{m} \xrightarrow{\delta}), enab(\hat{t}, \tilde{\boldsymbol{m}} \xrightarrow{\delta})\}$$

Selecting a_x in this way, $\delta a_x \hat{t}$ can be fired from $\boldsymbol{m}$ and also from $\tilde{\boldsymbol{m}}$. Then, apply the inductive hypothesis to $\boldsymbol{m} \xrightarrow{\delta a_x \hat{t}}$, $\tilde{\boldsymbol{m}} \xrightarrow{\delta a_x \hat{t}}$ and $\alpha_a\ (a_t - a_x)\hat{t}\ \alpha_b$, to find β',γ'. Finally, the sought sequences are: $\beta = \delta\beta'$ and $\gamma = \delta a_x \hat{t}\gamma'$.

If $\boldsymbol{m}' - \Delta\boldsymbol{m} \in \mathrm{L}RS_C(\mathcal{N}, \boldsymbol{m})$, then $\beta = \delta\beta'$ is a finite sequence fireable from $\boldsymbol{m}$.

■

The following result is obtained from [RTS98], applying Lemma 5.17, which is specific for continuous PN.

Theorem 5.18 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a live DSSP and $\boldsymbol{m}_a, \boldsymbol{m}_b \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$. Then $RS_C(\mathcal{N}, \boldsymbol{m}_a) \cap RS_C(\mathcal{N}, \boldsymbol{m}_b) = \emptyset$.*

A consequence of Theorem 5.18 is presented below.

Theorem 5.19 *Given a discrete DSSP system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ which is live, the continuous $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is also live.*

Proof.

Because of Theorem 5.18, for every pair of markings $\boldsymbol{m}_a$, $\boldsymbol{m}_b \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$, there exists a common successor. Consequently, none of them can be a deadlock marking. Consequently, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is deadlockfree as continuous, and live (because deadlock-freeness and liveness are equivalent in continuous DSSP systems[Rec98]).

■

And the result about spurious deadlocks, analogous to the one proposed for discrete consistent DSSP (using Lemma 5.17 and Theorem 5.18) is presented below.

Theorem 5.20 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a live and consistent DSSP. If $\boldsymbol{m}_a, \boldsymbol{m}_b \in$ lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$, then lim-$RS_C(\mathcal{N}, \boldsymbol{m}_a) \cap$ lim-$RS_C(\mathcal{N}, \boldsymbol{m}_b) \neq \emptyset$.*

Proof. We claim that every $\boldsymbol{m} \in$ lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$ has a successor in lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$. Then, both $\boldsymbol{m}_a$, $\boldsymbol{m}_b$, have reachable successors and applying Theorem 5.18, this successor have a common successor, which concludes the proof.

To proof the claim, we well see that $\Delta\boldsymbol{m}_0$ exists s.t. $\boldsymbol{m} + \Delta\boldsymbol{m}$ is reachable from $\boldsymbol{m}_0 + \Delta\boldsymbol{m}$. Then, applying Lemma 5.17, a successor of $\boldsymbol{m} + \Delta\boldsymbol{m}$ exists, $\tilde{\boldsymbol{m}}'$ verifying that $\tilde{\boldsymbol{m}} - \Delta\boldsymbol{m} \in RS_C(\mathcal{N}, \boldsymbol{m}) \cap RS_C(\mathcal{N}, \boldsymbol{m}_0)$, which proves the claim.

Let $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \alpha$. Let $\boldsymbol{x}$ be a T-semiflow which contains all the transitions ($\boldsymbol{x} > 0$). If we remove the buffer places from the state machine, a sequence α' with firing count vector $\boldsymbol{\alpha}' = \alpha + x$ is fireable. Since $\mathcal{N}$ is consistent, a T-semiflow $\boldsymbol{x}$ exists with $\|\boldsymbol{x}\| = T$. Then, we can add tokens to the buffers ($\Delta\boldsymbol{m}$) s.t. the marking of the buffers does not prevent the firing of sequences with firing count vectors α and $\boldsymbol{x}$. The firing of α' from $\boldsymbol{m}_0 + \Delta\boldsymbol{m}$ gives to $\boldsymbol{m} + \Delta\boldsymbol{m}$, since x is a T-semiflow.

■

Using Theorem 5.20, the following result about deadlock-freeness preservation in consistent DSSP systems is obtained.

Theorem 5.21 *Given a discrete* consistent *DSSP system* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ *which is live, the continuous* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ *is live and lim-live.*

Proof. Given that $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is live, then every transition is fireable from $\boldsymbol{m}_0$ (i.e. there are no empty siphons at $\boldsymbol{m}_0$). Due to the fact that $\mathcal{N}$ is consistent and there are not empty siphons at $\boldsymbol{m}_0$, lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$ = L$RS_C(\mathcal{N}, \boldsymbol{m}_0)$ ([RTS99], Theorem 3). Because of Theorem 5.20, there are not a marking $\boldsymbol{m}$ in L$RS_C(\mathcal{N}, \boldsymbol{m}_0)$ which is a deadlock. Consequently, $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is lim-deadlockfree as continuous, and lim-live (because deadlock-freeness and liveness are equivalent in DSSP systems, Corollary 2 in [RTS98]).

■

However, we can also prove this result in a direct way. The proof is inspired in the proof of Theorem 3.15 of [Rec98].

Theorem 5.22 *Given a discrete* consistent *DSSP system* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ *which is live, the continuous* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ *is also live.*

Proof. Suppose there exists a deadlock $\boldsymbol{m}_d$ in lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$. Then, $\forall t \in T, \exists p \in {}^\bullet t$ s.t. $m_d[p] = 0$.

Given that $\mathcal{N}_i$ are state machines, at least one place of $\mathcal{N}$ is marked. Hence, and at least a transition of $\mathcal{N}_i$ is disabled only because one buffer is empty: $\forall i \in \{1..n\}$, there exists at least a buffer b s.t. $dest(b) = i$ and it is empty ($\boldsymbol{m}_d[b] = 0$). And also all the internal transitions of the SM are not enabled.

Consider an arbitrary b which was emptied, buffer b s.t. $dest(b) = i$, and $\mathcal{N}_i$.

Since the net is consistent, we consider the there exists at least one transition $t_j \in T_j$ from a certain $\mathcal{N}_j$ such that $b \in {}^\bullet t_j$.

Consider an arbitrary j. SM $\mathcal{N}_j$ is also deadlocked, and it also have an input buffer which is empty at $\boldsymbol{m}_d$. Since the number of SM is finite, at the end of the backward recursion we will have considered every SM $\mathcal{N}_z$.

■

However, in the case of live but non consistent DSSP, there can be spurious solutions, and (lim-)liveness of the system after fluidization cannot be guaranteed. For example, the DSSP in Fig. 5.7 is not consistent, and it has a *killing* spurious marking, $\boldsymbol{m}_d = (1, 0, 0, 0, 1, 0, 0)$.

However, when more general subclasses of net systems are considered, the state equation can contain some *killing spurious* solutions. It is the case of *Multi-level Deterministically Synchronized Sequential Processes* ((DS)*SP) [RTS01], which may model complex *cooperation* relations; and *Systems of Simple Sequential Processes with Resources* (S^3PR) [ECM95], which focus on *competition* relations (see Fig. 2.4 and Fig. 3.2).

5.4.3 (DS)*SP

The subclass of (DS)*SP [RTS01], named *multi-level deterministically synchronized sequential processes*, are a generalization of DSSP which allows more complex relations (see Section 2.1).

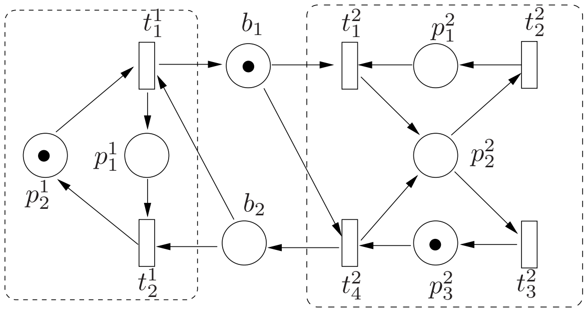


Figure 5.7: A live (but not consistent) DSSP with a spurious deadlock [RTS98].

Although they are a generalization of DSSP, some basic properties can not be generalized from DSSP to (DS)*SP in the discrete analysis, and also the results about deadlock-freeness and liveness preservation of *consistent* DSSP do not hold here.

Let us show, through the example, that discrete live (DS)*SP systems can have *killing spurious* solutions which become (lim-)reachable in the fluidified net system. The PN example in Fig. 2.4 is deadlock-free as discrete. However, it has a *killing spurious* solution which becomes (lim-)reachable in the fluidified net system. The system belongs to (DS)*SP [RTS01], a subclass of PN systems which models intricate *cooperation* relations.

Considering the initial marking depicted in the figure, the infinite firing sequence σ_d = $\frac{1}{2}t_{111}\ \frac{1}{2}t_{121}\ \frac{1}{4}t_{111}\ \frac{1}{4}t_{121}\ \frac{1}{8}t_{111}\ \frac{1}{8}t_{121}\ \ldots$ can be fired, which converges to a marking $\boldsymbol{m}_d$ in which $m_d[p_{112}]$ = $m_d[p_{122}]$ = $m_d[b_{12}]$ = 1, and the other places are empty [Rec98]. Notice that this marking is a deadlock (i.e., a *killing spurious* marking in the discrete system which becomes lim-reachable in the continuous one).

5.4.4 S^{3}PR

Other example of not preserving deadlock-freeness, even with finite sequences, is presented in Fig. 3.2 [GV99]. Without the dotted part, it models a system in which two sequential processes Pr_1 and Pr_2 share resources r_1, r_2 and r_3. It belongs to the subclass S^{3}PR [ECM95], characterized by the *competition* of processes. The PN system is live as discrete from the initial marking $\boldsymbol{m}_0$, where $m_0[r_1] = m_0[r_2] = m_0[r_3] = m_0[q_0] = m_0[p_0] = 1$ and all the other places are empty.

However, when the system is fludified, it can reach a deadlock: when the sequence σ_d = $\frac{1}{2}s_1\frac{1}{2}t_1\frac{1}{2}s_2\frac{1}{2}t_2\frac{1}{2}s_3\frac{1}{2}t_3\frac{1}{2}s_1\frac{1}{2}t_1\frac{1}{2}s_2\frac{1}{2}t_2\frac{1}{2}s_3\frac{1}{2}t_3$ is fired, a deadlock marking $\boldsymbol{m}_d$ is reached, where $m_d[r_3] = m_d[q_3] = m_d[p_3] = 1$, and the marking of all the other places is 0. Notice that $\boldsymbol{m}_d$ is a *killing spurious* marking in the discrete system, which can be reached by the continuous system with a *finite* firing sequence.

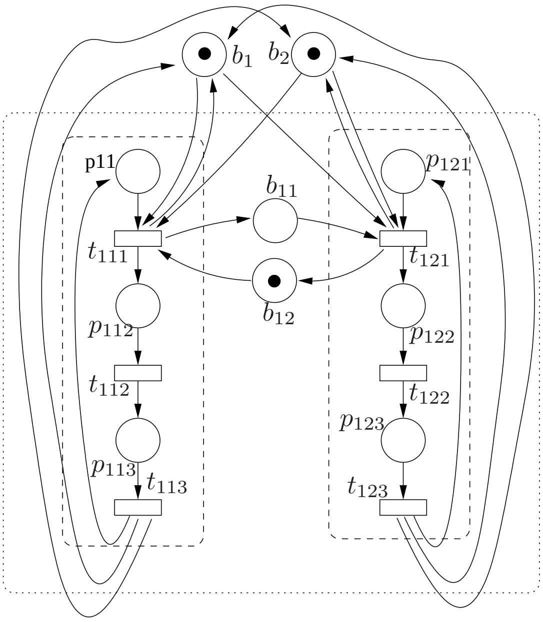


Figure 5.8: A (DS)*SP system with a spurious deadlock ([Rec98], p. 54).

5.5 Conclusions

Preservation of properties by the fluidization of untimed PN has been considered in this chapter. Basic properties such as boundedness, B-fairness, deadlock-freeness, liveness, and reversibility have been studied: If one of those properties, Π, has a *homothetic* behaviour in a discrete PN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$, Π will be preserved when the system is fluidified (i.e., in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$). Two basic kind of properties (or facts) are used to achieve these results: that every real number can be approximated by a rational one; and the properties of monotonicity and homothetic monotonicity of firing sequences in the continuous PN systems.

When boundedness or B-fairness are considered, homothetic Π_S is equivalent to Π_S of the continuous system, and also equivalent to lim-Π_S. Moreover, under some general conditions (every transition is fireable at least once) it is also equivalent to structural Π_S. However, when deadlock-freeness, liveness or reversibility, are considered, homothetic Π_L of the discrete system implies Π_L in the continuous, but it does not imply lim-Π_L. In contrast, lim-Π_L implies homothetic Π_L in the discrete system (see Fig. 5.5).

Some techniques are recalled from the discrete PN system analysis, and adapted to study homothetic deadlock-freeness and (lim-)deadlock-freeness preservation. Moreover, preservation of (lim-)liveness for some subclasses has been studied, as well as some subclasses for which, in general, deadlock-freeness or liveness is or is not preserved by fluidization.

6

Complexity analysis of continuous Petri net properties

"*Complexity is the prodigy of the world. Simplicity is the sensation of the universe.*
Behind complexity, there is always simplicity to be revealed.
Inside simplicity, there is always complexity to be discovered".
Gang Yu

An interesting issue in the study of the formalism of continuous Petri nets is its complexity analysis. This chapter continues with the research about the computational complexity, started in other works from the literature, of deciding the most common properties of untimed continuous Petri nets. The considered properties are reachability, boundedness, deadlock-freeness, liveness and reachability set inclusion.

Decidability results about properties such as boundedness and reachability set inclusion have been established. Moreover, new decision procedures for reachability and lim-reachability problems are proposed, with a better computational complexity than the previously known methods. A characterization of reachability set inclusion is proposed. It is an example of a property which is undecidable in discrete PN, and it becomes decidable in the continuous PN formalism.

The question of establishing lower bounds for CPN properties had not been considered before. In this chapter, those lower bounds are investigated for reachability, boundedness, deadlock freeness and liveness problems, with the aim of characterizing the exact complexity class.

Introduction

Given a new formalism, an interesting issue is to establish the decidability and the theoretical computational complexity of checking some of its properties. Because of that reason, the characterization of properties in the context of Continuous Petri nets and the analysis of its computational cost has been considered in some works in the literature.

One of the seminal papers in which CPN are formally defined and studied is [RTS99], in which finite reachability is considered, and the concept of lim-reachability is introduced. In that paper, it is established that assuming not empty siphons at $\boldsymbol{m}_0$ and consistency of the PN structure $\mathcal{N}$, the reachability space of a CPN is defined by its state equation, and hence it can be obtained in *polynomial* time. Boundedness property is also determined to be in *polynomial* time under these assumptions.

Concerning the decidability of *lim-reachability* of general CPN, in [RTS99] it is said: "For the moment, nothing can be said about the complexity of computing the lim-RS_C in the general case, not even if it is decidable or not."

Some years later, in [JRS03], the concepts of reachability and lim-reachability are studied again, providing a formal characterization of both concepts, as it is stated: "Reachability and lim-reachability spaces can be fully characterized using, among other elements, the state equation."

A *naïf* implementation of the characterization proposed in [JRS03] would result exponential time decision procedures for reachability and lim-reachability. These procedures are improved in this Chapter, with the aim to obtain a more accurate complexity bounds.

Regarding deadlock-freeness and liveness, it was shown in [RHS10] that (lim-)deadlock-freeness and (lim-)liveness belong in coNP. However, no lower bounds had been established before for the complexity of these properties. Table 6.1 summarises the results already known about the complexity of the associated decision problems.

From this results, it can be concluded that fluidization reduces the computational complexity of checking some properties w.r.t. discrete PN systems. Moreover, an open issue is if fluidization makes decidable some properties which are undecidable in the discrete formalism. In this work it is proved that reachability set inclusion problem, which is known to be undecidable for discrete systems [EN94], is decidable for continuous ones.

Along this chapter, $\longrightarrow$ denotes a finite firing sequence, $\longrightarrow_{\infty}$ denotes an infinite one, and $\longrightarrow_{(\infty)}$ denotes a sequence either finite or infinite.

The chapter is organized as follows. In Section 6.1, characterizations of reachability and boundedness are developed in an alternative way, with the aim to easy the decision methods. Afterwards in Section 6.2, the decision procedures for reachability, boundeness and reachability set inclusion are designed, and its computational complexity class is determined. Finally, complexity lower bounds are provided in Section 6.3: polynomial time for reachability and boundedness, and coNP for deadlock-freeness, liveness, reversibility and reachability set inclusion. Conclusions and perspectives are summarized in Section 6.4.

Table 6.1: Complexity bounds: previous results

Problems	Upper bounds
(lim-)reachability	in EXPTIME [JRS03] in PTIME for lim-reachability when all transitions are fireable at least once and the net is consistent [RTS99]
(lim-)boundedness	in PTIME when all transitions are fireable at least once [RTS99]
(lim-)deadlock-freeness	in coNP [RHS10]
(lim-)liveness	in coNP [RHS10]
(lim-)reachability set inclusion	no result

6.1 Properties characterizations

In this section, first some results about reachability of CPN are sumarized in Section 6.1.1. Based on that results, characterization of reachability and lim-reachability is presented.

6.1.1 Preliminary results about reachability and firing sequences

Most of the results of this subsection are generalisations of results given in [RTS99; JRS03].

The following lemma is an almost immediate consequence of firing definition and has for corollary the convexity of the (lim-)reachability set.

Lemma 6.1 *Given a CPN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$, (finite or infinite) sequences $\sigma, \sigma_1, \sigma_2$ markings $\boldsymbol{m}, \boldsymbol{m}', \boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}'_1, \boldsymbol{m}'_2$ and $\alpha, \alpha_1, \alpha_2 \in \mathbb{R}_{>0}$:*
(0) $\boldsymbol{m}_1 \xrightarrow{\sigma} \boldsymbol{m}'_1$ and $\boldsymbol{m}_1 \leq \boldsymbol{m}_2$ implies $\boldsymbol{m}_2 \xrightarrow{\sigma} \boldsymbol{m}'_2$ with $\boldsymbol{m}'_1 \leq \boldsymbol{m}'_2$
(1) $\boldsymbol{m} \xrightarrow{\sigma}_{(\infty)} \boldsymbol{m}$ iff $\alpha \cdot \boldsymbol{m} \xrightarrow{\alpha\sigma}_{(\infty)} \alpha \cdot \boldsymbol{m}'$
(2) $\boldsymbol{m} \xrightarrow{\sigma}_{\infty}$ iff $\alpha \cdot \boldsymbol{m} \xrightarrow{\alpha\sigma}_{\infty}$
(3) $\boldsymbol{m}_1 \xrightarrow{\sigma_1}_{(\infty)} \boldsymbol{m}'_1$ and $\boldsymbol{m}_2 \xrightarrow{\sigma_2}_{(\infty)} \boldsymbol{m}'_2$ implies $\boldsymbol{m}_1 + \boldsymbol{m}_2 \xrightarrow{\sigma_1+\sigma_2}_{(\infty)} \boldsymbol{m}'_1 + \boldsymbol{m}'_2$
(4) $\boldsymbol{m}_1 \xrightarrow{\sigma_1}_{\infty}$ and $\boldsymbol{m}_2 \xrightarrow{\sigma_2}_{\infty}$ implies $\boldsymbol{m}_1 + \boldsymbol{m}_2 \xrightarrow{\sigma_1+\sigma_2}_{\infty}$
(5) $\boldsymbol{m}_1 \xrightarrow{\alpha_1\sigma}_{(\infty)} \boldsymbol{m}'_1$ and $\boldsymbol{m}_2 \xrightarrow{\alpha_2\sigma}_{(\infty)} \boldsymbol{m}'_2$ implies $\boldsymbol{m}_1 + \boldsymbol{m}_2 \xrightarrow{(\alpha_1+\alpha_2)\sigma}_{(\infty)} \boldsymbol{m}'_1 + \boldsymbol{m}'_2$
(6) $\boldsymbol{m}_1 \xrightarrow{\alpha_1\sigma}_{\infty}$ and $\boldsymbol{m}_2 \xrightarrow{\alpha_2\sigma}_{\infty}$ implies $\boldsymbol{m}_1 + \boldsymbol{m}_2 \xrightarrow{(\alpha_1+\alpha_2)\sigma}_{\infty}$

The two next lemmas constitute a first step for the characterization of reachability since they provide sufficient conditions for reachability and lim-reachability in particular cases.

Lemma 6.2 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a continuous system, $\boldsymbol{m}$ be a marking and $\boldsymbol{\sigma} \in \mathbb{R}^T_{\geq 0}$ be a vector, that fulfill:*

- $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$*;*
- $\forall p \in {}^\bullet\|\boldsymbol{\sigma}\| \;\; m_0[p] > 0$*;*
- $\forall p \in \|\boldsymbol{\sigma}\|^\bullet \;\; m[p] > 0$.

Then there exists a finite sequence σ such that $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$ and its firing count vector is $\boldsymbol{\sigma}$.

Proof. Define $\alpha_1 \stackrel{\text{def}}{=} \min(\frac{m_0[p]}{\sum_{t \in \|v\|} \boldsymbol{Pre}[p,t]\boldsymbol{\sigma}[t]} \mid p \in {}^\bullet\|\boldsymbol{\sigma}\|)$
and $\alpha_2 \stackrel{\text{def}}{=} \min(\frac{m[p]}{\sum_{t \in \|v\|} \boldsymbol{Post}[p,t]\boldsymbol{\sigma}[t]} \mid p \in \|\boldsymbol{\sigma}\|^\bullet)$ with the convention that $\alpha_1 \stackrel{\text{def}}{=} 1$ (resp. $\alpha_2 \stackrel{\text{def}}{=} 1$) if ${}^\bullet\|\boldsymbol{\sigma}\|$ (resp. $\|\boldsymbol{\sigma}\|^\bullet$) is empty.
Due to the second and the third hypotheses α_1 and α_2 are positive.
Let $n \stackrel{\text{def}}{=} \max(\lceil \frac{1}{\min(\alpha_1, \alpha_2)} \rceil, 2)$.
Denote $\|v\| \stackrel{\text{def}}{=} \{t_1, \ldots, t_k\}$ and define $\sigma' \stackrel{\text{def}}{=} \frac{\boldsymbol{\sigma}[t_1]}{n} t_1 \ldots \frac{\boldsymbol{\sigma}[t_k]}{n} t_k$ and $\sigma \stackrel{\text{def}}{=} \sigma'^n$.
Here it is claimed that σ is the required firing sequence.
Let us denote $\boldsymbol{m}_i \stackrel{\text{def}}{=} \boldsymbol{m}_0 + \frac{i}{n}\boldsymbol{C} \cdot \boldsymbol{\sigma}$. Thus $\boldsymbol{m} = \boldsymbol{m}_n$.
By definition of α_1 and n, in $\mathcal{N}$ $\boldsymbol{m}_0 \xrightarrow{\sigma'} \boldsymbol{m}_1$ and by definition of α_2, $\boldsymbol{m}_n \xrightarrow{\sigma'^{-1}} \boldsymbol{m}_{n-1}$ in $\mathcal{N}^{-1}$.
So in $\mathcal{N}$ $\boldsymbol{m}_{n-1} \xrightarrow{\sigma'} \boldsymbol{m}_n$.
Let $1 < i < n - 1$.
Using lemma 6.1, $\frac{n-1-i}{n-1}\boldsymbol{m}_0 \xrightarrow{\frac{n-1-i}{n-1}\sigma'} \frac{n-1-i}{n-1}\boldsymbol{m}_1$ and $\frac{i}{n-1}\boldsymbol{m}_{n-1} \xrightarrow{\frac{i}{n-1}\sigma'} \frac{i}{n-1}\boldsymbol{m}_n$.
Using lemma 6.1 again and summing, one gets: $\boldsymbol{m} = \boldsymbol{m}_i \xrightarrow{\sigma'} \boldsymbol{m}_{i+1}$. ■

Lemma 6.3 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a continuous system, $\boldsymbol{m}$ be a marking and $\boldsymbol{\sigma} \in \mathbb{R}^T_{\geq 0}$ that fulfill:*

- $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$*;*
- $\forall p \in {}^\bullet\|\boldsymbol{\sigma}\|^\bullet \; m_0[p] > 0$.

Then there exists an infinite sequence σ such that $\boldsymbol{m}_0 \xrightarrow{\sigma}_\infty \boldsymbol{m}$ and its firing count vector is equal to $\boldsymbol{\sigma}$.

Proof. Let $\boldsymbol{m}_i$ be inductively defined by $\boldsymbol{m}_{i+1} = \frac{1}{2}\boldsymbol{m}_i + \frac{1}{2}\boldsymbol{m}$. and for $i \geq 1$, let $\boldsymbol{\sigma}_i = \frac{1}{2^i}\boldsymbol{\sigma}$ (thus $\|\boldsymbol{\sigma}_i\| = \|\boldsymbol{\sigma}\|$). Observe that $\boldsymbol{m}_i = \frac{1}{2^i}\boldsymbol{m}_0 + (1 - \frac{1}{2^i})\boldsymbol{m}$. So:

- $\boldsymbol{m}_{i+1} = \boldsymbol{m}_i + \boldsymbol{C}\boldsymbol{\sigma}_i$;
- $\forall p \in {}^\bullet\|\boldsymbol{\sigma}_i\|^\bullet \; \boldsymbol{m}_i[p] > 0$ and $\boldsymbol{m}_{i+1}[p] > 0$.

Applying lemma 6.2, for all $i \geq 1$ there exists σ_i such that $\boldsymbol{m}_i \xrightarrow{\sigma_i} \boldsymbol{m}_{i+1}$. Since $\lim_{i \to \infty} \boldsymbol{m}_i = \boldsymbol{m}$, the sequence $\sigma = \sigma_1\sigma_2 \ldots$ is the required sequence. ■

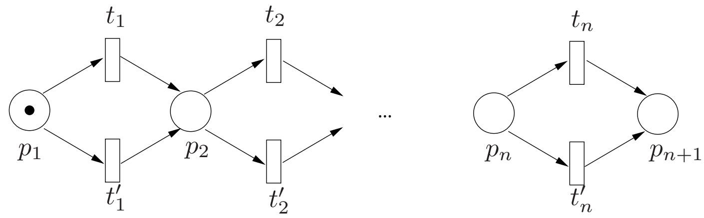


Figure 6.1: PN system with an exponentially sized firing set.

The key concept in order to get characterization of properties, is the notion of *firing set* of a CPN system [JRS03].

Definition 6.4 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a CPN system. Then its firing set $FS_C(\mathcal{N}, \boldsymbol{m}_0) \subseteq 2^T$ is defined by:*

$$FS_C(\mathcal{N}, \boldsymbol{m}_0) = \{\|\boldsymbol{\sigma}\| \mid \boldsymbol{m}_0 \stackrel{\sigma}{\longrightarrow}\}$$

Due to the empty sequence, $\emptyset \in \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$. The size of a firing set may be exponential w.r.t. the number of transitions of the net. For example, consider the CPN system of Fig. 6.1. Its firing set is:

$$\{T' \mid \forall 1 \leq j < i \leq n \ \{t_i, t'_i\} \cap T' \neq \emptyset \Rightarrow \{t_j, t'_j\} \neq \emptyset\}$$

Thus its size is at least $2^{\frac{|T|}{2}}$.

The next two lemmas establish elementary properties of the firing set and leads to new notions.

Lemma 6.5 *Let $\mathcal{N}$ be a CPN and $\boldsymbol{m}, \boldsymbol{m}'$ be two markings such that $\|\boldsymbol{m}\| = \|\boldsymbol{m}'\|$. Then $FS_C(\mathcal{N}, \boldsymbol{m}_0)(\mathcal{N}, \boldsymbol{m}) = FS_C(\mathcal{N}, \boldsymbol{m}')$.*

Proof. Since $\|\boldsymbol{m}\| = \|\boldsymbol{m}'\|$, there exists $\alpha > 0$ such that $\alpha \cdot \boldsymbol{m} \leq \boldsymbol{m}'$.
Let $\boldsymbol{m} \stackrel{\sigma}{\longrightarrow}$. Using lemma 6.1 $\alpha \cdot \boldsymbol{m} \stackrel{\alpha\sigma}{\longrightarrow}$. Since $\alpha \cdot \boldsymbol{m} \leq \boldsymbol{m}'$, $\boldsymbol{m}' \stackrel{\alpha\sigma}{\longrightarrow}$.
Thus $\mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}) \subseteq \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}')$. By symmetry, $\mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}) = \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}')$. ■

So given $P' \subseteq P$, without ambiguity it can be defined $\mathrm{FS}_C(\mathcal{N}, P')$ by:

$$\mathrm{FS}_C(\mathcal{N}, P') \stackrel{\text{def}}{=} \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}) \text{ for any } \boldsymbol{m} \text{ such that } P' = \|\boldsymbol{m}\|$$

Lemma 6.6 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a CPN system. Then $FS_C(\mathcal{N}, \boldsymbol{m}_0)$ is closed by union.*

Proof. Let $\boldsymbol{m}_0 \stackrel{\sigma}{\longrightarrow}$ and $\boldsymbol{m}_0 \stackrel{\sigma'}{\longrightarrow}$.
Then using three times lemma 6.1, $0.5\boldsymbol{m}_0 \stackrel{0.5\sigma}{\longrightarrow}$, $0.5\boldsymbol{m}_0 \stackrel{0.5\sigma'}{\longrightarrow}$ and $\boldsymbol{m}_0 \stackrel{0.5\sigma+0.5\sigma'}{\longrightarrow}$.
Since $\|0.5\sigma + 0.5\sigma'\| = \|\boldsymbol{\sigma}\| \cup \|\sigma'\|$, the conclusion follows. ■

Notation. It is denoted as $\max \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$ the maximal set of $\mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$ that is the union of all members of $\mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$.

The next proposition is a structural characterization for a subset of transitions to belong to the firing set. In addition, it shows that in the positive case, a "useful" corresponding sequence always exists and furthermore one may build this sequence in polynomial time.

Proposition 6.7 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a CPN system and T' be a subset of transitions. Then:*

$$T' \in FS_C(\mathcal{N}, \boldsymbol{m}_0) \text{ iff } \mathcal{N}_{T'} \text{ has no empty siphon in } \boldsymbol{m}_0.$$

Furthermore if $T' \in FS_C(\mathcal{N}, \boldsymbol{m}_0)$ then there exists $\sigma = \alpha_1 t_1 \dots \alpha_k t_k$ with $\alpha_i > 0$ for all i, $T' = \{t_1, \dots, t_k\}$ and a marking $\boldsymbol{m}$ such that:

- $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$;
- *for all place p, $m[p] > 0$ iff $m_0[p] > 0$ or $p \in {}^\bullet T'^\bullet$.*

Proof.
($\Longrightarrow$) Suppose $\mathcal{N}_{T'}$ contains an empty siphon Σ in $\boldsymbol{m}_0$. Then none of the transitions belonging $\Sigma^\bullet$ can be fired in the future. Since $\mathcal{N}_{T'}$ does not contain isolated places $\Sigma^\bullet (= {}^\bullet\Sigma \cup \Sigma^\bullet) \neq \emptyset$ and so $T' \notin \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$.

($\Longleftarrow$) Suppose that $\mathcal{N}_{T'}$ has no empty siphon in $\boldsymbol{m}_0$. We build by induction the sequence σ of the proposition. More precisely, we inductively prove for increasing values of i that:

- for every $j < i$ there exists a non empty set of transitions $T_j \subseteq T'$ that fulfill for all $j \neq j'$, $T_j \cap T_{j'} = \emptyset$;
- for every $j \leq i$ there exists a marking $\boldsymbol{m}_j$ with $m_j[p] > 0$ iff $m_0[p] > 0$ or $p \in {}^\bullet T_k{}^\bullet$ for some $k < j$;
- for every $j < i$ there exists a sequence $\sigma_j = \alpha_{j,1} t_{j,1} \dots \alpha_{j,k_j} t_{j,k_j}$ with $T_j = \{t_{j,1} \dots t_{j,k_j}\}$ and $\boldsymbol{m}_j \xrightarrow{\sigma} \boldsymbol{m}_{j+1}$.

There is nothing to prove for the basis case $i = 0$.
Suppose that the assertion holds until i. If $T' = T_1 \cup \dots \cup T_{i-1}$ then we are done.
Otherwise define $T'' = T' \setminus (T_1 \cup \dots \cup T_{i-1})$ and $T_i = \{t \text{ enabled in } \boldsymbol{m}_i \mid t \in T''\}$. It is claimed here that T_i is not empty. Otherwise for all $t \in T''$, there exists an empty place p_t in $\boldsymbol{m}_i$. Due to the inductive hypothesis, $\boldsymbol{m}_0(p_t) = 0$ and ${}^\bullet p_t \cap (T_1 \cup \dots \cup T_{i-1}) = \emptyset$. So the union of places p_t is an empty siphon of $\langle \mathcal{N}_{T'}, \boldsymbol{m}_0 \rangle$ which contradicts our hypothesis.

Let us denote $T_i = \{t_{i,1} \dots t_{i,k_i}\}$. Define $\alpha = \min(\frac{m_i(p)}{2k_i} \mid p \in {}^\bullet T_i)$ with the convention that $\alpha = 1$ if ${}^\bullet T_i = \emptyset$. The sequence $\sigma_i = \alpha t_{i,1} \dots \alpha t_{i,k_i}$ is fireable from $\boldsymbol{m}_i$ and leads to a marking $\boldsymbol{m}_{i+1}$ fulfilling the inductive hypothesis.

Since T'' is finite the procedure terminates. ■

The complexity result was proved in [JRS03]. Moreover, we include the complexity result below because its proof relies in a straightforward manner on the sufficiency proof of the previous proposition.

Algorithm 1: Fireable($\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C, T'$): status

Input: CPN system ($\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$), subset of transitions (T')
Output: the membership status of T' w.r.t. $\text{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$
Output: in the negative case the maximal firing set included in T'
Data: new: boolean; P': subset of places; T'': subset of transitions

$T'' = \emptyset$; $P' = \|\boldsymbol{m}_0\|$
while $T'' \neq T'$ **do**
  $new = \textbf{false}$
  for $t \in T' \setminus T''$ **do**
    if ${}^\bullet t \subseteq P'$ **then** $T'' = T'' \cup \{t\}$; $P' = P' \cup t^\bullet$; $new = \textbf{true}$;
  end
  if not new **then return** $(\textbf{false}, T'')$
end
return true

Corollary 6.8 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a CPN system and T' be a subset of transitions. Then Algorithm 1 checks in polynomial time whether $T' \in FS_C(\mathcal{N}, \boldsymbol{m}_0)$ and in the negative case returns the maximal firing set included in T' (when called with $T = T'$, it returns $\max FS_C(\mathcal{N}, \boldsymbol{m}_0)$).*

6.1.2 Characterisation of reachability and boundedness

Let us consider the basic (lim-)reachability problem, i.e., given a system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ and a real marking $\boldsymbol{m} \in \mathbb{R}^{|P|}$, does it hold $\boldsymbol{m} \in (\lim\text{-})RS_C(\mathcal{N}, \boldsymbol{m}_0)$?

A preliminary issue to consider the decidability of this problem is the representation of the input, given that $\boldsymbol{m} \in \mathbb{R}^{|P|}$. It is known that not every real number can be efficiently represented [Wei00].

However, because of the properties of PN, given a marking $\boldsymbol{m} \in \mathbb{R}^{|P|}$, there exists another marking $\boldsymbol{m}' \in \mathbb{Q}^{|P|}$ which is as near as desired to $\boldsymbol{m}$, as proved in Lemma 5.5. Hence, it can be assumed that $\boldsymbol{m} \in \mathbb{R}^{|P|}$ is a representable marking without loss of generality (as considered in this work).

In [JRS03] a characterization of reachability was presented. The theorem below is an alternative characterization that only relies on the state equation and firing sets.

Theorem 6.9 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a CPN system and $\boldsymbol{m}$ be a marking. Then $\boldsymbol{m} \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$ iff there exists $\boldsymbol{\sigma} \in \mathbb{R}^{|T|}_{\geq 0}$ such that:*

1. $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$
2. $\|\boldsymbol{\sigma}\| \in FS_C(\mathcal{N}, \boldsymbol{m}_0)$
3. $\|\boldsymbol{\sigma}\| \in FS_C(\mathcal{N}^{-1}, \boldsymbol{m})$

Proof.
($\Longrightarrow$) Let $\boldsymbol{m} \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$. So there exists a finite firing sequence σ such that $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$. By definition, $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$.

Since σ is fireable from $\boldsymbol{m}_o$ in $\mathcal{N}$, $\|\boldsymbol{\sigma}\| \in \text{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$. In $\mathcal{N}^{-1}$, $\boldsymbol{m} \xrightarrow{\sigma^{-1}} \boldsymbol{m}_0$. Since $\boldsymbol{\sigma} = \sigma^{-1}$, $\|\boldsymbol{\sigma}\| \in \text{FS}_C(\mathcal{N}^{-1}, \boldsymbol{m})$.

($\Longleftarrow$) Since $\|\boldsymbol{\sigma}\| \in \text{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$, using Proposition 6.7 and Lemma 6.1 there exists a sequence σ_1 such that $\|\boldsymbol{\sigma}\| = \|\sigma_1\|$, for all $0 < \alpha_1 \leq 1$, $\boldsymbol{m}_0 \xrightarrow{\alpha_1 \sigma_1} \boldsymbol{m}_1$ with $m_1[p] > 0$ for $p \in {}^\bullet\|\boldsymbol{\sigma}\|^\bullet$.

Since $\|\boldsymbol{\sigma}\| \in \text{FS}_C(\mathcal{N}^{-1}, \boldsymbol{m})$, using Proposition 6.7 and Lemma 6.1 there exists a sequence σ_2 such that $\|\boldsymbol{\sigma}\| = \|\sigma_2\|$, for all $0 < \alpha_2 \leq 1$, $\boldsymbol{m} \xrightarrow{\alpha_2 \sigma_2} \boldsymbol{m}_2$ in $\mathcal{N}^{-1}$ with $m_2[p] > 0$ for $p \in {}^\bullet\|\boldsymbol{\sigma}\|^\bullet$.

Choose α_1 and α_2 enough small such that the vector $\boldsymbol{\sigma'} = \boldsymbol{\sigma} - \alpha_1 \sigma_1 - \alpha_2 \sigma_2$ is non negative and $\|\boldsymbol{\sigma'}\| = \|\boldsymbol{\sigma}\|$. This is possible since $\|\boldsymbol{\sigma}\| = \|\sigma_1\| = \|\sigma_2\|$.

Since $\boldsymbol{m}_2 = \boldsymbol{m}_1 + \boldsymbol{C} \cdot \boldsymbol{\sigma'}$ and $\boldsymbol{m}_1, \boldsymbol{m}_2$ fulfill the hypotheses of Lemma 6.2, there exists a sequence σ_3 such that $\boldsymbol{\sigma'} = \sigma_3$ and $\boldsymbol{m}_1 \xrightarrow{\sigma_3} \boldsymbol{m}_2$.
Let $\sigma = (\alpha_1 \sigma_1) \sigma_3 (\alpha_2 \sigma_2)^{-1}$ then $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$. ■

The following characterization has been stated in [JRS03].

Theorem 6.10 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a CPN system and $\boldsymbol{m}$ be a marking.*
Then $\boldsymbol{m} \in$ lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$ iff there exists $\boldsymbol{\sigma} \in \mathbb{R}_{\geq 0}^{|T|}$ such that:

1. $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$
2. $\|\boldsymbol{\sigma}\| \in FS_C(\mathcal{N}, \boldsymbol{m}_0)$

Proof.
($\Longrightarrow$) Let $\boldsymbol{m} \in$ lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$. So there exists a firing sequence $\sigma = \alpha_1 t_1 \ldots \alpha_n t_n \ldots$ such that $\boldsymbol{m} = \lim_{n \to \infty} \boldsymbol{m}_n$, where $\boldsymbol{m}_n \xrightarrow{\alpha_{n+1} t_{n+1}} \boldsymbol{m}_{n+1}$.
Thus there exists $B \in \mathbb{N}$ such that for all $p \in P$ and all $n \in \mathbb{N}$, $m_n[p] \leq B$.
Let $T' \stackrel{\text{def}}{=} \{t \mid \exists i \in \mathbb{N}\ t = t_i\}$. There exists n_0 such that $T' = \{t \mid \exists i \leq n_0\ t = t_i\}$ and so $T' \in \text{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$.
Let $\alpha \in \mathbb{Q}_{>0}$ such that $\alpha \leq \min(\sum_{i \leq n_0, t_i = t} \alpha_i \mid t \in T')$.
Let us define LP_n an existential linear program where $\boldsymbol{\sigma} \in \mathbb{R}^T$ is the vector of variables by:

1. $\boldsymbol{m}_n - \boldsymbol{m}_0 = \boldsymbol{C} \cdot \boldsymbol{\sigma}$
2. $\forall t \in T'$, $\boldsymbol{\sigma}[t] \geq \alpha$
3. $\forall t \in T \setminus T'$, $\boldsymbol{\sigma}[t] = 0$

Due to the existence of the firing sequence σ, for all $n \geq n_0$ LP_n admits a solution. Using linear programming theory (see [PS98]), since $m_n[p] \leq B$ for all n and all p, there exists B' such that for all $n \geq n_0$, LP_n admits a solution $\boldsymbol{\sigma}_n$ whose items are bounded by B'.

So the sequence $\{\boldsymbol{\sigma}_n\}_{n \geq n_0}$ admits a subsequence that converges to some $\boldsymbol{\sigma}$. By continuity, $\boldsymbol{\sigma}$ fulfills $\boldsymbol{m} - \boldsymbol{m}_0 = \boldsymbol{C} \cdot \boldsymbol{\sigma}, \forall t \in T'\ \boldsymbol{\sigma}[t] \geq \alpha$ and $\forall t \in T \setminus T'\ \boldsymbol{\sigma}[t] = 0$.
So $\|\boldsymbol{\sigma}\| = T'$ and $\boldsymbol{\sigma}$ is the desired vector.

($\Longleftarrow$) Since $\|\boldsymbol{\sigma}\| \in \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$, using Proposition 6.7 and Lemma 6.1 there exists a sequence σ_1 such that $\|\boldsymbol{\sigma}\| = \|\sigma_1\|$, for all $0 < \alpha_1 \leq 1$, $\boldsymbol{m}_0 \xrightarrow{\alpha_1 \sigma_1} \boldsymbol{m}_1$ with $\boldsymbol{m}_1(p) > 0$ for $p \in {}^\bullet\|\boldsymbol{\sigma}\|^\bullet$.

Choose α_1 enough small such that the vector $\boldsymbol{\sigma'} = \boldsymbol{\sigma} - \alpha_1 \sigma_1$ is non negative and $\|\boldsymbol{\sigma'}\| = \|\boldsymbol{\sigma}\|$. This is possible since $\|\boldsymbol{\sigma}\| = \|\sigma_1\|$.

Since $\boldsymbol{m} = \boldsymbol{m}_1 + \boldsymbol{C} \cdot \boldsymbol{\sigma'}$ and $\boldsymbol{m}_1$ fulfills the hypotheses of lemma 6.3, there exists an infinite sequence σ_2 such that $\boldsymbol{\sigma'} = \sigma_2$ and $\boldsymbol{m}_1 \xrightarrow{\sigma_2}_\infty \boldsymbol{m}$.
Let $\sigma = (\alpha_1 \sigma_1)\sigma_2$ then $\boldsymbol{m}_0 \xrightarrow{\sigma}_\infty \boldsymbol{m}$. ∎

Using the previous theorem, a short proof is developed to show that iterating the lim-reachability is useless.

Theorem 6.11 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ be a CPN system. Then for all $\boldsymbol{m} \in$ lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$, lim-$RS_C(\mathcal{N}, \boldsymbol{m}) \subseteq$ lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$.*

Proof. Let $\boldsymbol{m'} \in$ lim-$RS_C(\mathcal{N}, \boldsymbol{m})$. Due to theorem 6.10, there exists $\boldsymbol{\sigma}, \boldsymbol{\sigma'} \in \mathbb{R}^{|T|}_{\geq 0}$ such that:

1. $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$ and $\boldsymbol{m'} = \boldsymbol{m} + \boldsymbol{C} \cdot \boldsymbol{\sigma'}$
2. $\|\boldsymbol{\sigma}\| \in \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$ and $\|\boldsymbol{\sigma'}\| \in \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m})$

Thus $\boldsymbol{m'} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot (\boldsymbol{\sigma} + \boldsymbol{\sigma'})$.

Due to proposition 6.7, since $\|\boldsymbol{\sigma}\| \in \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$ there exists a sequence σ and a marking $\boldsymbol{m}^*$ such that $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}^*$ and $\|\boldsymbol{\sigma}\| = \|\boldsymbol{\sigma}\|$ and $\|\boldsymbol{m}^*\| = \|\boldsymbol{m}_0\| \cup {}^\bullet\|\boldsymbol{\sigma}\|^\bullet$.

Since $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, $\|\boldsymbol{m}\| \subseteq \|\boldsymbol{m}^*\|$ and so $\|\boldsymbol{\sigma'}\| \in \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}^*)$. Hence $\|\boldsymbol{\sigma} + \boldsymbol{\sigma'}\| = \|\boldsymbol{\sigma}\| \cup \|\boldsymbol{\sigma'}\| \in \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$. Using in the other direction the characterization of theorem 6.10 with $\boldsymbol{\sigma} + \boldsymbol{\sigma'}$, one gets $\boldsymbol{m'} \in$ lim-$RS_C(\mathcal{N}, \boldsymbol{m}_0)$. ∎

A characterization of boundedness for CPN systems is presented below.

Theorem 6.12 *Given a CPN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$. Then $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is unbounded iff:*
There exists $\boldsymbol{\sigma} \in \mathbb{R}^T_{\geq 0}$ such that $\boldsymbol{C} \cdot \boldsymbol{\sigma} \gneq \boldsymbol{0}$ and $\|\boldsymbol{\sigma}\| \subseteq \max FS_C(\mathcal{N}, \boldsymbol{m}_0)$.

Proof.
($\Longleftarrow$) Assume there exists $\boldsymbol{\sigma} \in \mathbb{R}^T_{\geq 0}$ such that $\boldsymbol{C} \cdot \boldsymbol{\sigma} \gneq \boldsymbol{0}$ and $\|\boldsymbol{\sigma}\| \subseteq \max \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$. Denote $T' \stackrel{\text{def}}{=} \max \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$. Using proposition 6.7, there exists $\boldsymbol{m}_1 \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$ such that for all $p \in {}^\bullet T'^\bullet$, $\boldsymbol{m}_1(p) > 0$. Define $\boldsymbol{m}_2 \stackrel{\text{def}}{=} \boldsymbol{m}_1 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, thus $\boldsymbol{m}_2 \gneq \boldsymbol{m}_1$. Since

$\|\boldsymbol{\sigma}\| \subseteq T'$, $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ fulfill the hypotheses of lemma 6.2. Applying it, yields a firing sequence $\boldsymbol{m}_1 \xrightarrow{\sigma} \boldsymbol{m}_2$. Iterating this sequence establishes the unboundedness of $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$.

($\Longrightarrow$) Assume $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ is unbounded. Then there exists $p \in P$ and a family of firing sequences $\{\sigma_n\}_{n \in \mathbb{N}}$ such that $\boldsymbol{m}_0 \xrightarrow{\sigma_n} \boldsymbol{m}_n$ and $m_n[p] \geq n$. Since $\{\|\boldsymbol{\sigma}_n\|\}_{n \in \mathcal{N}}$ is finite by extracting a subsequence w.l.o.g. it can assumed that all these sequences have the same support, say $T' \subseteq \max \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$.

Let $\boldsymbol{\sigma}_n \stackrel{\text{def}}{=} \boldsymbol{C} \cdot \boldsymbol{\sigma}_n$. Define $\boldsymbol{w}_n = \frac{\boldsymbol{\sigma}_n}{\|\boldsymbol{\sigma}_n\|_1}$. Since $\{\boldsymbol{w}_n\}_{n \in \mathbb{N}}$ belongs to a compact set, there exists a convergent subsequence $\{\boldsymbol{w}_{\alpha(n)}\}_{n \in \mathbb{N}}$. Denote $\boldsymbol{w}$ its limit. Since $\|\boldsymbol{w}\|_1 = 1$, $\boldsymbol{w}$ is non null. It can be claimed that $\boldsymbol{w}$ is a non negative vector. Since $m_n[p] \geq n$, $\|\boldsymbol{\sigma}_n\|_1 \geq \boldsymbol{\sigma}_n[p] \geq n - m_0[p]$. On the other hand, for all $p' \in P$, $\boldsymbol{w}_n[p'] \geq \frac{-m_0[p']}{\|\boldsymbol{\sigma}_n\|_1}$. Combining the two inequalities, for $n > m_0[p]$, $\boldsymbol{w}_n[p'] \geq \frac{-m_0[p']}{n - m_0[p]}$. Applying this inequality to $\alpha(n)$ and letting n go to infinity yields $\boldsymbol{w}[p'] \geq 0$.

Due to standard results of polyhedra theory (see [AFP02] for instance), the set $\{\boldsymbol{C}[P, T'] \cdot \boldsymbol{u} \mid \boldsymbol{u} \in \mathbb{R}^{T'}_{\geq 0}\}$ is closed. So there exists $\boldsymbol{u} \in \mathbb{R}^{T'}_{\geq 0}$ such that $\boldsymbol{w} = \boldsymbol{C} \cdot \boldsymbol{u}$. Considering $\boldsymbol{u}$ as a vector of $\mathbb{R}^{T}_{\geq 0}$ by adding null components for $T \setminus T'$ yields the required vector. ■

6.2 Decision procedures

In this section, decision procedures for the characterization of (lim-)reachability are given, and it is proven to be polynomial time. Moreover, boundedness is proved to be polynomial time in the general case.

Naively implementing the characterization of reachability would lead to an exponential procedure since it would require to enumerate the items of $\mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$ (whose size is possibly exponential). For each item, say T', the algorithm would check in polynomial time (1) whether T' belongs to $\mathrm{FS}_C(\mathcal{N}^{-1}, \boldsymbol{m})$ and (2) whether the associated linear program $\boldsymbol{\sigma} > \boldsymbol{0} \wedge \boldsymbol{C}[P, T'] \cdot \boldsymbol{\sigma} = \boldsymbol{m} - \boldsymbol{m}_0$ admits a solution. Guessing T' shows that the reachability problem belongs to NP.

In fact, this upper bound is improved here with the help of Algorithm 15. When $\boldsymbol{m} \neq \boldsymbol{m}_0$, this algorithm maintains a subset of transitions T' which fulfills $\|\boldsymbol{\sigma}\| \subseteq T'$ for any $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$ (as will be proven in proposition 6.13). Initially T' is set to T. Then lines 4-10 build a solution to the state equation restricted to transitions of T' with a maximal support (if there is at least one). If there is no solution then the algorithm returns false. Otherwise T' is successively restricted to (1) the support of this maximal solution (line 11), (2) the maximal firing set in $\max \mathrm{FS}_C(\mathcal{N}_{T'}, m_0[{}^\bullet T' \cup T'^\bullet])$ (line 12) and, (3) the maximal firing set in $\max \mathrm{FS}_C(\mathcal{N}^{-1}_{T'}, m[{}^\bullet T' \cup T'^\bullet])$ (line 13). If the two last restrictions do not modify T' then the algorithm returns true. If T' becomes empty then the algorithm returns false.

Omitting line 13, Algorithm 15 decides the lim-reachability problem.

Proposition 6.13 *Algorithm 15 returns true iff $\boldsymbol{m}$ is reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$.*
Algorithm 15 without line 13 returns true iff $\boldsymbol{m}$ is lim-reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$.

Algorithm 2: Reachable($\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C, \boldsymbol{m}$): status

Input: CPN system ($\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$), marking ($\boldsymbol{m}$)
Output: boolean representing the reachability status of $\boldsymbol{m}$)
Output: firing count vector of one firing sequence in the positive case
Data: $nbsol$: integer; $\boldsymbol{\sigma}, \boldsymbol{sol}$: vectors; T': subset of transitions

if $\boldsymbol{m} = \boldsymbol{m}_0$ **then return** (**true**,**0**);
$T' = T$;
while $T' \neq \emptyset$ **do**
    $nbsol = 0$; $\boldsymbol{sol} = \mathbf{0}$;
    for $t \in T'$ **do**
        solve $\exists ? \boldsymbol{\sigma}\ \boldsymbol{\sigma} \geq \mathbf{0} \wedge \boldsymbol{\sigma}[t] > 0 \wedge \boldsymbol{C}[P, T'] \cdot \boldsymbol{\sigma} = \boldsymbol{m} - \boldsymbol{m}_0$;
        if $\exists \boldsymbol{\sigma}$ **then** $nbsol = nbsol + 1$; $\boldsymbol{sol} = \boldsymbol{sol} + \boldsymbol{\sigma}$;
    end
    if $nbsol = 0$ **then return false**;
    else $\boldsymbol{sol} = \frac{1}{nbsol} \boldsymbol{sol}$;
    $T' = \|\boldsymbol{sol}\|$;
    $T' = T' \cap \max \mathrm{FS}_C(\mathcal{N}_{T'}, m_0[{}^\bullet T' \cup T'^\bullet])$;
    $T' = T' \cap \max \mathrm{FS}_C(\mathcal{N}_{T'}^{-1}, m[{}^\bullet T' \cup T'^\bullet])$; *// deleted for lim-reachability* **if** $T' = \|\boldsymbol{sol}\|$ **then return** (**true**,$\boldsymbol{sol}$);
end
return false;

Proof. Only the non trivial case $\boldsymbol{m} \neq \boldsymbol{m}_0$ is considered.

Assume that the algorithm returns true at line 13.
By definition, vector $\boldsymbol{sol}$ which is a barycentre of solutions is also a solution with maximal support and so fulfills the first statement of Theorem 6.9. Since $T' = \|\boldsymbol{sol}\|$ at line 13, $\|\boldsymbol{sol}\| \in \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$ due to line 12 and $\|\boldsymbol{sol}\| \in \mathrm{FS}_C(\mathcal{N}^{-1}, \boldsymbol{m})$ due to line 13. Thus $\boldsymbol{m}$ is reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ since it fulfills the assertions of Theorem 6.9. In case of lim-reachability, line 13 is omitted. So the assertions of Theorem 6.10 are fulfilled and $\boldsymbol{m}$ is lim-reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$.

Assume the algorithm returns false.
We claim that at any time the algorithm fulfills the following invariant: for any $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$, $\|\boldsymbol{\sigma}\| \subseteq T'$.
This invariant initially holds since $T' = T$. At line 11 due to the first assertion of Theorem 6.9, for any such σ, $\|\boldsymbol{\sigma}\| \subseteq \|\boldsymbol{sol}\|$ since $\boldsymbol{sol}$ is a solution with maximal support. So the assignment of line 11 lets true the invariant. Due to the second assertion of Theorem 6.9 and the invariant, any σ fulfills $\|\boldsymbol{\sigma}\| \subseteq \max \mathrm{FS}_C(\mathcal{N}_{T'}, \boldsymbol{m}_0[{}^\bullet T' \cup T'^\bullet])$. So the assignment of line 12 lets true the invariant. Due to the third assertion of Theorem 6.9 and the invariant, any σ fulfills $\|\boldsymbol{\sigma}\| \subseteq \max \mathrm{FS}_C(\mathcal{N}_{T'}^{-1}, \boldsymbol{m}[{}^\bullet T' \cup T'^\bullet])$. So the assignment of line 13 lets true the invariant.

If the algorithm returns false at line 10 due to the invariant the first assertion of Theorem 6.9 cannot be satisfied. If the algorithm returns false at line 15 then $T' = \emptyset$. So due to the invariant and since $\boldsymbol{m} \neq \boldsymbol{m}_0$, $\boldsymbol{m}$ is not reachable from $\boldsymbol{m}_0$.

The case of lim-reachability is similarly handled with the following invariant: for any $\boldsymbol{m}_0 \xrightarrow{\sigma}_\infty \boldsymbol{m}$, $\|\boldsymbol{\sigma}\| \subseteq T'$. ■

Proposition 6.14 *The reachability and the lim-reachability problems for CPN systems are decidable in polynomial time.*

Proof. Let us analyse the time complexity of Algorithm 15. Since T' must be modified in lines 12 or 13 in order to start a new iteration of the main loop, there are at most $|T|$ iterations of this loop. The number of iterations of the inner loop is also bounded by $|T|$. Finally, solving a linear program can be performed in polynomial time [PS98] as well as computing the maximal item of a firing set (see corollary 6.8). ■

In [JRS03], it is proven that the lim-reachability problem for consistent CPN systems with no empty siphons in the initial marking is decidable in polynomial time. We improve this result by showing that this problem and a similar one belong to NC $\subseteq$ PTIME (a complexity class of problems that can take advantage of parallel computations, see [Pap94]).

Proposition 6.15 *The reachability problem for consistent CPN systems with no empty siphons in the initial marking and no empty siphons in the final marking for the reverse net belongs to NC.*
The lim-reachability problem for consistent CPN systems with no empty siphons in the initial marking belongs to NC.

Proof. Due to the assumptions on siphons and proposition 6.7 only the first assertion of Theorems 6.9 and 6.10 needs to be checked. Due to consistency, there exists $\boldsymbol{w} > \mathbf{0}$ such that $\boldsymbol{C} \cdot \boldsymbol{w} = \mathbf{0}$. Assume there is some $\boldsymbol{\sigma} \in \mathbb{R}^T$ such that $\boldsymbol{m} - \boldsymbol{m}_0 = \boldsymbol{C} \cdot \boldsymbol{\sigma}$. For some $n \in \mathbb{N}$ large enough, $\boldsymbol{\sigma'} \stackrel{\text{def}}{=} \boldsymbol{\sigma} + n \cdot \boldsymbol{w} \in \mathbb{R}^T_{\geq 0}$ and still fulfills $\boldsymbol{m} - \boldsymbol{m}_0 = \boldsymbol{C} \cdot \boldsymbol{\sigma'}$.

Now the decision problem $\exists ? \boldsymbol{\sigma} \in \mathbb{R}^T$ s.t. $\boldsymbol{m} - \boldsymbol{m}_0 = \boldsymbol{C} \cdot \boldsymbol{\sigma}$ belongs to NC [CLP01]. ■

Proposition 6.16 *The boundedness problem for CPN systems is decidable in polynomial time.*

Proof. Using the characterization of Theorem 6.12, it is first computed in polynomial time $T' = \max \text{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$ (see Corollary 6.8). Then for all $p \in P$, one solves the existential linear program $\exists ? \boldsymbol{\sigma} \geq \mathbf{0}\ \ \boldsymbol{C}[P, T'] \cdot \boldsymbol{\sigma} \geq \mathbf{0} \wedge (\boldsymbol{C}[P, T'] \cdot \boldsymbol{\sigma})[p] > 0$. The CPN system is unbounded if some of these linear programs admits a solution. ■

In discrete Petri nets, the reachability set inclusion problem is undecidable, while the restricted problem of home state is decidable (see [EN94] for a detailed survey about decidability results in PN). In CPN systems, this problem is decidable thanks to the special structure of the (lim-)reachability sets.

Proposition 6.17 *The reachability set inclusion and the lim-reachability set inclusion problems for CPN systems are decidable in exponential time.*

Proof. Let us define $TP \stackrel{\text{def}}{=} \{(T', P') \mid T' \in \text{FS}_C(\mathcal{N}, \boldsymbol{m}_0) \wedge P' \subseteq P \wedge T' \in \text{FS}_C(\mathcal{N}^{-1}, P')\}$. For every pair $(T', P') \in TP$, define the polyhedron $E[T', P']$ over $\mathbb{R}^P \times \mathbb{R}^{T'}$ by:

$$E[T', P'] \stackrel{\text{def}}{=} \{(\boldsymbol{m}, \boldsymbol{\sigma}) \mid \boldsymbol{m}[P'] > \mathbf{0} \wedge \boldsymbol{m}[P \setminus P'] = \mathbf{0} \wedge \boldsymbol{\sigma} > \mathbf{0} \wedge \boldsymbol{m} = \boldsymbol{C}[P, T'] \cdot \boldsymbol{\sigma}\}$$

and $R[T', P']$ by: $R[T', P'] \stackrel{\text{def}}{=} \{\boldsymbol{m} \mid \exists \boldsymbol{\sigma}\ (\boldsymbol{m}, \boldsymbol{\sigma}) \in E[T', P']\}$
Using the characterization of Theorem 6.9 and Lemma 6.5,
$RS_C(\mathcal{N}, \boldsymbol{m}_0) = \bigcup_{(T', P') \in TP} R_{T', P'}$.
Due to Lemma 6.1, the reachability set of a CPN system is convex. So $RS_C(\mathcal{N}, \boldsymbol{m}_0)$ can be rewritten as:

$$RS_C(\mathcal{N}, \boldsymbol{m}_0) = \{ \sum_{(T', P') \in TP} \gamma[T', P'] \boldsymbol{m}[T', P'] \mid$$

$$\sum_{(T', P') \in TP} \gamma[T', P'] = 1 \wedge \forall (T', P') \in TP,\ \gamma[T', P'] \geq 0 \wedge \boldsymbol{m}[T', P'] \in R[T', P']\}$$

Observe that this representation is exponential w.r.t. the size of the CPN system.

Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ and $\langle \mathcal{N}', \boldsymbol{m}'_0 \rangle$ be two CPN systems for which one wants to check whether $RS_C(\mathcal{N}, \boldsymbol{m}_0) \subseteq RS_C(\mathcal{N}', \boldsymbol{m}'_0)$. One builds the representation above for $RS_C(\mathcal{N}, \boldsymbol{m}_0)$ and $RS_C(\mathcal{N}', \boldsymbol{m}'_0)$. Then one transforms the representation of the set $RS_C(\mathcal{N}', \boldsymbol{m}'_0)$ as a system of linear constraints. This can be done in polynomial time w.r.t. the original representation [BHZ05]. So the number of constraints is still exponential w.r.t. the size of $\langle \mathcal{N}', \boldsymbol{m}'_0 \rangle$.

Afterwards for every constraint of this new representation, one adds its negation to the representation of $RS_C(\mathcal{N}, \boldsymbol{m}_0)$ and check for a solution of such a system. $RS_C(\mathcal{N}, \boldsymbol{m}_0) \nsubseteq RS_C(\mathcal{N}', \boldsymbol{m}'_0)$ iff at least one of these linear programs admits a solution. The overall complexity of this procedure is still exponential w.r.t. the size of the problem. The procedure for lim-reachability set inclusion can be straightforwardly obtained from this one. ■

6.3 Hardness results

In this section, lower bounds for most of the properties considered in previous sections are provided.

Two complexity problems which will be used in the reductions are recalled first, and then some reductions from the considered properties to the classical problems are proposed.

6.3.1 Classical complexity problems

A NP-complete and a PTIME-complete problems that will be used in some proofs in the next section are presented here.

The satisfiability problem for conjunctive normal form propositional formulas, a classical NP-complete problem is explained below.

Let $\boldsymbol{x} = \{x_1, x_2, ...x_n\}$ be a set of n atomic boolean propositions. Let $cl_j \stackrel{\text{def}}{=} l_{j1} \vee l_{j2} \vee l_{j3}$ be m clauses of 3 literals, where $l_{jk} \in \{x_1, ..., x_n, \neg x_1, ..., \neg x_n, \}$.

The **satisfiability problem** is stated as: Does there exist an interpretation ν, with $\nu : \boldsymbol{x} \longrightarrow \{False, True\}$, such that the m clauses cl_j are true?

A classical PTIME-complete problem is to **determine the value of a circuit**.

A circuit $\mathcal{C}$ is composed of four types of gates: $False$, $True$, AND, OR. Each gate has an output. $False$ and $True$ gates (which are unique) have no entries. Gates AND and OR have at least two inputs. Any input of a gate is connected to an output of another gate. If we define a relations $next$ between the gates by a $next$ b iff one input of a is connected to an output of b, then the transitive closure of $next$ is irreflexive. In other words, $next$ defines a partial order between gates. One of the gates of the circuit is distinguished and denoted out.

The value of the inputs and outputs of a circuit is defined inductively according to the relation $next$. The output of gate $False$ (resp. $True$) is $False$ (resp. $True$). The input of a gate is set to the value of the output to which it is connected. The output of a gate AND, OR is obtained by its inputs, applying its truth table. The problem of determining the value of a circuit is to determine the value of the output of the gate out.

6.3.2 Hardness reductions

In this section, matching lower bounds are provided for almost all problems analysed in the previous sections.

Proposition 6.18 *The reachability, lim-reachability and boundedness problems for CPN systems are PTIME-complete.*

Proof. Due to propositions 6.14 and 6.16, it is only needed to prove that these problems are PTIME-hard. So we design a LOGSPACE reduction from the circuit value problem (a PTIME-complete problem [Pap94]) to these problems.

A circuit $\mathcal{C}$ is composed of four kinds of gates: `False`, `True`, `AND`, `OR`. Each gate has an output. There is a single `False` gate and a single `True` gate and they have no inputs. Gates whose type is `AND` or `OR` have two inputs. Any input of a gate is connected to an output of another gate. Let the binary relation $\prec$ between the gates be defined by: $a \prec b$ if the output of a is connected to an input of b. Then one requires that the transitive closure of $\prec$ is irreflexive. One of the gates of the circuit, out, is distinguished and its output is not the input of any gate. The value of the inputs and outputs of a circuit is defined inductively according to the relation $\prec$. The output of gate `False` (resp. `True`) is **false** (resp. **true**). The input of a gate is equal to the value of the output to which it is connected. The output of a gate `AND` or `OR` is obtained by applying its truth table to its inputs. The circuit value problem consists in determining the value of the output of gate out.

The reduction is done as follows: The gate `True` is modelled by a place $p_{\texttt{True}}$ initially containing a token. This is the only place initially marked. The gate `False` is modelled by a place $p_{\texttt{False}}$. Any gate c of kind `AND` yields a place p_c and a transition t_c whose inputs and outputs is represented in Fig. 6.2(a) and any gate c of kind `OR` yields a place p_c and two transitions t_{c1} and t_{c2} whose inputs and outputs are represented in Fig. 6.2(b). Finally, the subnet represented in Fig. 6.3 is added, with one transition $clean_p$ per place p different from p_{out}. This reduction can be performed in LOGSPACE.

We prove by induction on $\prec$ that a transition t_c (resp. t_{c1} or t_{c2}) is enabled iff the gate c of kind `AND` (resp. `OR`) has value **true**.

Assume that gate c of kind `AND` has value **false**. Then one of its input say a has value **false**. If a is the gate `False` then p_a is initially unmarked and cannot be marked since it has no input. If a is a gate of kind `AND` then by induction on $\prec$, t_a is never enabled and so p_a will always be empty. If a is a gate of kind `OR` then by induction on $\prec$, t_{a1} and t_{a2} are never enabled and so p_a will always be empty. Thus whatever the case t_c can never be enabled. The case of a gate c of kind `OR` is similar.

Assume that gate c of kind `AND` has value **true**. Then both its inputs say a and b have value **true**. If a (resp. b) is the gate `True` then p_a (resp. p_b) is initially marked. If a (resp. b) is a gate of kind `AND` then by induction on $\prec$, t_a (resp. t_b) can be enabled. If a is a gate of kind `OR` then by induction on $\prec$, some t_{ai} (resp. t_{bi}) can be enabled. Now consider the sequence $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$ of proposition 6.7 w.r.t. $\max \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$. In $\boldsymbol{m}$, every place initially marked or output of a transition that belongs to $\max \mathrm{FS}_C(\mathcal{N}, \boldsymbol{m}_0)$ is marked. So t_c is enabled in $\boldsymbol{m}$. The case of a gate c of kind `OR` is similar.

Now observe that the total amount of tokens in the net can only be increased by transition $grow$ and in this case place p_{out} is unbounded. Since p_{out} can contain tokens iff the value of gate out is **true**, it has been proved that the CPN system is unbounded iff the gate out is **true**.

Finally, let $\boldsymbol{m}$ be defined by $m[p_{out}] = 1$ and $m[p] = 0$ for all $p \neq p_{out}$. If the value of gate out is **false** then p_{out} will never be marked and so $\boldsymbol{m}$ is nor (lim-)reachable. If the value of gate out is **true** then transition t_{out} can be fired by some small amount say $0 < \varepsilon \leq 1$. Then all the other places can be unmarked by transitions $clean_p$ followed by a finite number of firings of $grow$ in order to reach $\boldsymbol{m}$. So $\boldsymbol{m}$ is (lim-)reachable iff the value of gate out is **true**. ■

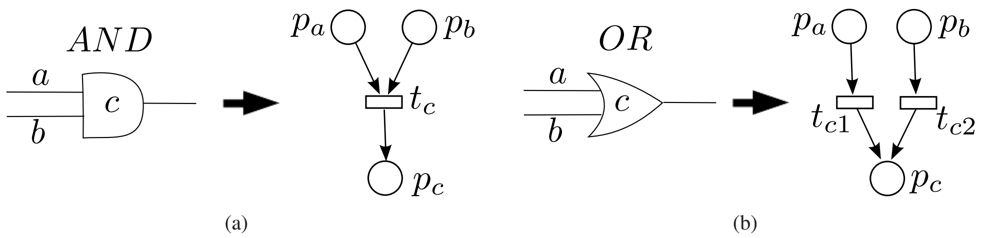


Figure 6.2: Reductions of the gates (a) AND and (b) OR to CPN.

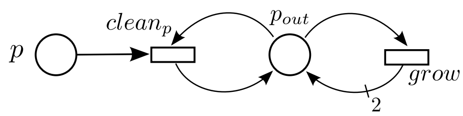


Figure 6.3: An additional subnet.

In order to prove that the lower bounds are robust, the subclass of free-choice PN (see definition in Chapter 2) has been considered.

Proposition 6.19 *The (lim-)deadlock-freeness and (lim-)liveness problems in free-choice CPN systems are coNP-hard.*

Proof. We use almost the same reduction from the 3SAT problem as the one proposed for free-choice Petri nets in [DE95]. However the proof of correctness is specific to continuous nets.

Let $\{x_1, x_2, \ldots, x_n\}$ denote the set of propositions and $\{c_1, c_2, \ldots, c_m\}$ denote the set of clauses. Every clause c_j is defined by $c_j \stackrel{\text{def}}{=} lit_{j1} \vee lit_{j2} \vee lit_{j3}$ where for all j, k, $lit_{jk} \in \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$. The satisfiability problem consists in the existence of an interpretation $\nu : \{x_1, x_2, \ldots, x_n\} \longrightarrow \{\mathbf{false}, \mathbf{true}\}$, such that for all clauses c_j, $\nu(c_j) = \mathbf{true}$.

Every proposition x_i yields a place b_i initially marked with a token (all other places are unmarked) and input of two transitions t_i, f_i corresponding to the assignment associated with

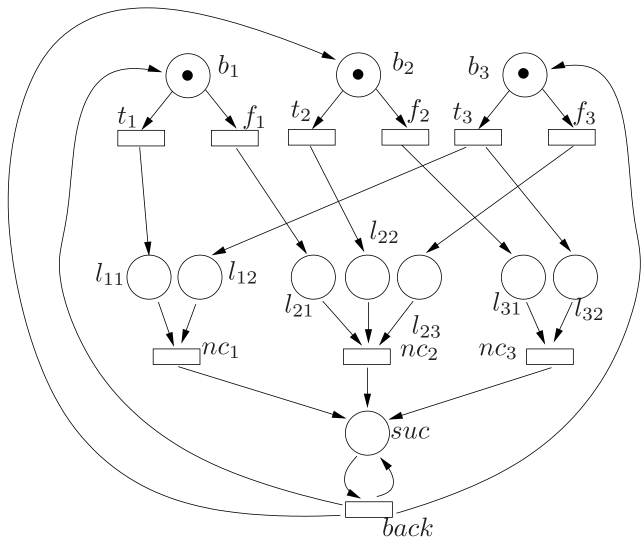


Figure 6.4: CPN corresponding to formula $(\neg x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3)$.

an interpretation. Every of literal lit_{jk} yields a place l_{jk} which is the output of transition t_i if $lit_{jk} = x_i$ or transition f_i if $lit_{jk} = \neg x_i$ Every clause c_j yields a transition nc_j with three input "literal" places corresponding to literals $\neg lit_{j1}, \neg lit_{j2}, \neg lit_{j3}$. An additional place suc is the output of every transition nc_j. Finally, transition $back$ has suc as a loop place and b_i for all i as output places. The reduction is illustrated in Fig. 6.4.

Assume that there exists ν such that for all clause c_j, $\nu(c_j) = \mathbf{true}$. Then fire the following sequence $\sigma = 1t_1^* \dots 1t_n^*$ where $t_i^* = t_i$ when $\nu(x_i) = \mathbf{true}$ and $t_i^* = f_i$ when $\nu(x_i) = \mathbf{false}$. Consider $\boldsymbol{m}$ the reached marking. Because $\nu(c_j) = \mathbf{true}$, at least one input place of nc_j is empty in $\boldsymbol{m}$. Moreover $m[suc] = m[b_i] = 0$ for all i. Hence $\boldsymbol{m}$ is a deadlock.

Assume that there does not exist ν such that for all clause c_j, $\nu(c_j) = \mathbf{true}$. Observe that given a marking $\boldsymbol{m}$ such that $m[suc] > 0$ all transitions will be fireable in the future and suc will never decrease (thus $m[suc] > 0$ for a lim-reachable marking $\boldsymbol{m}$ as well).

So only reachable markings $\boldsymbol{m}$ such that $m[suc] = 0$ are considered, i.e. when no transitions nc_j have been fired. The goal here is to prove that from such marking there is a sequence that produces tokens in suc. Examining the remaining transitions, the following invariants hold. For all atomic proposition x_i, and reachable marking $\boldsymbol{m}$, one has

$$\forall i \; m[b_i] + \sum_{l_{jk} \in \{x_i, \neg x_i\}} m[l_{jk}] \geq 1$$

$$\forall j, k, j', k' \; lit_{jk} = lit_{j'k'} \Rightarrow m[l_{jk}] = m[l_{j'k'}]$$

If for some i, $m[b_i] > 0$, t_i is fired in order to empty b_i. Thus the invariants become:

$$\forall i \sum_{l_{jk} \in \{x_i, \neg x_i\}} m[l_{jk}] \geq 1$$

$$\forall j, k, j', k' \; lit_{jk} = lit_{j'k'} \Rightarrow m[l_{jk}] = m[l_{j'k'}]$$

Now define ν by $\nu(x_i) = \mathbf{true}$ if for some $lit_{jk} = x_i$, $m[l_{jk}] > 0$. Due to the hypothesis, there is a clause c_j such that $\nu(c_j) = \mathbf{false}$. Due to our choice of ν and the invariants, all inputs of nc_j are marked. So firing nc_j marks suc. ■

Moreover, it is shown below that even the hypotheses that allow the lim-reachability to belong in NC (i.e.,consistency and absence of initially empty siphons) do not reduce the complexity of other problems.

Proposition 6.20 *The (lim-)deadlock-freeness, (lim-)liveness and reversibility problems in consistent CPN systems with no initially empty siphons are coNP-hard.*

Proof.

Another reduction from the 3SAT problem is used here, as already described in the proof of proposition 6.19.

Every proposition x_i yields a place b_i initially marked with a token (all other places are unmarked) and input of two transitions: (1) t_i with output place p_i and, (2) f_i with output place n_i corresponding to the assignment associated with an interpretation. Every clause c_j yields a transition nc_j. Transition nc_j has three loop places corresponding to literals lit_{jk}: if $lit_{jk} = x_i$ then the input is n_i, if $lit_{jk} = \neg x_i$ then the input is p_i. An additional place suc is the output of transition nc_j. A transition nd has suc for input place and no output place. Finally,for every x_i, there are transitions tb_i and fb_i which are respectively reverse transitions of t_i and f_i with an additional loop over place suc. The reduction is illustrated in Fig. 6.5. The net is consistent with consistency vector: $\sum_i (t_i + tb_i + f_i + fb_i) + \sum_j (nc_j + nd)$. It does not contain an initially empty siphon since every siphon includes some place b_i. This proves that every transition can be fired at least once from $\boldsymbol{m}_0$.

Assume that there exists ν such that for all clause c_j, $\nu(c_j) = \mathbf{true}$. Then fire the following sequence $\sigma = 1t_1^* \dots 1t_n^*$ where $t_i^* = t_i$ when $\nu(x_i) = \mathbf{true}$ and $t_i^* = f_i$ when $\nu(x_i) = \mathbf{false}$. Consider $\boldsymbol{m}$ the reached marking. Since $\nu(c_j) = \mathbf{true}$, at least one input place of nc_j is empty in $\boldsymbol{m}$. Moreover $m[suc] = m[b_i] = 0$ for all i. So $\boldsymbol{m}$ is dead and the net is not reversible.

Assume that there does not exist ν such that for all clause c_j, $\nu(c_j) = \mathbf{true}$. Our goal is to prove that from any (lim-)reachable marking there is a sequence that comes back to $\boldsymbol{m}_0$. Since from $\boldsymbol{m}_0$ all transitions are fireable at least once this proves that the net is (lim-)live and (lim-)deadlock free.

For all atomic proposition x_i, and reachable marking $\boldsymbol{m}$, one has

$$\forall i \; m[b_i] + m[p_i] + m[n_i] = 1$$

Since a lim-reachable marking is a limit of reachable markings, this invariant also holds for lim-reachable markings.

If for some i, $m[b_i] > 0$, t_i is fired in order to empty b_i. Thus the invariant becomes:

$$\forall i \; m[p_i] + m[n_i] = 1$$

Now define ν by $\nu(x_i) = \mathbf{true}$ if $m[p_i] > 0$. Due to the hypothesis, there is a clause c_j such that $\nu(c_j) = \mathbf{false}$. Due to our choice of ν and the invariant, all inputs of nc_j are marked. So firing nc_j marks suc. Now fire transitions tb_i and fb_i in order to empty places p_i and n_i. So $m[b_i] = 1$. Finally, nd is fired in order to empty place suc and it is done. ■

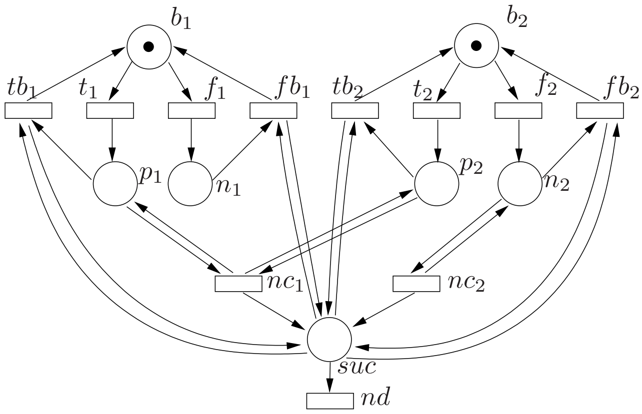


Figure 6.5: CPN system corresponding to formula $(\neg x_1 \vee \neg x_2) \wedge x_2$.

6.4 Conclusions

In this chapter, decidability and complexity analysis of the properties considered in this thesis has been analysed for continuous Petri nets. The complexity class of most of these properties has been characterized by designing new decision procedures and/or providing reductions to complete problems. It has also been shown that the reachability set inclusion, undecidable for Petri nets, becomes decidable in the continuous framework. This result brings up the fact that the formalism of continuous PN makes decidable some properties which are undecidable in discrete PN. These results are summarised in Table 6.2.

Table 6.2: Complexity bounds

Problems	**Upper and lower bounds**
(lim-)reachability	PTIME-complete in NC for lim-reachability (resp. reachability) when all transitions are fireable at least once (resp. and also in the reverse CPN) and the net is consistent
(lim-)boundedness	PTIME-complete
(lim-)deadlock-freeness and (lim-)liveness	coNP-complete coNP-hard even for free-choice CPN or for CPN when all transitions are fireable at least once and the net is consistent
(lim-)reachability set inclusion	in EXPTIME coNP-hard even for reversibility in CPN when all transitions are fireable at least once and the net is consistent

Part III

On the fluidization of timed Petri nets

7 Previous concepts on the fluidization of timed Petri nets

"*You may delay, but time will not*".
Benjamin Franklin

In this chapter, fluidization of PN with a time interpretation associated to the firing of transitions is considered. Among the different time interpretations which have been proposed in the literature, timed continuous PN under *infinite server semantics*, which is a frequently used semantics, is defined and studied. Then, the approximation of the behaviour of discrete PN with time such as stochastic PN is considered. Finally, we show that the previous transformation (addition of implicit places) of the original discrete system proposed in Chapter 3 provides a better approximation of the original discrete system when fluidified.

Introduction

By introducing time to the untimed PN, timed PN are obtained. This chapter introduces the basic concepts about continuous PN with a time interpretation. Among the different firing semantics, the focus is on *infinite server semantics* (in Section 7.1, which has been proved to provide a better approximation to discrete systems under some usual conditions.

Moreover, the fluidization of the *Stochastic Petri Nets* (SPN) defined in Chapter 2 is considered. Specifically, the approximation of the steady state throughput of discrete stochastic PN by timed continuous PN is studied. Section 7.2 illustrates that the preliminary transformations of the original discrete system proposed in Chapter 3 provide a better approximation to the original discrete system when fluidified.

7.1 Timed continuous Petri nets under infinite servers semantics

When a timed interpretation is introduced to the autonomous model, the firing of transitions depends explicitly on time. And, consequently, the state equation of the system does also depend on time:

$$\boldsymbol{m}(\tau) = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau) \tag{7.1}$$

This equation can be derived with respect to time:

$$\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\boldsymbol{\sigma}}(\tau) \tag{7.2}$$

The derivative of the firing sequence is denoted as the *(firing) flow*, i.e.

$$\boldsymbol{f}(\tau) = \dot{\boldsymbol{\sigma}}(\tau) \tag{7.3}$$

And, hence, equation 7.2 is rewritten as follows, which is the equation describing the dynamics of the timed system:

$$\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \boldsymbol{f}(\tau). \tag{7.4}$$

In TCPN, a firing rate λ_j is associated to each transition t_j. And depending of the definition of the flow $\boldsymbol{f}$ we deal with *Timed Continuous Petri Nets* (TCPN) under different firing semantics.

In this work, TCPN under *Infinite Server Semantics* (ISS) is considered. Other firings semantics from the literature are *finite servers semantics* and *product semantics*. *Finite servers semantics*, considered in [AD98], distinguishes among *strongly enabledness* and *weak enabledness*. Moreover, *product semantics* [SR02] is obtained through decolouration of coloured nets, in which the *minimum* operator of ISS is replaced by a *product*.

In *infinite server semantics*, the flow a transition is proportional to its firing rate and to its enabling degree. The flow of a transition t_i at a marking $\boldsymbol{m}$ is defined as follows:

$$\boldsymbol{f}[t_i] = \boldsymbol{\lambda}[t_i] \cdot enab(t_i, \boldsymbol{m}) = \lambda[t_i] \cdot \min_{p_j \in {}^\bullet t_i} \frac{m[p_j]}{Pre[p_j, t_i]}, \tag{7.5}$$

The resulting dynamical system is a *piecewise linear* system in which switches occur due to the minimum operators.

The TCPN system is defined as a net structure $\mathcal{N}$, an initial marking $\boldsymbol{m}_0$ and a vector of firing rates $\boldsymbol{\lambda}$:

Definition 7.1 *A TCPN system is a tuple* $\langle \mathcal{N}, \boldsymbol{m}_0, \boldsymbol{\lambda} \rangle_C$ *where* $\mathcal{N}$ *is the structure,* $\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ *is the initial marking and* $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^{|} T|$ *is the vector of firing rates.*

It has been proved in the literature that TCPN under ISS have the capability to simulate Turing machines [RHS10]. As a consequence, they have an big expressive power, and certain important properties become *undecidable* (for example, marking coverability, submarking reachability or the existence of a steady-state).

If the TCPN converges to a steady state (it is not the case in all TCPN systems, see for example [Jim+04]), then the throughput of a transition t_i is equal to its flow f_i, and it is denoted as $\chi_{TCPN}(t_i)$. From equations (7.4) and (7.5), $\dot{\boldsymbol{m}} = \boldsymbol{C} \cdot \boldsymbol{f}_{ss} = 0$ is obtained, where $\boldsymbol{f}_{ss}$ is the flow vector of the timed system at the steady state, i.e., $\boldsymbol{f}_{ss} = \lim_{\tau \to \infty} \boldsymbol{f}(\tau)$.

The definitions of deadlock-freeness and liveness of untimed CPN are directly extended to TCPN, defined over the flow of the system at its steady state:

Definition 7.2 *Let* $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle_C$ *be a timed continuous* PN *system and* $\boldsymbol{f}_{ss}$ *be the flow at its steady state.*

- $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle_C$ *is* timed-deadlock-free *if* $\boldsymbol{f}_{ss} \neq \mathbf{0}$*;*
- $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle_C$ *is* timed-live *if* $\boldsymbol{f}_{ss} > \mathbf{0}$*;*
- $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ *is* structurally timed-live *if* $\exists\, \boldsymbol{m}_0$ *such that* $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ *is* timed-live.

The firing rate vector imposes some constrains can be imposed over the evolution of the system. For example, a property which is not satisfied in the untimed system (for example, deadlock-freeness) can be satisfied by the time system for a specific combination of $\boldsymbol{\lambda}$. For example, the PN system in Fig. 2.1(b) is deadlock-free if a firing vector $\boldsymbol{\lambda}$ is chosen in which $\lambda_1 = \lambda_2$, although it deadlocks as an untimed system.

7.1.1 The concept of configuration

The dynamics of the TCPN under ISS is described as a piecewise linear system, due to the min operator which appears in (7.5). Three related concepts are derived from it, which are relevant in the understanding of the behaviour of TCPN:

Definition 7.3

- *A configuration* $\mathcal{C}_k$*. It is the set of places defining the enabling degree of the transitions. It is defined as a set of pairs* (p, t) *covering every transition.*

- *A region $\mathcal{R}_k$. It is the convex set of markings in which a certain configuration $\mathcal{C}_k$ holds. It is a subset of its reachability space: $\mathcal{R} \subseteq RS_C(\mathcal{N}, \boldsymbol{m}_0)$.*
- *Operation mode. It corresponds to the* linear system *which drives the flow in a given region.*

Moreover, associated to each configuration $\mathcal{C}_k$, a configuration matrix $\boldsymbol{\Pi}_k \in \mathbb{Q}|T| \times |P|$ is defined as follows:

$$\boldsymbol{\Pi}_k[t,p] = \begin{cases} \frac{1}{Pre[p,t]}, & \text{if } (p,t) \in \mathcal{C}_k \\ 0, & \text{otherwise} \end{cases} \tag{7.6}$$

The *operation mode*, which describes the evolution inside a region $\mathcal{R}_k$ is defined as:

$$\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \boldsymbol{f}(\tau) = \boldsymbol{C} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m}(\tau), \tag{7.7}$$

where $\boldsymbol{\Lambda} = diag(\boldsymbol{\lambda})$ is a diagonal $|T| \times |T|$ matrix containings the firing rates of transitions and the configuration matrix is $\boldsymbol{\Pi}(\boldsymbol{m}) = \boldsymbol{\Pi}_k$ where $\boldsymbol{\Pi}_k$ is the configuration matrix associated to $\mathcal{R}_k$.

The number of configurations is upper bounded by $\prod_{t \in |T|} |^\bullet t|$ [MRS10].

7.1.2 Monotonicity and paradoxes

Some interesting properties can be directly extracted from the definition of the infinite server semantics.

The flow of a system is homothetic monotonous with respect to the marking of the system and its firing rates:

- Given a system $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle_C$, if $\boldsymbol{\lambda}$ is multiplied by k then the same markings are reached, but the flow of the system is k times bigger;
- Given a system $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle_C$, if $\boldsymbol{m}_0$ is multiplied by k, the reachable markings are the same ones but multiplied by k, and the flow is also k times bigger.

Two monotonicity results of the steady-state throughput are satisfied under some general conditions [MRS09]:

Proposition 7.4 *Assume $\langle \mathcal{N}, \boldsymbol{\lambda}_i, \boldsymbol{m}_i \rangle_C$, $i = 1, 2$ are mono-T-semiflow TCPN under infinite server semantics that reach a steady state. Assume that the set of places belonging to the arcs of the steady state configuration contains the support of a P-semiflow.*

1. *$\langle \mathcal{N}, \boldsymbol{\lambda}_1, \boldsymbol{m}_1 \rangle_C$ and $\langle \mathcal{N}, \boldsymbol{\lambda}_1, \boldsymbol{m}_2 \rangle_C$ verify $\boldsymbol{m}_1 \leq \boldsymbol{m}_2$ or*
2. *$\langle \mathcal{N}, \boldsymbol{\lambda}_1, \boldsymbol{m}_1 \rangle_C$ and $\langle \mathcal{N}, \boldsymbol{\lambda}_2, \boldsymbol{m}_1 \rangle_C$ verify $\boldsymbol{\lambda}_1 \leq \boldsymbol{\lambda}_2$,*

then the steady state flows satisfy $\boldsymbol{f}_1 \leq \boldsymbol{f}_2$.

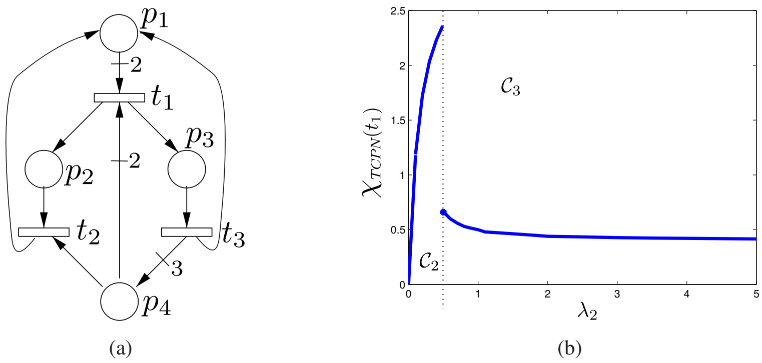


Figure 7.1: (a) Mono-T-semiflow PN system. Steady state throughput for different values of λ_2, with $\lambda_1 = \lambda_3 = 1$, and $\boldsymbol{m}_0 = (15, 1, 1, 0)$. Its steady state throughput is counter-intuitive (faster transition t_2, slower behaviour).

By contrast, some counter-intuitive behaviours appear in the analysis of TCPN under ISS. For example, the throughput of a TCPN is not in general an upper bound of the throughput of the discrete PN. Moreover, the steady state throughput is not monotone in general, i.e., if only some components of $\boldsymbol{\lambda}$ or only some components of $\boldsymbol{m}_0$ are increased the steady state throughput can decrease [SR04].

Consider the mono-T-semiflow TCPN under ISS in Fig. 7.1(a), with $\lambda = \mathbf{1}$ and $\boldsymbol{m}_0 = (15, 1, 1, 0)$. Different mode operation can drive the evolution of the system at its steady-state. For example, if $0 < \lambda_2 \leq 0.5$, the flow in steady-state is $f_1(\tau) = m_1(\tau)$, $f_2(\tau) = m_4(\tau)$ and $f_3(\tau) = m_3(\tau)$, respectively. Therefore, $\mathcal{C}_2 = \{(p_1, t_1), (p_4, t_2), (p_3, t_3)\}$ is the steady-state configuration and places $\{p_1, p_4, p_3\}$ give the flow. If rate λ_2 is increased, $\mathcal{C}_3 = \{(p_4, t_1), (p_2, t_2), (p_3, t_3)\}$ becomes the steady-state configuration and monotonicity does not hold (Fig. 7.1(b)).

7.2 Approximation of the steady state throughput of stochastic Petri nets

In this section, the approximation of the steady state throughput of the stochastic Petri nets (see Section 2.3) obtained by the TCPN under ISS is considered First, the effect of the preliminary transformations that can be done over the original discrete PN system (as proposed in Chapter 3) is studied. Then, it is stated that the deterministic limit of stochastic Petri nets coincides with TCPN, and hence they coincide for large populations.

7.2.1 Improvements obtained by the removal of spurious solutions

Some techniques to transform the original discrete PN system before its fluidization where proposed in Chapter 3. They consist in adding some places which are *implicit* in the discrete system but not in the continuous one. The goal of this section is to illustrate the effect that those *implicit* places have over the throughput of the system.

For that purpose, every PN example from Chapter 3 is recalled here. For each example, the steady state throughput of the system considered as SPN is compared with the steady state throughput of the TCPN without the implicit places, and with the implicit places (denoted as TCPN+impl). Different certain combinations of $\boldsymbol{\lambda}$ are chosen for each example, to illustrate that in some cases there is not effect or it is not important; and for certain firing rate combinations, the improvements are significant.

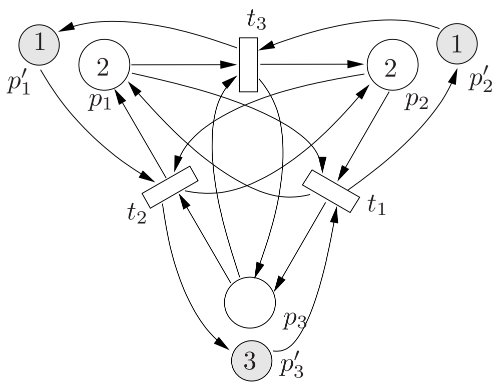


Figure 7.2: Without the grey places, PN system with three spurious deadlocks. They are removed with the implicit places drawn in grey colour.

Table 7.1: Steady state throughput of t_1 of the PN in Fig. 7.2. Comparative of different methods.

Method	$\chi(t_1)$ $\boldsymbol{\lambda} = (1,1,1)$	$\chi(t_1)$ $\boldsymbol{\lambda} = (2,1,1)$	$\chi(t_1)$ $\boldsymbol{\lambda} = (3,2,1)$
SPN	0.5714	0.6377	0.8403
TCPN	1.0110	0.0393	0.0495
TCPN+impl	1.0132	0.6701	0.8716

Example 7.5 ***Removing empty traps***
Consider Fig.7.2 and Table 7.1. First column shows a firing rate vector for which the implicit places do not have effect over the steady state throughput. Second and third columns, in

which the firing rates of the different transitions are different, the value obtained in by the TCPN+impl (TCPN with implicit places) approximates better the SPN than the TCPN.

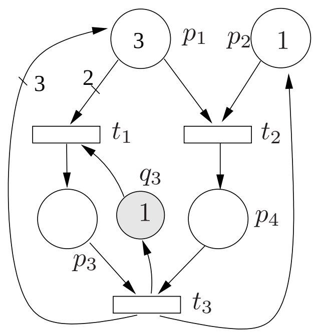


Figure 7.3: Petri net system. The *marking truncation* place q_3 drawn in grey colour is implicit in the discrete system, but removes the undesired markings from its LRS_C.

Table 7.2: $\chi(t_1)$ of the PN in Fig. 7.3, with different $\boldsymbol{\lambda}$.

Method	$\chi(t_1)$ $\boldsymbol{\lambda} = (5, 5, 1)$	$\chi(t_1)$ $\boldsymbol{\lambda} = (5, 1, 1)$
SPN	0.769	0.492
TCPN	0.833	0.025
TCPN+impl	0.833	0.499

Example 7.6 ***Marking truncation places***

Consider the PN in Fig. 7.3. The marking truncation *place q_3 is added. Places q_1, q_2 and q_4 would not modify the continuous system, so they have not been added. The improvement given by q_3 is not significant when $\lambda_1 = \lambda_2$ (see first column of Table 7.2), but it is specially big when $\lambda_1 > \lambda_2$ (see second column of the table). In this case, $\chi_{TCPN}(t_1) = 0.025$ because the steady state marking is near from the spurious deadlock $\boldsymbol{m}_3$.*

Example 7.7 ***Vertex cutting places***

Consider the PN in Fig. 7.4. Its throughput for different combinations of $\boldsymbol{\lambda}$ is shown in Table 7.3. For $\boldsymbol{\lambda} = \mathbf{1}$, the improvement is not relevant. However, for $\boldsymbol{\lambda} = (1, 1, 10, 1, 1)$ the TCPN with the implicit places gives a better approximation to the original SPN.

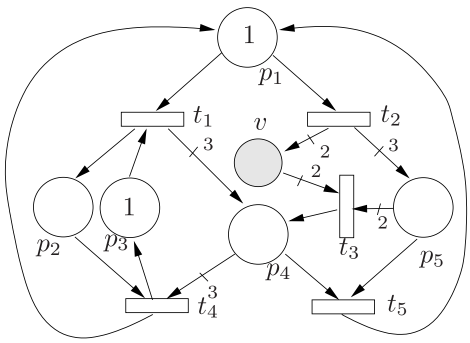


Figure 7.4: PN system. The *vertex cutting* place cuts the undesired deadlock.

Table 7.3: $\chi(t_1)$ of the PN in Fig. 7.4 with different $\boldsymbol{\lambda}$.

Method	$\chi(t_1)$ $\boldsymbol{\lambda} = \mathbf{1}$	$\chi(t_1)$ $\boldsymbol{\lambda} = (1, 1, 10, 1, 1)$
SPN	0.25	0.322
TCPN	0.2527	0.003
TCPN+impl	0.2521	0.3573

Example 7.8 ***Enabling truncation places***
Consider the PN in Fig. 7.5 with $k = 3$*,* $q = 5$ *and* $\boldsymbol{\lambda} = (1, 5)$*. The SEB of* t_1 *is* $SEB(t_1) = \lfloor 1.67 \rfloor = 1$*. Hence, an* enabling truncation *place* tp_1 *with initial marking equal to 1 is added (see place* tp_1*, drawn in grey colour Fig. 3.5). Place* tp_1 *is implicit in the discrete PN because it does not modify its behaviour (*$\chi_{SPN}(t_1) = 0.83$*, see first row at Table 7.4). However,* tp_1 *is not implicit in the continuous PN system, and it makes the continuous system to be more faithful w.r.t the original discrete system (*$\chi_{TCPN+impl}(t_1) = 1.00$*, which is a better approximation than the* $\chi_{TCPN}(t_1) = 1.45$*). Only* tp_1 *has been added because places* tp_2 *and* tp_3 *do not affect to the continuous PN.*

7.2.2 Deterministic limit of stochastic Petri nets

In population dynamics, the *deterministic limit* [JS02] describes the trajectory towards which the population densities of a discrete stochastic system converge as its size tends to infinity. Let us consider a SPN with initial marking $\boldsymbol{m}_0 = k \cdot \mathbf{x}_0 \in \mathbb{N}^{|P|}_{\geq 0}$ where $\mathbf{x}_0 \in \mathbb{R}^{|P|}_{\geq 0}$ represents the initial marking density of the system, and $k \in \mathbb{R}$ represents the system size (or volume).

Let us define the vector field for place p_j as $F_j(\mathbf{x}) = \sum_{t_i \in (^\bullet p_j \cup p_j{}^\bullet)} C[p_j, t_i] \cdot f_i$, where $f_i = \lambda_i \cdot enab(t_i, \mathbf{x})$ (notice that F_j is a nonnegative function of real arguments on the system densities). Let $F(\mathbf{x})$ be a vector composed of the vector field functions $F_j(\mathbf{x})$ of every place p_j. The two following conditions can be easily checked: a) $F(\mathbf{x})$ is Lipschitz continuous, i.e.,

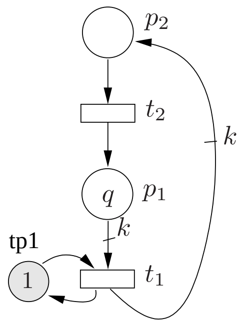


Figure 7.5: PN example to illustrate the *enabling truncation* places.

Table 7.4: $\chi(t_1)$ of the PN in Fig. 7.5, with $\boldsymbol{\lambda} = (1, 5)$.

Method	$\chi(t_1)$
SPN	0.90
TCPN	1.33
TCPN+impl	1.00

$\exists H \geq 0$ such that $|F(\mathbf{x}) - F(\mathbf{y})| \leq H \cdot |\mathbf{x} - \mathbf{y}|$; b) $\sum_{t_i \in (\bullet p_j \cup p_j \bullet)} |C[p_j, t_i]| \cdot f_i(\mathbf{x}) < \infty$. Then, the deterministic limit behaviour of the marking densities $\mathbf{x}$ of the SPN when k tends to infinity is given by the following set of differential equations [EK86; JS02]: $\dot{\mathbf{x}} = F(\mathbf{x}) = \boldsymbol{C} \cdot \boldsymbol{f}$.

Thus, the deterministic limit of a SPN matches with the time evolution defined for TCPN, and therefore a TCPN captures faithfully the behaviour of a SPN with high markings. However, in order to obtain a suitable continuous approximation for SPN with low markings, further manipulations are required on the TCPN.

7.3 Conclusions

In this chapter, fluidization of timed PN has been introduced. The most common *semantics* associated to the flow of transitions have been enumerated. Timed Continuous Petri nets under *infinite server semantics* is the semantics considered in this thesis, and its basic concepts have been explained in this section.

Moreover, the basic transformations performed over a discrete PN system before its fluidization proposed in Chapter 3 have been considered. This transformations add some places which are implicit in the discrete system.

In this chapter, it has been shown that the added places can modify the throughput of the system when it is fluidified, providing a better approximation to the steady state throughput of discrete stochastic PN.

8

The "bound reaching problem"

"*The only reason for time is so that everything doesn't happen at once*".
Albert Einstein

This chapter deals with the basic operation of fluidization of *timed* Petri nets. More precisely, the "bound reaching problem" is identified, which is an interesting problem which may appear when "relatively small" populations are considered. This problem points out the differences between discrete and continuous behaviour that appear when the probability of a transition to be enabled is low in the discrete case. The "bound reaching problem" refers to the situation in which the enabling bound of a transition is equal to 1. Among the different concerns related to this problem, the approximation of the throughput of discrete Markovian PN by Timed Continuous PN under *infinite server semantics*is considered. An approach denoted ρ-$semantics$ based in *infinite servers semantics* is proposed to tackle this problem and it is compared with other methods.

Introduction

This chapter identifies a certain situation in which the fluidization of timed systems might be not accurate. As it occurs for untimed PN, the behaviour of the timed continuous model can be completely different just because the integrality constraint has been dropped. Some works have been proposed to improve the accuracy of the original fluid approximation, such as [VS12], [LL12].

When the population of the system is large, the continuous PN system often approximates adequately the behaviour of the discrete one (see Section 7.2.2). Unfortunately, this is not the case in systems in which the population is “relatively small” in a part of the system. The technique presented here is specially interesting when the population is “relatively” small in some parts of the system (if a place has a large population, but also a big amount of tokens is required for its output transition, it can be said that it is “relatively small”). We are considering neither very small populations (in which fluidization is not needed) nor very large ones (in which fluidization usually provides a good approximation of the original PN). This chapter focuses on the approximation of timed discrete PN by means of continuous PN in the cases in which the maximum marking of a place is equal to the weight of one of its output arcs. It is what is called here as the *Bound Reaching Problem* (BRP).

The BRP is a challenging problem that appears in many practical cases. It can arise in systems in which very low and very large populations are combined. In particular, it also appears when inhibitor arcs of a bounded system are removed and simulated with regular arcs and places, because the complementary place that is added presents exactly this problem: its marking bound is equal to the weight of at least one of its output arcs.

Among the different concerns related to the BRP, we will concentrate on the approximation of the mean throughput of a stochastic PN system by its continuous counterpart.

The rest of the chapter is organized as follows. Section 8.1 introduces the *bound reaching problem*. In Section 8.2, the BRP is considered for transitions which a unique input place (i.e., transitions which are not joins). Two preliminary approaches to address it are considered. Then, a continuous approach to tackle the *bound reaching problem* is proposed, which is a macro structure of immediate and ISS transitions, whose behaviour is compacted and denoted as ρ-$semantics$. It is applied to two examples. The ρ-$semantics$ is generalized to join transitions in Section 8.3, and it is applied to other two examples. Conclusions are shown in Section 8.4.

8.1 The Bound Reaching Problem

The *bound reaching problem* studies a particular situation in which the continuous approximation does not approximate correctly the behaviour of the discrete PN.

As previously pointed (Section 7.2.2), TCPN approximate reasonably well the behaviour of SPN when the populations are relatively large. However, when “relatively small” populations are also considered it is not the case.

Table 8.1: $\chi(t_1)$ in the PN in Fig. 8.1, with $\boldsymbol{\lambda} = (10, 1, 1)$; considered as SPN and as TCPN.

Meth	k = 1	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8	k = 9	k = 10
SPN	0.833	0.417	0.242	0.144	0.085	0.049	0.028	0.016	0.008	0.005
TCPN	0.833	0.833	0.833	0.833	0.833	0.833	0.833	0.833	0.833	0.833

This lack of accuracy is related to the fact that synchronizations are strongly relaxed when the net is fluidified. Consider a transition t and a place p such that ${}^{\bullet}t = \{p\}, Pre[p,t] = k$ (for example, t_1 and p_1 in Fig. 8.2(a)). Considered as a discrete system, t is only enabled when $m[p] \geq k$, and the probability of that transition to be enabled can be low. However, as continuous, t is enabled for any positive amount of tokens $m[p] > 0$, regardless of the arc weight k. An extreme case occurs when the maximum possible amount of tokens in p is equal to k. Then, this place p needs to "reach its bound" in order to enable transition t. This lack of accuracy and the search of alternative fluid schemes to improve it is what we call the BRP.

We identify that a transition t in a system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ suffers from the BRP when its structural enabling bound is equal to 1. We do not consider the case in which it is smaller than 1, because in that case t can never be fired in the discrete model, so it can be removed)

A transition t_i in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is said to suffer from the BRP if its structural enabling bound is equal to 1 (if it is smaller than 1, t_i never can be fired in the discrete model, so it can be removed).

Let us define the set of transitions which suffer from the *bound reaching problem* as *Bound Reaching Transition Set* ($BRTS$), based in the *structural enabling bound* defined in Chapter 2.

$$BRTS = \{t \mid SEB(t) = 1\} \tag{8.1}$$

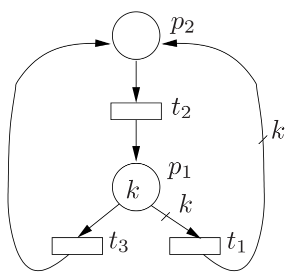


Figure 8.1: PN system in which transition t_1 suffers from the BRP.

Consider the PN example in Fig. 8.1 as a discrete system. Apparently, it has four parameters (λ_1, λ_2, λ_3 and k). However, one of the firing rates can be fixed (here, $\lambda_2 = 1$), which seen just as a time scale: only three parameters can be considered, without lost of generality. In this example, the BRP appears in t_1. Considered as discrete, transitions t_1 and

t_3 are enabled at $\boldsymbol{m}_0 = (k, 0)$. Suppose t_1 is fired first. Then, all the tokens are moved to p_2, and only t_2 is enabled. After one or more firings of t_2, transitions t_2, t_3 are enabled an can fired. Transition t_3 is enabled when $m[p_1] \geq 1$. However, t_1 can only fire when $m[p_1] = k$. Consequently, transition t_1 is not fired very often and its throughput is low.

Moreover, if k grows, then the probability of having k tokens in p_1 decreases, and hence the steady state throughput $\chi_{SPN}(t_1)$ of the SPN decreases. It can be seen in Table 8.1, in which the steady state throughput of t_1 for different values of k is shown in the first row for SPN. The steady state throughput also depends on the ratio among the firing rates λ_i. Assuming λ_2 a scale constant, $\chi_{SPN}(t_1)$ increases monotonically with λ_1, and $\chi_{SPN}(t_1)$ decreases when λ_3 increases (see the obtained $\chi_{SPN}(t_1)$ for different combinations of them in next paragraph).

Consider the PN example in Fig. 8.1. Transition $t_1 \in BRTS$, in fact $\forall \boldsymbol{m} \in RS_D(\mathcal{N}, \boldsymbol{m}_0)$, $enab(t_1, \boldsymbol{m}) \leq 1$. Moreover, the BRP described here can be more relevant depending on the firing rates of the transitions.

Let us see with some examples the evolution of $\chi_{SPN}(t_1)$. Let us fix $\lambda_2 = 1$. Tables 8.2, 8.3 and 8.4 show the calculated $\chi_{SPN}(t_1)$ for different values of λ_1, λ_3 and k, which complement the results in Table 8.1.

Table 8.2: $\chi_{SPN}(t_1)$ of the SPN in Fig. 8.1, with $k = 4$, $\lambda_2 = 1$, for different values of λ_1 and λ_3.

λ_1	0.1	1	10
λ_3 = 0.1	0.0591	0.2666	0.4111
λ_3 = 1	0.0060	0.0468	0.1442
λ_3 = 10	0.0000	0.0000	0.0005

Table 8.3: $\chi_{SPN}(t_1)$ of the SPN in Fig. 8.1, with $\lambda_1 = 1$, $\lambda_2 = 1$, for different values of λ_3 and k.

k	2	4	8
λ_3 = 0.1	0.3623	0.2666	0.1822
λ_3 = 1	0.1666	0.0468	0.0033
λ_3 = 10	0.0078	0.0000	0.0000

However, in the case of the TCPN, $\chi_{TCPN}(t_1)$ is independent of k, as it can be seen in the second row in Table 8.1.

The schema of this example is important in practice. It appears when an inhibitor arc is removed by the system with the addition of a *complementary* place which usually has this structure.

Table 8.4: $\chi_{SPN}(t_1)$ of the SPN in Fig. 8.1, with $\lambda_2 = 1$, $\lambda_3 = 1$, for different values of λ_1 and k.

k	2	4	8
λ_1 = 0.1	0.0238	0.0060	0.0004
λ_1 = 1	0.1666	0.0468	0.0033
λ_1 = 10	0.4166	0.1442	0.0154

8.2 The ρ-$semantics$ for transitions with one input place

In order to tackle the BRP, we will start with the the simple but representative example in Fig. 8.2(a), in which p_1 has only one output transition, $|p_1^\bullet|$ = 1. In other words, transition t_3 is dropped from Fig. 8.1.

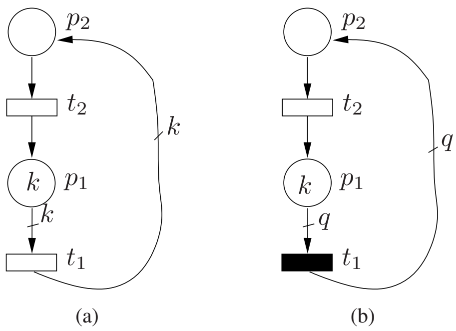


Figure 8.2: (a) SPN system in which t_1 suffers from the BRP, its firing rates are $\boldsymbol{\lambda} = (10, 1)$. (b) Hybrid PN system in which t_1 is discrete (black transition) and t_2 is continuous, arc weights are modified to q.

Consider the PN in Fig. 8.2(a) as a discrete PN. In the steady state, the total cycle time Θ from $\boldsymbol{m}_0 = (k, 0)$ is the addition of the average time to fire t_1 from $\boldsymbol{m}_0$ (which happens with a mean value of $\frac{1}{\lambda_1}$), the mean time to fire t_2 when $m[p_2] = k$ (which is $\frac{1}{k \cdot \lambda_2}$), the one when $m[p_2] = k-1$ (which is $\frac{1}{(k-1)\cdot\lambda_2}$), etc. Thus, the average cycle time is $\Theta = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} \cdot \sum_{i=1}^{k} \frac{1}{i}$. The mean throughput of t_1 is equal to $\frac{1}{\Theta}$ (see row "SPN" in Table 8.5):

$$\chi_{SPN}(t_1) = \frac{\lambda_1 \cdot \lambda_2}{\lambda_1 \cdot \sum_{i=1}^{k} \frac{1}{i} + \lambda_2} = \lambda_2 \cdot \frac{1}{\sum_{i=1}^{k} \frac{1}{i} + \frac{\lambda_2}{\lambda_1}} \tag{8.2}$$

Considered as TCPN, t_1 is enabled for any marking $m[p_1] > 0$. Moreover, the firing of t_1, and hence the behaviour of the system, is not modified by k. The throughput of t_1 as TCPN is *time homothetic* (i.e., its steady state flow is proportional to $\boldsymbol{\lambda}$), and it is equal to: $\chi_{TCPN}(t_1) = \lambda_2/(1 + \frac{\lambda_2}{\lambda_1})$.

Table 8.5: $\chi(t_1)$ in the PN in Fig. 8.2(a), with $\boldsymbol{\lambda} = (10, 1)$, for different methods.

Method	k = 1	k = 2	k = 3	k = 4	k = 5	...	k=10	k=50	k=100
SPN	0.909	0.625	0.517	0.458	0.420	...	0.330	0.217	0.189
TCPN	0.909	0.909	0.909	0.909	0.909	...	0.909	0.909	0.909
Meth1	0.909	0.729	0.607	0.522	0.459	...	0.296	0.090	0.051
Meth2, 2τ	0.545	0.545	0.545	0.545	0.545	...	0.545	0.545	0.545
Meth2, 2τ	0.475	0.475	0.475	0.475	0.475	...	0.475	0.475	0.475
Meth2, 3τ	0.324	0.324	0.324	0.324	0.324	...	0.324	0.324	0.324

Considering $\boldsymbol{\lambda} = (10, 1)$, $\chi_{TCPN}(t_1) = 0.909$ for any value of k. The continuous throughput coincides with the discrete one for $k = 1$, but it provides a bad approximation for $k > 1$ (see Table 8.5), which gets worse when k grows. This difference among continuous and discrete PN is the root of the problem.

8.2.1 First approaches to the Bound Reaching Problem

After identifying the BRP, our aim is to find methods or techniques to approach it, i.e., to fluidify the net system to obtain a good approximation to the original discrete one. These techniques can range from fully continuous to hybrid.

Let us describe two very basic methods which can be used to approach the BRP. The first one is a continuous approximation with a firing semantics different from the usual ISS. The second one is a hybrid PN in which the transitions are continuous or discrete, but some arc weights must be modified. These methods are also described in [ZS10]. They will be compared with the ρ-$semantics$ proposed in this Section.

- **(Meth1)Ad hoc continuous flow estimation.** It is based on the heuristic idea that the k tokens are considered to be independent. The probability of $m[p_1] = 1$ in the discrete system is "considered" to be $\frac{m[p_1]}{k}$, while the probability of $m[p_1] = k$ would be $(\frac{m[p_1]}{k})^k$. Based on this heuristic reasoning, a marking-dependent flow approximation for transition t_1 can be defined as follows:

$$f(t_1) = \lambda_1 \cdot \left(\frac{m[p_1]}{k}\right)^k \tag{8.3}$$

  With this semantics, the enabling degree of t_1 is *not linear* (see Fig. 8.5(b)). The flow given by this technique has the advantages of getting continuous fully differentiable models. However, with this semantics, some properties of ISS are lost, for example the flow evolution is not marking *homothetic* (given $\boldsymbol{f}$ the flow in $\langle \mathcal{N}, \boldsymbol{m}_0, \boldsymbol{\lambda} \rangle$, the one in $\langle \mathcal{N}, k \cdot \boldsymbol{m}_0, \boldsymbol{\lambda} \rangle$ is not $k \cdot \boldsymbol{f}$).

  A drawback of this approach is that although it provides a reasonable approximation of the SPN throughput for small (see k between 1 and 5 in Table 8.5), it is not so good for high values of k (see k=50 or k=100 in Table 8.5).

- **(Meth2) Hybrid (with scaled arc weights).** Let us consider that every transition is continuous except the transition in $BRTS$, which is discrete. A *hybrid* PN (HPN) is obtained.

  Consider the PN system in Fig. 8.2(a), then t_1 is considered as discrete and t_2 as continuous). Transition t_1 can be fired from $\boldsymbol{m}_0$. However, filling again place p_1 up to capacity k "lasts forever", so t_1 can never be enabled again after the first firing. A modification of the arc weights is needed to allow the firing of t_1. The filling of p_1 is like a basic first order time invariant linear system (for example, it corresponds to the loading of a capacitor in a basic electrical RC-circuit). It is a classical result that the 63.2% of k is reached at time $\tau = RC$ (and 86.7%, 95% are reached at time 2τ, 3τ, respectively).

  Using this idea, the weight of the input and output arcs of t_1 (Fig. 8.2(b)) are modified. And the resulting throughputs of t_1, which is independent of k, are the following (also shown in Table 8.5):

  - For τ, $q = 0.632 \cdot k$, and $\chi_{HPN}(t_1) = 0.545$.
  - For 2τ, $q = 0.867 \cdot k$, and $\chi_{HPN}(t_1) = 0.475$.
  - For 3τ, $q = 0.95 \cdot k$, and $\chi_{HPN}(t_1) = 0.324$.

  A characteristic of this approach is the fact that the flow is discontinuous, being a hybrid net in the classical sense. Notice that the estimation of the steady state throughput is independent of k, but nevertheless very sensible to the choice among τ, 2τ and 3τ.

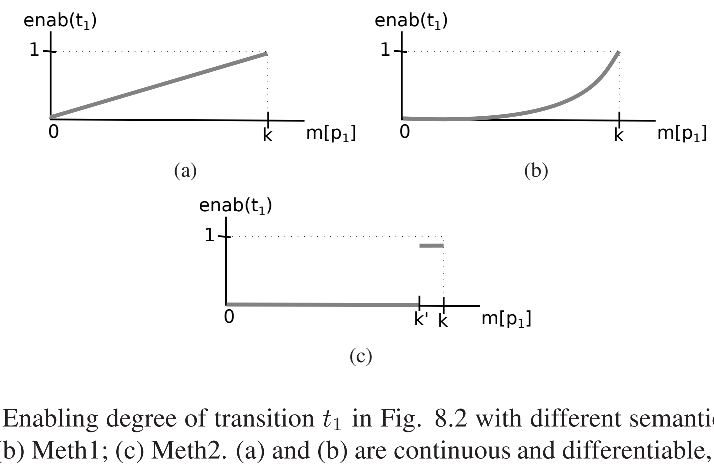


Figure 8.3: Enabling degree of transition t_1 in Fig. 8.2 with different semantics: (a) TCPN under ISS; (b) Meth1; (c) Meth2. (a) and (b) are continuous and differentiable, (c) is discontinuous, a *hybrid* net in the classical sense.

These two approaches are interesting, but just a first approach to the BRP. Hence, different techniques should be investigated.

In the following subsections, a new semantics for transitions is proposed to approach the BRP, denoted as ρ-$semantics$. The preliminary idea which inspires this semantics is to try to simulate the "wait until there are enough tokens to fire" of the discrete net. This behaviour can be obtained with timed and immediate transitions and some additional places (Section 8.2.2). An *immediate* transitions which has some tokens in its input places, fires "immediatelly" (in 0 time units). Then, the desired behaviour is obtained by the definition of the new firing semantics for the transition, as explained in Subsection 8.2.3.

8.2.2 Simulating discrete behaviour with immediate transitions

Considering the PN in Fig. 8.2(a), a key difference between the behaviour of the SPN and the TCPN is that in the SPN, t_1 can fire only when the k tokens are in p_1; while in the continuous case, it is not needed to "wait until the k tokens" are in p_1 to fire t_1.

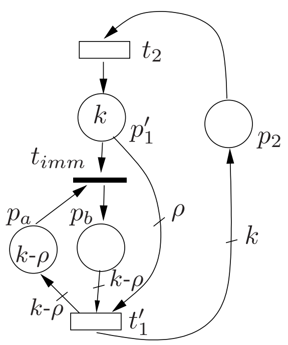


Figure 8.4: Transformation of the PN in Fig. 8.2(a). The white transitions are continuous under ISS, while the thin black transition is an *immediate* transition.

This is the idea exploited in this approach: to simulate the "wait" of t_1 until it has k tokens. As explained, waiting until p_1 has k tokens would last infinite time in a TCPN. Hence, it has no sense to wait until k, but until some other smaller value, such as $k - \rho$ (where ρ comes from "the rest"). This behaviour can be obtained by transforming p_1, t_1 (see Fig. 8.2(a)) to a subnet composed of p'_1, t'_1, p_a, p_b, t_{imm} (see Fig. 8.4), such that t'_1 is not enabled for "the first" k - ρ tokens, and it is enabled for higher amounts.

Immediate transitions are difficult to handle in TCPN [RMS06]. A first approximation can be to consider immediate transitions as timed transitions which are several orders of magnitude faster than the other transitions (for example, $\lambda_{imm} = 10000$ in the TCPN in Fig. 8.4). However, this has some disadvantages: If λ_{imm} is relatively not very high, then the steady state might not be the desired one (because for large populations, t_{imm} could be part of the bottleneck); while for very high values of λ_{imm}, *stiffness* problems can appear.

8.2.3 Defining the ρ-$semantics$

In this concrete construction, we can abstract the structure given by p'_1, t'_1, p_a, p_b, t_{imm} by a unique transition with a new semantics, which compacts the desired behaviour.

Let us detail the marking of the new structure (places p'_1, p_a, p_b) and the enabling degree of t'_1, with respect to the possible marking of p_1 in the original net, i.e., $m[p_1]$:

- $m[p_1] \leq k - \rho$. Then transition t_{imm} is enabled, the marking is moved *immediately* to p_b and there is no remaining token at p'_1. Hence, $m[p'_1] = 0$, $m[p_a] = k - \rho - m[p_1]$, $m[p_b] = m[p_1]$ and $enab(t'_1) = 0$.

- $k - \rho < m[p_1] \leq k$. Then p_a is empty, transition t_{imm} is disabled, and there is some remaining token at p'_1. Hence, $m[p_a] = 0$, $m[p_b] = k - \rho$, $m[p'_1] = m[p_1] - (k - \rho)$ and $enab(t'_1) = min\{\frac{m[p_b]}{k-\rho}, \frac{m[p'_1]}{\rho}\} = \frac{m[p'_1]}{\rho} = \frac{m[p_1]-(k-\rho)}{\rho}$.

The difference between the enabling of t_1 and t'_1 is depicted in Fig. 8.3(a) and Fig. 8.5: t_1 is enabled for $m[p_1] > 0$, but t'_1 is enabled for $m[p_1] > k - \rho$.

The new transition t'_1 has a specific firing semantics, different from ISS ($f_1 = \lambda_1 \cdot enab(t_1)$), obtained from $\lambda_1 \cdot enab(t'_1)$. The flow of a transition t_1 under the ρ-$semantics$ is given by the following formula:

$$f_1 = \begin{cases} 0 & \text{if } m[p_1] \leq Pre[p_1, t_1] - \rho \\ \lambda_1 \cdot \frac{m[p_1]-(Pre[p_1,t_1]-\rho)}{\rho} & \text{otherwise} \end{cases} \tag{8.4}$$

The transient flow of t_1 is still a continuous function, but it is piecewise defined, introducing certain "hybridization" in the behaviour of the transition. The computation done by this approach is local t_1 and it is simple and fast to calculate.

The evolution of the flow of transition t_1 in Fig. 8.2(a) with the ρ-$semantics$ is illustrated in Fig. 8.6, where it is compared with the transient flow of ISS.

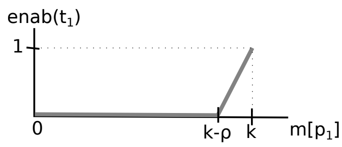


Figure 8.5: Enabling degree of transition t_1 in Fig. 8.2(a) with the ρ-$semantics$. Certain "hybrid" behaviour is obtained in the firing of the transition, the flow is a piecewise function.

8.2.4 Selection of an appropriate ρ

With the proposed ρ-$semantics$, the throughput of the system can be "tuned" from 0 (when $\rho \sim 0$) to the throughput of the TCPN (when ρ is equal to $Pre[p_1, t_1]$).

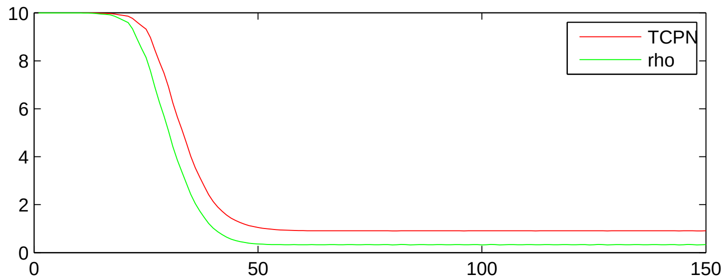


Figure 8.6: Transient evolution of the flow of t_1, as TCPN and as ρ-$semantics$.

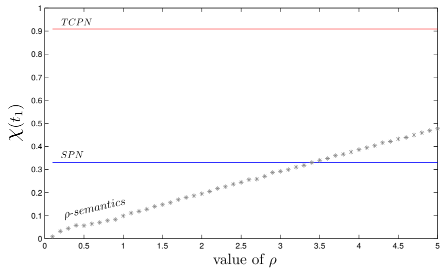


Figure 8.7: Considering the PN system in Fig. 8.2(a) with $k = 10$, the grey points represent the steady state throughput obtained by the ρ-$semantics$ for different values of ρ. The blue line corresponds to $\chi_{SPN}(t_1)$, and the red one to χ_{TCPN}.

For example, consider the PN system in Fig. 8.2(a) with $k = 10$. The steady state throughput of t_1 is equal to 0.330 as SPN, while it is equal to 0.909 as TCPN under ISS (see Table 8.5). Figure 8.7 shows the steady state throughput of t_1 obtained if the ρ-$semantics$ is applied to t_1, for different values of ρ.

The challenge is how to select ρ to approximate the steady state throughput of the SPN. Here we compute ρ first for the PN in Fig. 8.2(a), and then apply that *heuristics* on any PN system in which the transition t_1 which suffers from the BRP has the same structure: t_1 has only one input place, which has only one input and one output transition, $|^\bullet p_1| = |p_1{}^\bullet| = 1$.

Considering the ρ-$semantics$ for t_1, and ISS for t_2, the throughput of t_1 at the steady state can be calculated.

Some notions for the analytical computation of $\chi_\rho(t_1)$ for the PN in Fig. 8.2(a) are presented here. In that PN system, t_1 has a ρ-$semantics$, while t_2 is a usual continuous transition, with ISS semantics.

Let $\boldsymbol{f}_{ss} = (f_{ss1}, f_{ss_2}) = (\chi_\rho(t_1), \chi_\rho(t_2))$ denote the steady state throughput of t_1 and

t_2, and let $\boldsymbol{m}_{ss}$ denote the steady state marking. At steady state, $\boldsymbol{m}_{ss}$ keeps constant, and hence it holds $\boldsymbol{C} \cdot \boldsymbol{f}_{ss} = \boldsymbol{0}$. In this example:

$$f_{ss2} = k \cdot f_{ss1} \tag{8.5}$$

Given that the net system is live, it holds that $\boldsymbol{f}_{ss} > 0$. Then, by (8.4), the steady state throughput of t_1 is:

$$f_{ss1} = \lambda_1 \cdot \frac{m_{ss}[p_1] - (Pre[p_1, t_1] - \rho)}{\rho} \tag{8.6}$$

And by the ISS semantics, the steady state throughput of t_2 is:

$$f_{ss2} = \lambda_2 \cdot \frac{m_{ss}[p_2]}{Pre[p_2, t_2]} \tag{8.7}$$

From the net structure, the following relation is obtained:

$$m_{ss}[p_1] + m_{ss}[p_2] = k \tag{8.8}$$

From (8.5), (8.6), (8.7) and (8.8), the value of $\chi_\rho(t_1)$ is:

$$\chi_\rho(t_1) = f_{ss1} = \frac{\lambda_1 \cdot \lambda_2 \cdot \rho}{\lambda_1 \cdot k + \lambda_2 \cdot \rho} = \lambda_2 \cdot \frac{\rho}{k + \rho \cdot \frac{\lambda_2}{\lambda_1}} \tag{8.9}$$

Given (8.9) for the ρ-$semantics$ and (8.2) for the SPN, forcing $\chi(t_1) = \chi_\rho(t_1)$, an analytical formula for the value of ρ is obtained, which is dependent on k (see the value of ρ for different values of k in Table 8.6):

$$\rho = \frac{k}{\sum_{i=1}^{k} \frac{1}{i}} \tag{8.10}$$

Table 8.6: Value of ρ for different values of k, obtained from equation (8.10).

k	1	2	3	4	5	10	50
ρ	1	1.33	1.64	1.92	2.19	3.41	11.9

Some values of ρ obtained from formula (8.10) for some values of k between 1 and 50 are drawn as blue points in Fig. 8.8. A linear approximation for this formula obtained with the method of least squares is $\rho = 0.1846 + 1.425 \cdot k$ (red line in Fig. 8.8), while an approximation with a potential function is $\rho = 0.6824 \cdot k^{0.7081}$ (red line in Fig. 8.8). However, in most practical cases it is enough to calculate the ρ directly from equation 8.10, since $k = Pre[p, t]$ is a known natural number.

Interestingly, the value of ρ given by (8.10) is independent of $\boldsymbol{\lambda}$. We can think about the PN in Fig. 8.2(a) as a simplification of any net with analogous (in essence, for example, a simple cycle) structure. Hence, it will be possible to use the formula of ρ calculated here as a heuristics for ρ in any system with the same structure.

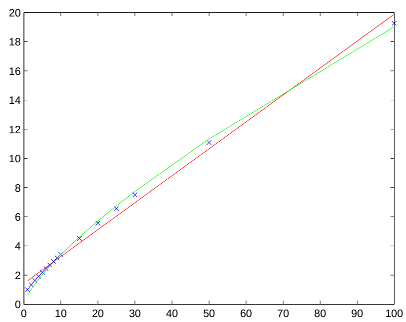


Figure 8.8: The blue points correspond with calculated values of ρ for different values of k, from equation (8.10). The red straight line corresponds to the linear approximation ($\rho = 0.1846 + 1.425 \cdot k$). The green curve corresponds to the potential approximation ($\rho = 0.6824 \cdot k^{0.7081}$).

8.2.5 Case studies

In this section, the proposed ρ-$semantics$ is applied to two case studies. It will be compared with Meth1 and Meth2, the two methods described in Section 8.2.1.

Example 8.1 ***Manufacturing system***

Consider the PN in Fig. 8.9, which represents a manufacturing system in which tables are assembled and painted, in which cooperation and synchronization relations appear. Every transition of the net has the same speed, $\boldsymbol{\lambda} = \mathbf{1}$.

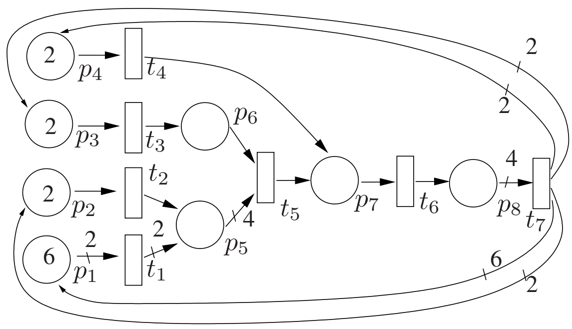


Figure 8.9: Example 8.1. PN system which models a manufacturing system which assembles tables (derived from [RS01]).

In this PN system, we will check if any its enabling bounds is near to 1. In this example, $SEB(t_1) = 3$, $SEB(t_2) = SEB(t_3) = SEB(t_4) = SEB(t_5) = SEB(t_6) = 2$ *and* $SEB(t_7) = 1$. *Hence,* $BRTS = \{t_7\}$, *i.e, transition* t_7 *suffers from the BRP, and which the structure identified in Section 8.2. If seen as a SPN system, the throughput of the output transition of the system,* t_7, *is equal to 0.6573 (see the first row in Table 8.7). However, when the system is fluidified, the throughput of* t_7 *in the TCPN is equal to 1.1429, which is not very accurate.*

Applying the ρ-$semantics$ *to transition* t_7. *From (8.10), we set* $\rho = \frac{4}{1+1/2+1/3+1/4} = 1.92$. *The obtained throughput is shown in the last row in Table 8.7:* $\chi_\rho(t_7) = 0.6443$, *which in comparison with the other methods, is the best approximation of the original SPN system.*

An interesting issue is to consider not only the steady state throughput, but also to compare the transient behaviour of the TCPN under ISS and the ρ-$semantics$. *They are compared in Figure 8.10, from where it can be seen that the time to reach the steady state in the ISS is smaller, but the obtained steady state throughput is better with the* ρ-$semantics$.

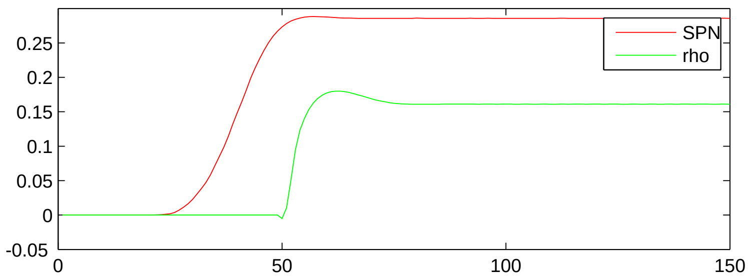


Figure 8.10: Example 8.1. Transient evolution of the flow of t_7, as TCPN and as ρ-$semantics$.

Table 8.7: Throughput of the output transition, $\chi(t_7)$, in Fig. 8.9 (Example 8.1), with $\boldsymbol{\lambda} = \mathbf{1}$. Comparative of different methods.

Method	$\chi(t_7)$
SPN	0.6573
TCPN	1.1429
Meth1	0.6030
Meth2, $q = 0.623 \cdot k$	0.4409
Meth2, $q = 0.867 \cdot k$	0.2636
Meth2, $q = 0.95 \cdot k$	0.1514
ρ-$semantics$	0.6443

Other methods can be also applied to approximate the steady state throughput, as seen in Table 8.7. Meth1 and Meth2 for τ, 2τ and 3τ (with t_7 discrete and all the other transitions continuous, and weight arcs equal to $0.623 \cdot k$, $0.867 \cdot k$ and $0.95 \cdot k$ respectively) are better than TCPN.

The following example is obtained after the decolourization of a net which models a *Multi-Computer Programmable Logic Controller* (MCPLC) in [ZS10]. It is a *Generalized Stochastic PN* (GSPN) [Bal+87], which is a SPN enriched with immediate transitions (represented as thin black transitions in the figure).

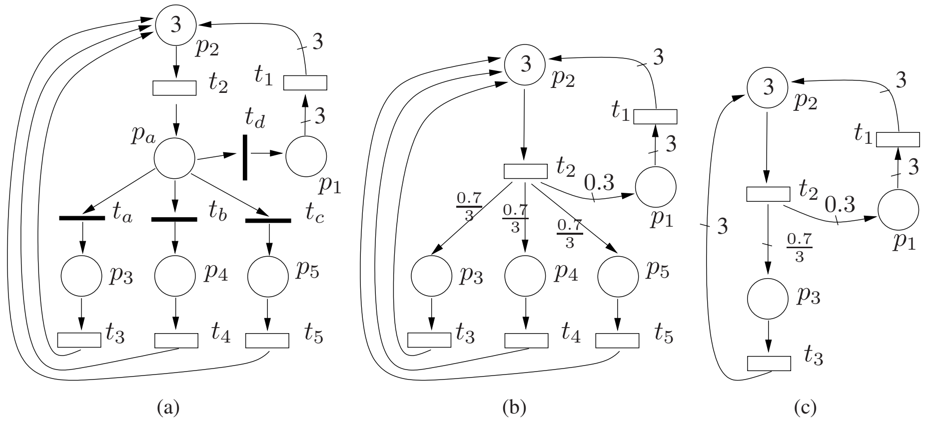


Figure 8.11: Example 8.2. (a) Discrete GSPN system which models a MCPLC [ZS10]. (b) TCPN system obtained after the fluidization of (a). (c) Reduced TCPN system.

Example 8.2 ***A Multi-Computer Programmable Logic Controller***

Consider the PN example in Fig. 8.11(a). As said, the original discrete system has some immediate *transitions. An interesting issue is how to model immediate transitions when a GSPN system is fluidified. Using the rules defined in [RMS06], the immediate transitions in Fig. 8.11(a) can be reduced: First, transitions t_a, t_b, t_c and t_d, being in topologically equal conflict relation are merged into a fork transition. Then, that transition and the timed transition t_2 are transformed into a single transition, t_2 in Fig. 8.11(b). Moreover, given that this new net has three symmetric branches, they can be reduced to only one [MS12], obtaining the TCPN system in Fig. 8.11(c).*

Let us first consider this example with the following vector of transition rates: $\boldsymbol{\lambda} = (2, 10, 5, 5, 5)$. For this $\boldsymbol{\lambda}$, the values of $\chi(t_1)$ are illustrated in the first column in Table 8.8. In this net system, $BRTS = \{t_3\}$, so the ρ-$semantics$ is applied to t_3. It can be seen that the throughput when using ρ-$semantics$, in which $\rho = \frac{3}{1+1/2+1/3} = 1.6364$, is not as good as the one of Meth1.

Table 8.8: Steady state throughput of t_2 of the PN in Fig. 8.11 (Example 8.2). Comparative of different methods.

Method	$\chi(t_2)$ $\boldsymbol{\lambda} = (2, 10, 5, 5, 5)$	$\chi(t_2)$ $\boldsymbol{\lambda}' = (100, 1, 10, 10, 10)$
GSPN	4.666	1.525
TCPN	7.693	2.796
Meth1	4.756	2.427
Meth2, $q = 0.623 \cdot k$	5.18	0.727
Meth2, $q = 0.867 \cdot k$	3.86	0.607
Meth2, $q = 0.95 \cdot k$	2.97	0.499
ρ-$semantics$	5.085	1.527

However, for different combinations of $\boldsymbol{\lambda}$, for example $\boldsymbol{\lambda}' = (100, 1, 10, 10, 10)$, the results are different (see the second column in Table 8.8). The throughput of t_2 for GSPN is equal to 1.525. In this case, Meth1 and Meth2 do not provide a good approximation for the GSPN. Nevertheless, the ρ-semantics, obtains good results for the approximation of the throughput of the system at the steady state, it is $\chi(t_2) = 1.527$.

8.3 Generalization of the ρ-$semantics$ to join transitions

The ρ-$semantics$ introduced in Section 8.2 has been designed for transitions with a unique input arc. The aim of this section is to generalize it to transitions with more than one input place (*join* transitions).

An interesting way to do it is to add a *representative* places introduced in Chapter 2 for each of the join transitions, and to apply the ρ-$semantics$ to the representative place, while ISS can be considered for the rest of places in $^\bullet t_i$.

8.3.1 Applying the ρ-$semantics$ to a representative place

As explained in Section 2.2, place r_i is added as the representative place of t_i, which is obtained as a linear combination of the places in $^\bullet t_i$. The value of ρ_i needs to be obtained with the same linear combination of the values of $\rho(Pre(p_i, t_j))$ for the places $p_j \in {^\bullet t_i}$.

For transitions in Class 2 (see Section 2.2.1), implicit place r_i has been obtained as $\boldsymbol{C}[r_i, T] = \sum_{p \in {^\bullet t_i}} \boldsymbol{C}[p, T]$ (see Section 2.2.3). The value of ρ_i should be computed in an analogous way, because it affects to the addition of the markings in $p \in {^\bullet t_i}$, and it is obtained as follows. In this formula, $\rho(x)$ corresponds to the value of ρ from equation (8.10), replacing k in (8.10) by parameter x:

$$\rho_i = \sum_{p \in {^\bullet t_i}} \rho(Pre(p, t_i)) = \frac{Pre[\pi, t_i]}{\sum_{n=1}^{Pre[\pi, t_i]} \frac{1}{n}} \qquad (8.11)$$

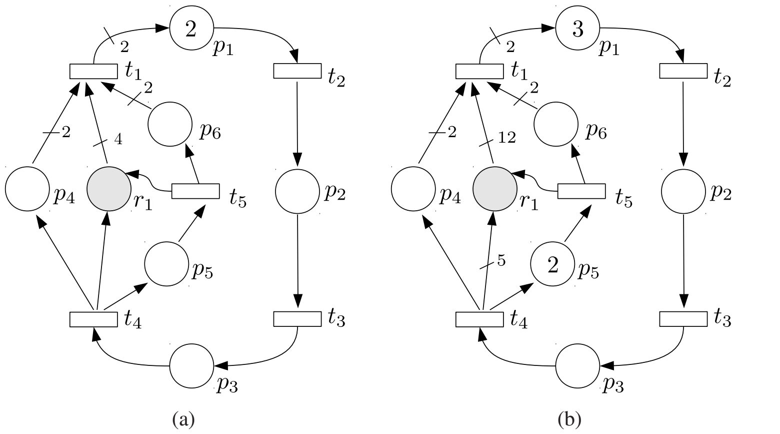


Figure 8.12: Without the grey places, PN system with two different initial markings: (a) $\boldsymbol{m}_0 = (2, 0, 0, 0, 0, 0)$ and (b) $\boldsymbol{m}'_0 = (3, 0, 0, 0, 2, 0)$. The representative place r_1 is added for transition t_1 in each discrete PN system.

For transitions in Class 3 (see Section 2.2.1), r_i has been obtained as $\boldsymbol{C}[r_i, T] = C[\pi, T] + \sum_{p \in \{^\bullet t_i \setminus \pi\}} SB(\pi) \cdot \boldsymbol{C}[p, T]$ then ρ_i is obtained as follows:

$$\rho_i = \rho(Pre[\pi, t_i]) + SB(\pi) \cdot \sum_{p \in \{^\bullet t_i \setminus \pi\}} \rho(Pre[p, t_i]) = \frac{Pre[\pi, t_i]}{\sum_{n=1}^{Pre[\pi, t_i]} \frac{1}{n}} + SB(\pi) \cdot \sum_{p \in \{^\bullet t_i \setminus \pi\}} \frac{Pre[p, t_i]}{\sum_{n=1}^{Pre[p, t_i]} \frac{1}{n}} \tag{8.12}$$

Once determined the value of ρ_i, the flow of the transition defined in equation (8.4), can be generalized to *join* transitions, in which a term analogous to the one in (8.4) is used for the *representative place*, and ISS is used for the rest of the places in $^\bullet t_i$:

$$f_i = \lambda_i \cdot \begin{cases} 0 & \text{if } m[r_i] \leq Pre[r_i, t_i] - \rho_i \\ min\{\frac{m[r_i] - (Pre[r_i, t_i] - \rho_i)}{\rho_i}, \min_{p \in ^\bullet t_i}\{\frac{m[p]}{Pre[p, t_i]}\}\} & \text{otherwise} \end{cases} \tag{8.13}$$

Table 8.9: $\chi(t_1)$ of the PN in Fig. 8.12, with $\boldsymbol{\lambda} = (10, 1, 1, 1, 1)$.

Method	$\chi(t_1)$, with $\boldsymbol{m}_0 = (2, 0, 0, 0, 0)$	$\chi(t_1)$, with $\boldsymbol{m}'_0 = (3, 0, 0, 2, 0)$
SPN	0.193	0.396
TCPN	0.244	0.484
ρ-*semantics*	0.195	0.410

Example 8.3 *Consider transition t_1 in the PN in Fig.8.12(a). Transitions t_2, t_3, t_4 and t_5 have a unique input place, hence In this case, ${}^\bullet t_1 = \{p_1, p_2\}$, so $|{}^\bullet t_1| > 1$ and the ρ-$semantics$ cannot be directly applied. The first step is add the* representative place *r_1 which represents the marking of the input places of t_1 (see Section 2.2.3). Then, the value of ρ_1 can be obtained from equation (8.11). The ρ-$semantics$ proposed in equation (8.13) can be applied to t_1. In an analogous way, the ρ-$semantics$ is applied to the PN in Fig.8.12(b), using equation (8.12) for the value of ρ_1.*

Considering $\boldsymbol{\lambda} = (10, 1, 1, 1, 1)$ and $\boldsymbol{m}_0$ (see first column in Table 8.9), the ρ-$semantics$ ($\chi_\rho(t_1) = 0.195$) gives a better approximation than the TCPN ($\chi_{TCPN}(t_1) = 0.244$) to the original SPN ($\chi_{SPN}(t_1) = 0.193$). In this case, if (8.10) instead of (8.11) would be used, a throughput value 0.121 would be obtained, which is not so accurate. Also for $\boldsymbol{m}'_0$ the approximation given by the ρ-$semantics$ is better than the TCPN (see second column in Table 8.9). If equation (8.10) would be used, a throughput equal to 0.309 *would be obtained, which is not accurate either.*

8.3.2 Case study

In this section, a case study is considered, in which the BRP appears in a join transition. A representative place is added to the system and ρ-$semantics$ is applied.

Example 8.4 ***A manufacturing system***

Consider now the PN in Fig. 8.13, the model of a manufacturing system in which tables are assembled. Consider $\boldsymbol{\lambda} = (10, 10, 10, 1, 1, 1, 10, 1)$.

The BRP appears in t_7. A representative *place r_7 is added. In this case, $SB(p_7) = Pre[p_7, t_7]$. The implicit place r_7 is built as $\boldsymbol{C}[r_7, T] = SB(p_8) \cdot \boldsymbol{C}[p_7, T] + \boldsymbol{C}[p_8, T]$, as explained in Section 2.2.3.*

The throughput of the output transition for the different methods is presented in table 8.10. In Meth2, transition t_8 is the one considered as discrete. $\boldsymbol{\lambda} = (20, 4, 4, 1, 1, 5, 5, 1, 1)$ and $\boldsymbol{\lambda}' = (10, 10, 10, 1, 1, 1, 10, 1)$.

Table 8.10: Steady state throughput of t_8 of the PN in Fig. 8.13 (Example 8.4). Comparative of different methods.

Method	$\chi(t_8), \boldsymbol{\lambda}$	$\chi(t_8), \boldsymbol{\lambda}'$
SPN	0.388	0.356
TCPN	0.443	0.453
Meth1	0.349	0.349
Meth2, 2τ	0.310	0.325
Meth2, 3τ	0.264	0.244
ρ-$semantics$	0.383	0.370

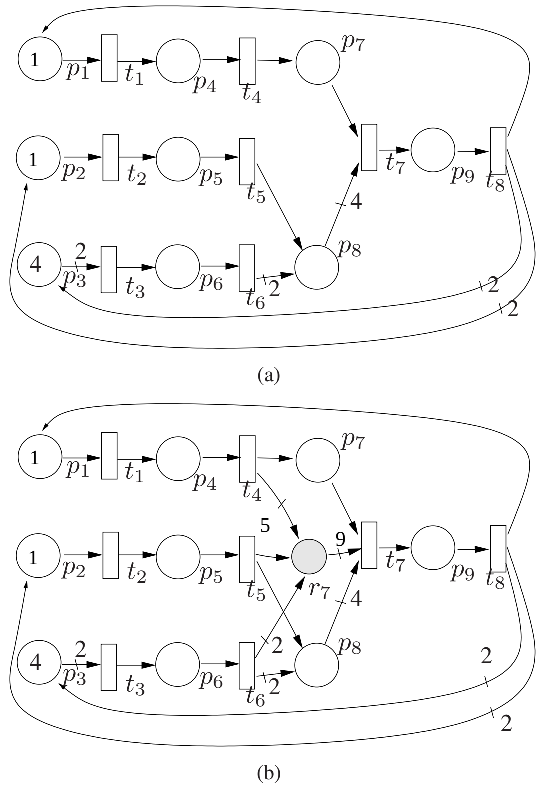


Figure 8.13: Example 8.4. (a) PN system which models a manufacturing system which assembles tables (derived from [RS01]). (b) Implicit place drawn in grey colour has been added.

8.4 Conclusions

This chapter deals with a challenging problem which appears in the fludization of discrete PN with a localized part of not very large populations, which is denoted *bound reaching problem*: although the steady state throughput of a SPN system with large populations is well approximated by its TCPN counterpart, it is not the case for small populations. It is due to the relaxation of the integrality constraints of the original PN, which is specially relevant when the marking bound of a place coincides with the weight of one of its output arcs.

Different approaches for the (partial) fluidization of this kind of systems have been described. In particular, a new semantics has been proposed, the ρ-$semantics$, in which the transition suffering the BRP is enabled only if it has a certain amount of tokens in its input place. The ρ-$semantics$ is discussed, and it is compared with the other methods. We are conscious of the fact that this is a first and partially heuristic consideration of a difficult problem that is essential to improve the quality of fluid approximation of discrete event systems.

Moreover, the ρ-$semantics$ has been generalized to more general schemas, adapting the semantics to transitions with several input places (*rendez-vous* or *join*). This generalization takes advantage of a technique developed for discrete system, the *linear enabling functions*, which are implemented by the addition of certain *representative* places.

Part IV

Hybrid adaptive Petri nets

9 Hybrid adaptive Petri nets: definition and deadlock-freeness preservation

"*It is not the strongest or the most intelligent who will survive but those who can best manage change*".
Charles Darwin (paraphrased by Leon C. Megginson)

This chapter studies the *Hybrid Adaptive Petri Nets* (HAPN), presented in the literature for timed systems, and considered here as an untimed formalism. These HAPN combine discrete and continuous behaviour from the discrete and continuous Petri nets. They attempt to partially fluidify discrete PN model in order to maintain its relevant properties.

In HAPN, the partition between continuous and discrete transitions is not *static* but *dynamic*: each transition of a HAPN can behave as discrete or as continuous depending on its load, i.e., on its enabling degree. An adaptive transition has two different modes: *continuous* and *discrete*, continuous mode for high transition load (in this case, enabling degree higher than a given threshold) and discrete in other case. In most engineering problems, the higher the workload the better the continuous approximation. Consequently, it makes sense to commute to a discrete mode when the load becomes low. The HAPN formalism provides a common framework that includes three well known formalisms: discrete, continuous and hybrid PN. Thus, the modelling power of HAPN subsumes the modelling power of the other formalisms, and any analysis technique applicable to HAPN can also be applied to the others. It captures in a compact way the fact that the behaviour of a highly loaded system can be approximated by continuous dynamics while the behaviour of a system with low loads usually require explicit discrete dynamics. The HAPN formalism is intended to solve the existing trade-off between the high complexity and accuracy of discrete PN and the low-complexity but potential loss of accuracy associated to continuous PN. Moreover, it allows to preserve some properties of the discrete PN systems that are not directly preserved by continuous ones.

Introduction

This chapter introduces a Petri net based formalism in which the firing of transitions is partially relaxed, *Hybrid Adaptive Petri nets* (HAPN). They were proposed for timed PN in [YLL09], and they are reinterpreted here in a untimed context. The transitions of HAPN can behave in two different modes: *continuous* and *discrete*. The continuous mode will be chosen when transition workload is higher than a given threshold. It makes sense because the higher the workload the better the continuous approximation. Consequently, it also makes sense to commute to a discrete mode when the load becomes low.

This way, a HAPN is able to *adapt* its behaviour to the net workload; it offers the possibility to represent more faithfully the discrete model and simplifies analysis techniques by behaving as continuous when the load is high. In contrast to [YLL09], HAPN will be defined and studied in the untimed framework. Notice that the introduction of time in a given system would produce a particular system trajectory that is also achievable in the untimed one. Thus, the results for some properties as deadlock-freeness in the untimed framework can be almost straightforwardly applied on timed systems.

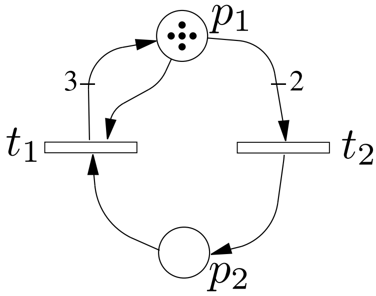


Figure 9.1: A Petri net system that deadlocks as continuous but is deadlock-free as hybrid adaptive with appropriate thresholds.

Example 9.1 *Let us consider the PN system in Fig. 9.1 [RTS99] to introduce the behaviour of HAPN. Let the initial marking of the system be* $\boldsymbol{m}_0 = (5, 0)$*. If considered as a discrete system, it is deadlock-free: from the initial marking* $\boldsymbol{m}_0$ *only* t_2 *can fire, reaching* $\boldsymbol{m}_1 = (3, 1)$*. From* $\boldsymbol{m}_1$*, both* $\boldsymbol{m}_0$ *and* $\boldsymbol{m}_2 = (1, 2)$ *can be reached by firing* t_1 *and* t_2 *respectively. This behaviour is represented in the reachability graph and reachability set in Fig. 9.2. None of the reachable markings deadlocks the system, hence it is deadlock-free.*

Consider now that the system is continuous [JRS03] , i.e., each transition can be fired in any non-negative real amount less than or equal to its enabling degree. Given that at $\boldsymbol{m}_0$ *the enabling degree of* t_2 *is* 2.5*,* t_2 *can fire in any amount in the interval* $[0, 2.5]$*. Figure 9.3 shows the reachability set of the continuous PN. The firing of* t_2 *in an amount lower than* 2.5 *produces positive markings in both places and both transitions are enabled. However, the*

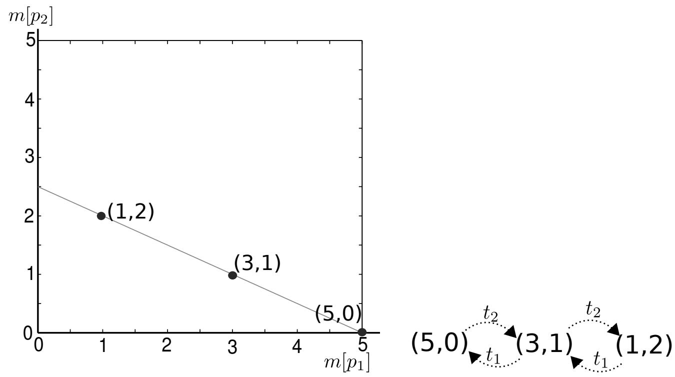


Figure 9.2: Reachability space of the PN in Fig. 9.1 considered as discrete.

firing of t_2 in 2.5 from $\boldsymbol{m}_0$ leads to $(0, 2.5)$ where no transition is enabled and the system deadlocks. Consequently, deadlock-freeness is not preserved by the continuous PN.

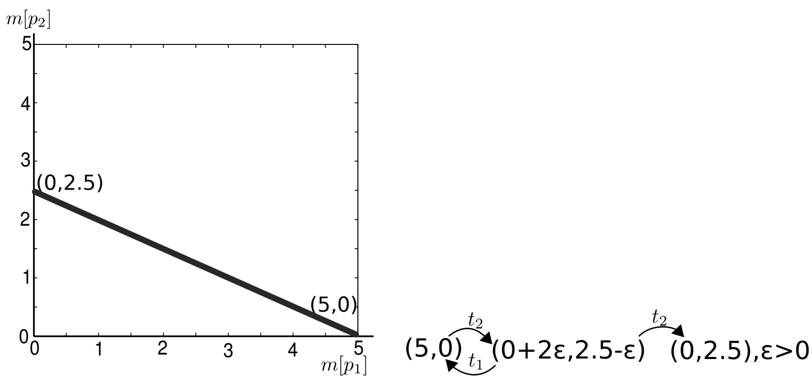


Figure 9.3: Reachability space of the PN in Fig. 9.1 considered as continuous.

Let us finally assume that the net system is adaptive. For these systems, a transition t_i can have two different firing modes: continuous *and* discrete*. It behaves as continuous when its enabling degree is higher than a given threshold μ_i. Otherwise, t_i behaves as discrete.*

When a discrete system is considered as adaptive, appropriate thresholds have to be defined. Let us define $\mu_1 = 1$ for t_1 and $\mu_2 = 1.5$ for t_2 for the system of Fig. 9.1. At the initial marking $\boldsymbol{m}_0 = (5, 0)$, t_1 is not enabled, and t_2 behaves as continuous, and it can fire in real amounts while it remains continuous. If t_2 is fired in an amount of 1, $\boldsymbol{m}_1 = (3, 1)$ is reached. At $\boldsymbol{m}_1$, both t_1 and t_2 are enabled as discrete. The firing of t_1(t_2) from $\boldsymbol{m}_1$ leads to $\boldsymbol{m}_0$($\boldsymbol{m}_2 = (1, 2)$). At $\boldsymbol{m}_2$ both transitions are discrete but only t_1 is enabled, whose firing leads to $\boldsymbol{m}_1$. Hence, although the adaptive system still keeps some continuous behaviour, it

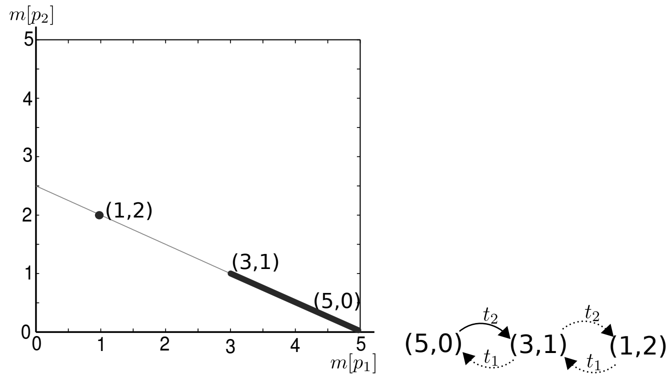


Figure 9.4: Reachability space of the PN in Fig. 9.1 considered as hybrid adaptive.

preserves the deadlock-freeness property of the discrete system. Figure 9.4 shows the reachability set of the HAPN. The arrows of the reachability graph *below the reachability set are solid for the continuous firings and dotted for the discrete ones.*

In summary, deadlock-freeness property of a discrete system might not be preserved by the continuous approximation; nevertheless, it could be preserved by the *hybrid adaptive* approximation.

In contrast to [YLL09], in this work HAPN are generalized to the *untimed* framework, allowing non determinism. The introduction of time in a given system would produce a particular *system trajectory* that is also achievable in the untimed one. Thus, some results for properties as deadlock-freeness in the untimed framework can be almost straightforwardly applied on timed systems.

Related work

Some works in the literature deal with timed elements of very different orders of magnitude, for example in the study of manufacturing systems, chemical reactions, biological systems, etc. In this kind of systems, there are very fast and very slow reactions, leading to *stiffness* problems [BCP96]. Moreover, in these systems (e.g. a genetic network) there are some species with very *small* populations (e.g. a gene), or with very *large* ones (e.g. proteins). In those cases, small populations should be modelled as discrete stochastic processes [Gil77], while large populations can be well approximated with ordinary differential equations [Har02]. In the field of PN, small populations could be modelled with *discrete (Markovian) PN* [Mar+04], while large populations could be modelled with *(timed) continuous PN* [DA10; Sil+11].

An important issue in this context is the partition of the populations into *large* ones and *small* ones (and hence, the transitions in continuous and discrete). With *static* partitions, each element of the system is modelled either as continuous or as discrete, but not both, dealing to *hybrid* PN [DA10]. With *dynamic* partitions, the elements are split during the evolution of the system. Inspired in fuzzy logic theory, the *fuzzy multimodel* is proposed in [HLEM01], where the partition is determined by a fuzzy function, i.e., a *dynamic* partition which allows some uncertainty. Another alternative is to *adapt* dynamically the partitions with a *crisp* function which uses a *threshold* associated to the transitions. HAPN, proposed in [YLL09] in a timed context and studied in [Fra+11] as untimed, deal with this modelling feature. Dynamic *adaptation* to the workload has been also considered for hybrid simulation in [Alf+05; Gri+06].

9.1 Formal definition

HAPN are a wide formalism which includes other known formalisms: discrete, continuous and hybrid PN. This point of view allows us to have an integrated view about them. Furthermore, methods and properties on HAPN may also be suitable for the particular formalisms.

This section defines basic concepts related to discrete, continuous, hybrid and hybrid adaptive PN. First, HAPN are defined in Subsection 9.1.1. Based on this general definition, the other mentioned PN formalisms are defined in the following subsection, as particular cases.

9.1.1 Hybrid adaptive Petri nets: definitions

The formalism of HAPN is formally introduced in this section.

The following example illustrates the behaviour of an adaptive transition, explaining the behaviour of a PN with just one place and one transition.

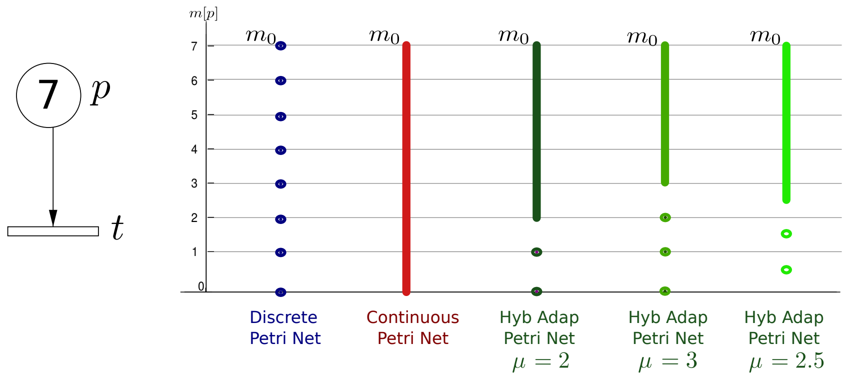
Example 9.2 *Figure 9.5 (b) explains the behaviour of the Petri net in Fig. 9.5 (a), from the initial marking* $\boldsymbol{m}_0 = (7)$*. Firstly, the possible markings reachable from the PN when it is a discrete Petri net are shown in blue color. The net starts with the initial marking* $m[p] = 7$*, and it decreases with the discrete firings of* t *until it reaches* $m[p] = 0$*.*

Secondly, the red line of the figure represents the possible reachable markings of the net when it is a continuous PN. The marking of place p *can decrease in any amount by the firing of the continuous transition* t*.*

Finally, the net is considered as HAPN. Different values of the threshold μ *are considered, and the reachable markings for each* μ *are sketched in green colour in Fig. 9.5. Notice that transition* t *has an associated threshold* μ*. When the marking of* p*, is bigger than* μ*, the firings of* t *are continuous. Otherwise, the firings are discrete. For example, when the threshold is* $\mu = 2.5$ *, the firings are continuous from* $m[p] = 7$ *to* $m[p] = 2.5$*, and it is discrete for* $m[p] \leq 2.5$*. In this example, half token will remain in* p*.*

The structure of a HAPN has the same elements of the discrete one and a threshold vector, which associates a *threshold* μ_i to each transition $t_i \in T$.

Definition 9.3 *A HAPN is a tuple* $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post}, \boldsymbol{\mu} \rangle$ *where:*



(a) (b)

Figure 9.5: Example of a Petri net system and the possible markings of the place p.

- *$P = \{p_1, p_2, ..., p_n\}$ and $T = \{t_1, t_2, ..., t_m\}$ are disjoint and finite sets of places and transitions*
- *$\mathbf{Pre}$ and $\mathbf{Post}$ are $|P| \times |T|$ sized, natural valued, incidence matrices*
- *$\boldsymbol{\mu} \in \{\mathbb{R}_{\geq 0} \cup \infty\}^{|T|}$ is the threshold vector*

Given a place (or transition) $v \in P$ (or T), its *preset*, ${}^\bullet v$, is defined as the set of its input transitions (or places), and its *postset* $v^\bullet$ as the set of its output transitions (or places). We assume all transitions have at least one input place: $\forall t \in T, |{}^\bullet t| \geq 1$.

A *marking* $\boldsymbol{m}$ of a HAPN is defined as a $|P|$ sized, non negative, real valued vector: $\boldsymbol{m} \in \mathbb{R}_{\geq 0}^{|P|}$.

A HAPN system is defined as follows:

Definition 9.4 *A HAPN system is a tuple $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$, where:*

- *$\mathcal{N}$ is a HAPN*
- *$\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ is the initial marking*

The enabling degree of a transition t_i at a marking $\boldsymbol{m}$ is defined as:

$$enab(t_i, \boldsymbol{m}) = min_{p \in {}^\bullet t_i} \left\{ \frac{m[p]}{Pre[p, t_i]} \right\} \tag{9.1}$$

The threshold μ_i of a transition t_i determines the values of the enabling degree for which the transition behaves in *continuous* (C) or in *discrete* (D) mode:

$$mode(t_i, \boldsymbol{m}) = \begin{cases} C & \text{if } enab(t_i, \boldsymbol{m}) > \mu_i \\ D & \text{otherwise} \end{cases} \tag{9.2}$$

If a transition t_i is in *continuous* mode, i.e., $enab(t_i, \boldsymbol{m}) > \mu_i$, then t_i is enabled as continuous and it can fire. On the other hand, if t_i is in *discrete* mode, i.e., $enab(t_i, \boldsymbol{m}) \leq \mu_i$, then it is enabled iff $\lfloor enab(t_i, \boldsymbol{m}) \rfloor \geq 1$. These two conditions together imply that t_i is enabled (either as discrete or continuous) iff the following expression is true:

$$(mode(t_i, \boldsymbol{m}) = C) \vee (mode(t_i, \boldsymbol{m}) = D \wedge enab(t_i, \boldsymbol{m}) \geq 1) \tag{9.3}$$

This expression is equivalent to:

$$(enab(t_i, \boldsymbol{m}) > \mu_i) \vee (enab(t_i, \boldsymbol{m}) \leq \mu_i \wedge enab(t_i, \boldsymbol{m}) \geq 1) \tag{9.4}$$

what simplifies to:

$$enab(t_i, \boldsymbol{m}) > \mu_i \vee enab(t_i, \boldsymbol{m}) \geq 1, \text{ with } \mu_i \in \mathbb{R}_{\geq 0} \tag{9.5}$$

A transition t_i that is enabled can fire. The admissible firing amounts depend on its mode:

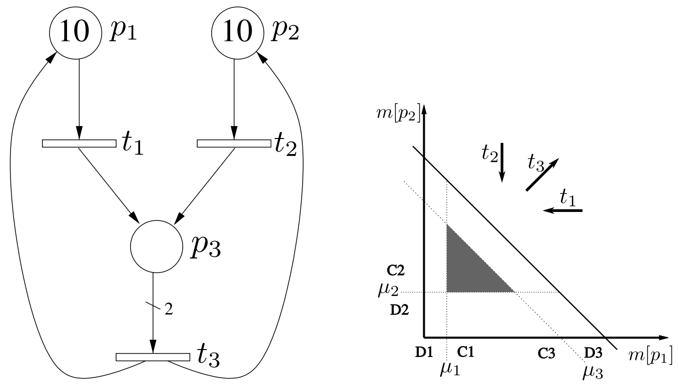
- If $mode(t_i, \boldsymbol{m}) = C$, t_i can fire in any non negative real amount $\alpha \in \mathbb{R}_{>0}$ that does not make the enabling degree cross the threshold μ_i, i.e., $0 < \alpha \leq enab(t_i, \boldsymbol{m}) - \mu_i$.
- If $mode(t_i, \boldsymbol{m}) = D$, t_i can fire as a usual discrete transition in any natural amount $\alpha \in \mathbb{N}$ such that $0 < \alpha \leq enab(t_i, \boldsymbol{m})$.

As in discrete nets (see Chapter 2), the firing of t_i from marking $\boldsymbol{m}$ in a certain amount α leads to a new marking $\boldsymbol{m}'$, and it is denoted as $\boldsymbol{m} \xrightarrow{\alpha t_i} \boldsymbol{m}'$. It holds $\boldsymbol{m}' = \boldsymbol{m} + \alpha \cdot \boldsymbol{C}[P, t_i]$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the token flow matrix (incidence matrix if $\mathcal{N}$ is self-loop free). Hence, as in discrete systems, $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, the state (or fundamental) equation summarizes the way the marking evolves, where $\boldsymbol{\sigma}$ is the firing count vector of the fired sequence. Right and left natural annullers of the token flow matrix are called T- and P-semiflows, respectively. When $\exists \boldsymbol{y} > \boldsymbol{0}$ s.t. $\boldsymbol{y} \cdot \boldsymbol{C} = \boldsymbol{0}$ the net is said to be *conservative*, and when $\exists \boldsymbol{x} > \boldsymbol{0}$ s.t. $\boldsymbol{C} \cdot \boldsymbol{x} = \boldsymbol{0}$ the net is said to be *consistent*. The *support* of a vector $\boldsymbol{v} \geq \boldsymbol{0}$ is $||\boldsymbol{v}|| = \{v_i | v_i > 0\}$, the set of positive elements of $\boldsymbol{v}$. As in DPN and CPN, the *reverse* net of $\mathcal{N}$ is defined as $\mathcal{N}^{-1} = \langle P, T, \boldsymbol{Post}, \boldsymbol{Pre}, \boldsymbol{\mu} \rangle$, in which places, transitions and thresholds coincide, and arcs are inverted.

The following example illustrates which is the behaviour mode of each transition of a given HAPN system.

Example 9.5 *Figure 9.6 (b) illustrates the behaviour of the transitions of the HAPN in Fig. 9.6 (a), with any $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$. Notice that the three arrows (t_1, t_2, t_3) of Fig. 9.6 (b) indicate the "direction" in which the marking "moves" when t_1, t_2 or t_3 are fired. The figure shows the regions in which t_1, t_2 and t_3 behave as discrete (regions D_1, D_2, D_3) or continuous (C_1, C_2, C_3). For example, t_3 behaves as continuous (C_3) below the dotted line corresponding to μ_3 and discrete above the line (D_3). In the triangular grey region of the centre of Fig. 9.6 (b), the PN behaves as continuous, and in the other regions, it has a partially discrete behaviour (some transitions behave as discrete and some as continuous).*

Figure 9.6 (c) summarizes the behaviour of each one of the transitions in the different areas identified in the Fig. 9.6 (b). For example, in the area A, t_1 and t_3 behave as discrete while t_2 behaves as continuous.



(a) (b)

Figure 9.6: (a) An hybrid adaptive Petri net system, (b) behaviour of its transitions.

9.1.2 Discrete, continuous and hybrid Petri nets as hybrid adaptive Petri nets

In general, an *adaptive* transition has both discrete and continuous behaviours, depending on its threshold and its workload. However, if very high or very low thresholds are chosen, the transition can exhibit only discrete or only continuous behaviours.

Specifically, a transition t_i whose threshold is equal to ∞ (or equal to its enabling bound), behaves always in *discrete* mode (because its enabling degree will be always lower than ∞). It will be enabled at a marking $\boldsymbol{m}$ iff $enab(t_i, \boldsymbol{m}) \geq 1$, and it can fire a natural amount α smaller or equal to its enabling degree: $0 < \alpha \leq enab(t_i, \boldsymbol{m})$.

On the other hand, a the transition t_i whose associated threshold is equal to 0, behaves always in *continuous* mode. It will be enabled at a marking $\boldsymbol{m}$ when $enab(t_i, \boldsymbol{m}) > 0$, and it can fire in any real amount α smaller than or equal to its enabling degree: $0 < \alpha \leq enab(t_i, \boldsymbol{m})$.

Using these properties, the formalisms of discrete, continuous and hybrid PN can be obtained within the formalism of HAPN.

- *Discrete Petri Net* (DPN) systems [DiC+93; Mur89] (denoted as $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ in this work) are HAPN systems in which all the thresholds are equal to "∞", i.e., all transitions fire in discrete amounts. In the case of bounded PN, it will be enough to have a threshold equal to $SEB(t_i)$ for every transition t_i.

- *Continuous Petri Net* (CPN) systems [DA10; Sil+11] (denoted as $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ in this work), which represent the full fluidization of DPN, are a particular case of HAPN in which all the thresholds are equal to 0.

- *Hybrid Petri Net* (HPN) systems [DA10] (denoted as $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_H$ in this work) are partially fluidified Petri nets in which the set of transitions T is partitioned into two sets, T^c and T^d, such that if $t_i \in T^d$ then t_i always behaves as discrete (as HAPN, its threshold μ_i is equal to ∞), and if $t_i \in T^c$ then t_i always behaves as continuous (as HAPN, its threshold μ_i is equal to 0).

In conclusion, $\boldsymbol{\mu}$ defines a DPN if $\boldsymbol{\mu} = \infty$; $\boldsymbol{\mu}$ defines a CPN if $\boldsymbol{\mu} = \mathbf{0}$; and $\boldsymbol{\mu}$ defines a HPN if $\boldsymbol{\mu} \in \{0, \infty\}^{|T|}$.

9.2 Alternative approaches to represent *adaptation*

HAPN proposes the concept of *adaptive* transitions, in which each transition can adapt its behaviour to its workload. In this subsection, modelling adaptive transitions with other known formalisms is considered.

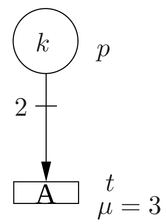


Figure 9.7: HAPN, with thresholds associated to transitions.

A first approximation to the problem would be to add p', the complementary place of p, which exists for bounded PN. For the example, consider that $SB(p) = 100$. For every adaptive transition, two transitions are needed: a continuous and a discrete one. Then, add some self loops to the discrete and the continuous transitions, as presented in Fig. 9.8. However, it would not model the behaviour of the adaptive transition because the continuous would be enabled for any marking $m[p] > 0$, not only when the marking is bigger than the threshold.

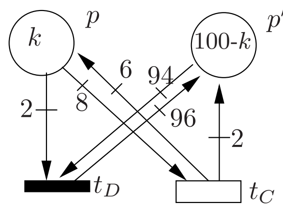


Figure 9.8: This HPN does not behave as the of the HAPN in Fig. 9.7.

Two more complex transformations are considered above, in which the adaptive transitions are modelled with HPN with inhibitor arcs or with HPN with discrete priority.

9.2.1 Hybrid Petri nets with inhibitor arcs

In the case of bounded PN, the behaviour of the HAPN can obtained with Hybrid Petri nets with inhibitor arcs, which are defined below.

Definition 9.6 *A HPN with inhibitor arcs (HPNin) is a tuple $\mathcal{N} = \langle P, T, Pre, Post, I \rangle$ where P is the set of places, $T = T^c \cup T^d$ is the set of continuous and discrete transitions, Pre and $Post$ indicates the arcs of the net, and I indicates the inhibitor arcs.*

The inhibitor arcs are not defined as classical inhibitor arcs of the discrete Petri nets, but as *threshold tests* [DA10]. Given a *threshold test arc* that goes from place p_i to transition t_i, labeled with s, transition t_i can be enabled only when $m[p_i] < s$.

The *adaptation* to the workload of the transitions can be obtained with discrete and continuous transitions (HPN) which are enabled or disabled depending on the workload. To obtain it, each place p of the HAPN has to be "unfolded" into two places: p, with its original marking, and p', with its complementary marking $B - m[p]$, where B is the bound of the place p. Each transition t of the HAPN is also "unfolded" into two transitions, a discrete one, t_D, and a continuous one, t_C. The needed arcs and inhibitor arcs needed to obtain the *adaptive* behaviour are shown in Fig. 9.9.

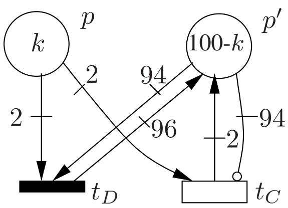


Figure 9.9: Representation of the HAPN in Fig. 9.7 with HPN with inhibitor arcs.

9.2.2 Hybrid Petri nets with *discrete priority*

Finally, in the case of bounded PN, the behaviour of the HAPN can also be obtained with HPN with *discrete priority*. It is, the Hybrid Petri nets in which discrete transitions have more priority than continuous ones. Conflict resolution between continuous and discrete transitions can be managed with different rules, being this one a common assumption [DA10] (Page 223. Rule 6.1):

If there is a conflict between a discrete and a continuous transition, the discrete transition takes priority over the continuous transition.

In this case, the behaviour of the place and the transition of the HAPN in Fig. 9.7 can be obtained with the HPN in Fig. 9.10. Due to discrete transitions have more priority than continuous, transition t_C in Fig. 9.10 will be enabled only in the case p_{C} is marked, it is the enabling degree of the transition ($\frac{m[p]}{Pre(p,t)}$) is bigger than the threshold. Complementary, transition t_D is enabled only then ($\frac{m[p]}{Pre(p,t)}$) is smaller or equal to the threshold (and p_{D} is marked).

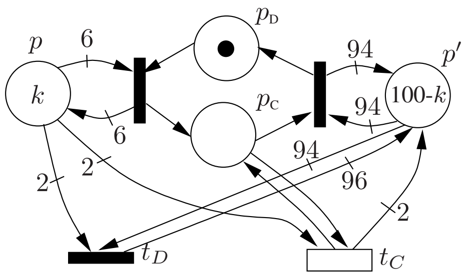


Figure 9.10: Representation of the HAPN in Fig. 9.7 with HPN *discrete priority*.

Notice that the definition of HPN where the places are indistinguishable is considered here. If the definition of HPN where the places are divided in two sets ($P = P^c \cup P^d$) were used, then places p_{D} and p_{C} would be in P^c and p, p' would be in P^d. And it would have the same behaviour.

9.2.3 Petri nets with guards enabling transitions

The concept of adaptation can be also obtained with transitions with guards. In this Petri nets, each transition has an associated guard that is a logic assertion such that if the assertion is not true, the transition is not enabled. When the assertion is true, the transition behaves normally.

Using this concepts, HAPN can be expressed with Hybrid Petri nets and guards associated to transitions. Each *adaptive* transition will be *unfolded* into two guarded transitions: a discrete transition, with an associated guard that enables the transition when $enab(t, m) \leq \mu$; and a continuous transition, whose guard enables the transition when $enab(t, m) > \mu$.

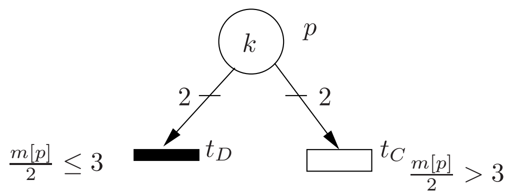


Figure 9.11: HAPN. Guards enabling transitions.

Figure 9.11 represents the same hybrid adaptive PN of Fig. 9.7 but with discrete (t_D) and continuous (t_C) guarded transitions. The discrete transition t_D will be enabled when $\frac{m[p]}{2} \leq 3$, and otherwise continuous transition t_C will be enabled.

9.3 Reachability inclusion among different formalisms

In this section, the reachability set (RS) of HAPN systems is studied and compared to the RS of discrete and continuous systems.

In order to compare the reachability sets, the same initial marking $\boldsymbol{m}_0 \in \mathbb{N}^{|P|}$ is considered for all three types of Petri nets (discrete, continuous or hybrid adaptive).

For the study of the RS we will focus on ordinary PN. Notice that although ordinary PN are a subclass of general PN, any non-ordinary Petri net can be converted to an equivalent ordinary PN [Sil85]. It will be proved that, under rather general conditions, the RS of a HAPN, $RS_A(\mathcal{N}, \boldsymbol{m}_0)$, contains the discrete RS, $RS_D(\mathcal{N}, \boldsymbol{m}_0)$. Moreover, the RS of a CPN, $RS_C(\mathcal{N}, \boldsymbol{m}_0)$, contains the RS of any HAPN with the same PN structure. This is a straightforward consequence of the fact that, in contrast to continuous nets, HAPN are a partial, non-full, relaxation of discrete nets.

Theorem 9.7 $RS_D(\mathcal{N}, \boldsymbol{m}_0) \subseteq RS_A(\mathcal{N}, \boldsymbol{m}_0)$ *for any ordinary* $\mathcal{N}$*, with* $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$.

Proof. Let $\boldsymbol{m} \in RS_D(\mathcal{N}, \boldsymbol{m}_0)$. Then, there exists $\sigma_d = t_{\gamma_1} \ldots t_{\gamma_k}$ such that $\boldsymbol{m}_0 \stackrel{1t_{\gamma_1}}{\longrightarrow} \boldsymbol{m}_1 \stackrel{1t_{\gamma_2}}{\longrightarrow} \boldsymbol{m}_2 \cdots \stackrel{1t_{\gamma_k}}{\longrightarrow} \boldsymbol{m}_k = \boldsymbol{m}$ in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$. We will prove that there exists a sequence $\sigma_a = \beta_1 t_{\gamma_1} \ldots \beta_k t_{\gamma_k}$ such that $\boldsymbol{m}_0 \stackrel{\beta_1 t_{\gamma_1}}{\longrightarrow} \boldsymbol{m}_1 \stackrel{\beta_2 t_{\gamma_2}}{\longrightarrow} \boldsymbol{m}_2 \cdots \stackrel{\beta_k t_{\gamma_k}}{\longrightarrow} \boldsymbol{m}_k = \boldsymbol{m}$ in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$.

Let us start with t_{γ_1}, and let us check if $\beta_1 = 1$ can be chosen. Two cases must be considered.

a) $enab(t_{\gamma_1}, \boldsymbol{m}_0) \leq \mu_{t_{\gamma_1}}$. From the definition of HAPN, t_{γ_1} behaves as discrete, i. e., $mode(t_{\gamma_1}, \boldsymbol{m}_0) = D$. Given that t_{γ_1} is enabled in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$, it holds that $enab(t_{\gamma_1}, \boldsymbol{m}_0) = min_{p \in {}^\bullet t_{\gamma_1}} \{m_0[p]\} \geq 1$. Hence, it is also enabled in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ in the same amount. Therefore, $\beta_1 = 1$ can be chosen, and the same $\boldsymbol{m}_1$ of the discrete system is reached.

b) $enab(t_{\gamma_1}, \boldsymbol{m}_0) > \mu_{t_{\gamma_1}}$. From the definition of HAPN, t_{γ_1} behaves as continuous, i.e., $mode(t_{\gamma_1}, \boldsymbol{m}_0) = C$. Because $\mu_{t_{\gamma_1}} \in \mathbb{N}$ and $enab(t_{\gamma_1}, \boldsymbol{m}_0) > \mu_{t_{\gamma_1}}$, it holds that $enab(t_{\gamma_1}, m_0) - \mu_{t_{\gamma_1}} \geq 1$. Therefore, $\beta_1 = 1 \leq enab(t_{\gamma_1})$ - $\mu_{t_{\gamma_1}}$ can be chosen and $\boldsymbol{m}_1$ is reached.

The same reasoning can be applied to the rest of the transitions in the sequence $t_{\gamma_2} \ldots t_{\gamma_k}$. ■

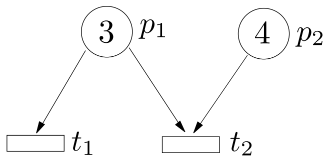


Figure 9.12: A PN system whose reachability set as discrete is not contained in the reachability set as adaptive with with $\boldsymbol{\mu} = (1.5, 1.5)$, see Fig. 9.13.

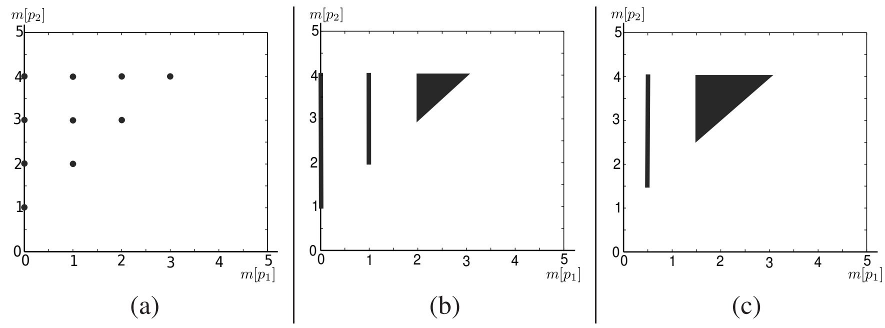


Figure 9.13: RS of the Petri Net of Fig. 9.12 behaving as: (a) DPN; (b) HAPN with $\boldsymbol{\mu} = (2, 2)$; and (c) HAPN with $\boldsymbol{\mu} = (1.5, 1.5)$.

If non ordinary PN or non natural thresholds are considered, $RS_D(\mathcal{N}, \boldsymbol{m}_0)$ is in general not contained in $RS_A(\mathcal{N}, \boldsymbol{m}_0)$. Let us show both cases through examples.

When non natural thresholds, $\boldsymbol{\mu} \notin \mathbb{N}^{|T|}$, are considered, $RS_D(\mathcal{N}, \boldsymbol{m}_0)$ is in general not contained in $RS_A(\mathcal{N}, \boldsymbol{m}_0)$ for ordinary HAPN. Let us show it with the following example. Consider the net of Fig. 9.12 as discrete with the initial marking $\boldsymbol{m}_0$ = (3,4). Both t_1 and t_2 can be fired until the place p_1 is empty (when enabling degree is 0). Its reachability set $RS_D(\mathcal{N}, \boldsymbol{m}_0)$ is represented in Fig. 9.13 (a). Let us consider now the net as adaptive, with $\boldsymbol{\mu} = (1.5, 1.5)$ Thus, t_1 can fire as continuous while $m[p_1] > 1.5$. And t_2 can fire as continuous while $m[p_1] > 1.5$ and $m[p_2] > 1.5$. When $m[p_1] = 1.5$, t_1 changes from continuous to discrete, and it can fire a discrete amount. Analogously, t_2 changes to discrete and can fire as discrete when $m[p_1] = 1.5$. Its reachability set is shown in Fig. 9.13 (c). Notice that $RS_D(\mathcal{N}, \boldsymbol{m}_0)$ contains some markings that are not reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$. For example, the marking $\boldsymbol{m}_2$ = (1, 4) $\in RS_D(\mathcal{N}, \boldsymbol{m}_0)$, but $\boldsymbol{m}_2 \notin RS_A(\mathcal{N}, \boldsymbol{m}_0)$.

If non-ordinary PN are considered, $RS_D(\mathcal{N}, \boldsymbol{m}_0)$ is in general no contained $RS_A(\mathcal{N}, \boldsymbol{m}_0)$, with $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$. This can be shown through an example. The reachability set of the HAPN in Fig. 9.1 with $\boldsymbol{\mu} = (1, 1)$ is shown in Fig. 9.14. Transition t_2 is enabled as continuous from marking $(5, 0)$ to $(2, 1.5)$, where it changes to discrete. If t_2 is fired as discrete (from $(2, 1.5)$), $(0, 2.5)$ is reached. In $(0, 2.5)$ none of the transitions are enabled (and the net deadlocks). Transition t_1 is enabled as continuous from $(2, 1.5)$ to $(3, 1)$, where it is enabled as discrete. When t_1 is fired as discrete from $(3, 1)$, $(5, 0)$ is reached and t_1 becomes not enabled.

The marking $\boldsymbol{m} = (1, 2)$ is reachable in the discrete Petri net, but not in the adaptive one with $\forall \mu, \mu = 1$. Therefore, $RS_D(\mathcal{N}, \boldsymbol{m}_0)$ is not, in general, included in $RS_A(\mathcal{N}, \boldsymbol{m}_0)$ with $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$ for non ordinary HAPN.

On the other hand, it is straightforward to prove that, given that HAPN allow real-valued markings, $RS_A(\mathcal{N}, \boldsymbol{m}_0)$ is not, in general, included in $RS_D(\mathcal{N}, \boldsymbol{m}_0)$. Nonetheless, if $\boldsymbol{\mu} = \infty$, the HAPN always behaves as discrete and its RS is trivially identical to that of the discrete PN.

Let us finally compare the RS of the HAPN to the RS of its associated continuous PN.

Theorem 9.8 $RS_A(\mathcal{N}, \boldsymbol{m}_0) \subseteq RS_C(\mathcal{N}, \boldsymbol{m}_0)$ *with* $\boldsymbol{\mu} \in \mathbb{R}^{|T|}_{\geq 0}$.

Proof. Let $\boldsymbol{m} \in RS_A(\mathcal{N}, \boldsymbol{m}_0)$. Therefore, there exists $\sigma_a = \beta_1 t_{\gamma_1} \ldots \beta_k t_{\gamma_k}$ such that $\boldsymbol{m}_0 \xrightarrow{\beta_1 t_{\gamma_1}} \boldsymbol{m}_1 \xrightarrow{\beta_2 t_{\gamma_2}} \boldsymbol{m}_2 \cdots \xrightarrow{\beta_3 t_{\gamma_k}} \boldsymbol{m}_k = \boldsymbol{m}$ *where* $\beta_i \in \mathbb{R}^+$ *if* $mode(t_{\gamma_i}, m_{i-1}) = C$ *and* $\beta_i \in \mathbb{N}^+$ *if* $mode(t_{\gamma_i}, \boldsymbol{m}_{i-1}) = D$.

For any of the β_i of σ_a, if $mode(t_{\gamma_i}, \boldsymbol{m}_{i-1}) = C$, then t_{γ_i} will be also enabled in $\langle \mathcal{N}, \boldsymbol{m}_{i-1} \rangle_C$ and the same $\beta_i \in \mathbb{R}^+$ can be chosen. If $mode(t_{\gamma_i}, m_{i-1}) = D$, then t_{γ_i} will be also enabled in $\langle \mathcal{N}, \boldsymbol{m}_{i-1} \rangle_C$ and also the same $\beta_i \in \mathbb{N}^+$ can be chosen because $\mathbb{N}^+ \subset \mathbb{R}^+$. Consequently, the same firing sequence σ_a of the HAPN system can be chosen in the continuous system and the same marking $\boldsymbol{m}$ is obtained. ■

The following corollary is straightforwardly obtained from Theorems 9.7 and 9.8.

Corollary 9.9 $RS_D(\mathcal{N}, \boldsymbol{m}_0) \subseteq RS_A(\mathcal{N}, \boldsymbol{m}_0) \subseteq RS_C(\mathcal{N}, \boldsymbol{m}_0)$ *for ordinary nets with* $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$.

Furthermore, let us show through an example that the RS of the continuous system is, in general, not contained in the RS of the HAPN system, i.e., $RS_C(\mathcal{N}, \boldsymbol{m}_0) \nsubseteq RS_A(\mathcal{N}, \boldsymbol{m}_0)$ with $\boldsymbol{\mu} \in \mathbb{R}^{|T|}$. In the PN system of Fig. 9.1 (with $\boldsymbol{\mu}$ = (1.5, 1.5)), the marking $\boldsymbol{m}$ = (0.5, 2) is included in $RS_C(\mathcal{N}, \boldsymbol{m}_0)$, but cannot be reached by the HAPN, i.e., it is not included in $RS_A(\mathcal{N}, \boldsymbol{m}_0)$. Both spaces are trivially equal if all the transitions of the HAPN always behave as continuous, i.e., when $\boldsymbol{\mu} = \boldsymbol{0}$.

9.4 Deadlock-freeness in hybrid adaptive Petri nets

This section studies the deadlock-freeness property of HAPN, and relates it to deadlock-freeness of the equivalent discrete PN. Although for arbitrary $\boldsymbol{\mu}$ deadlock-freeness of the discrete PN is, in general, not preserved by the HAPN, it is shown that the appropriate selection of $\boldsymbol{\mu}$ can preserve the property for a large class of nets.

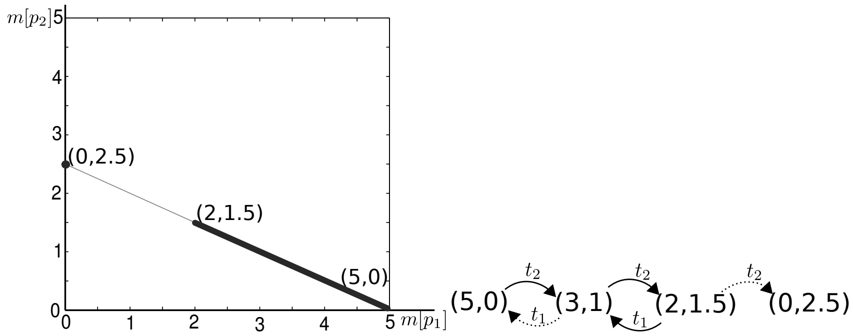


Figure 9.14: Reachability space and reachability schema of the PN in Fig. 9.1 behaving as HAPN with $\boldsymbol{\mu} = (1, 1)$.

Definition 9.10 ***Deadlock-freeness***. *A HAPN system* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ *is deadlockfree if* $\forall \boldsymbol{m} \in RS_A(\mathcal{N}, \boldsymbol{m}_0)$, $\exists t \in T$ *such that t is enabled at* $\boldsymbol{m}$.

Let us first show, by considering the net in Fig. 9.1, that:

$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is deadlock-free $\not\Rightarrow$ $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ is deadlock-free.

The system in Fig. 9.1 with $\boldsymbol{m}_0 = (5, 0)$ is deadlock-free if considered as discrete. However, if considered as HAPN with $\boldsymbol{\mu} = (1, 1)$ it deadlocks after firing t_2 as continuous in an amount of 1.5, and again t_2 as discrete, see Fig. 9.14.

Furthermore, in general, deadlock-freeness of a HAPN system does not guarantee deadlock-freeness of the equivalent discrete system:

$\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ is deadlock-free $\not\Rightarrow$ $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is deadlock-free.

The system in the Fig. 9.1 with $\boldsymbol{m}_0 = (4, 0)$ deadlocks as discrete. If considered as HAPN, it is deadlock-free with $\boldsymbol{m}_0 = (4, 0)$ and $\boldsymbol{\mu} = (1.5, 1.5)$ because t_2 commutes from continuous to discrete when $m[p_1] = 3$, and $m[p_1]$ never empties.

Although the deadlock-freeness property of discrete systems is not preserved in general by HAPN with arbitrary $\boldsymbol{\mu}$, it will be proved that for choice free nets with $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$ deadlock-freeness of the HAPN system is necessary and sufficient for deadlock-freeness of the discrete system. Let us first prove that it is a sufficient condition.

Theorem 9.11 *Let* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ *be an ordinary deadlock-free HAPN system with* $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$. *Then, the discrete system* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ *is deadlock-free.*

Let us assume that the discrete $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ deadlocks at a marking $\boldsymbol{m}$. According to Theorem 9.7, marking $\boldsymbol{m}$ can be reached by $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$. Given that the net is ordinary, for every transition t, there exists $p \in {}^\bullet t$ such that $m[p] = 0$, i.e., $\boldsymbol{m}$ is a deadlock for $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$.

For the necessary condition, two technical lemmas are introduced before stating the final result. The first one states that if a sequence σ is fireable in the adaptive system, its *ceil sequence* $\lceil \sigma \rceil$ is also fireable in the discrete one.

Definition 9.12 *Let $\sigma = \alpha_1 t_{\gamma_1}\ \alpha_2 t_{\gamma_2} \ldots \alpha_k t_{\gamma_k}$ be a firing sequence of a given HAPN $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$. The ceil sequence, $\lceil \sigma \rceil$ of σ is defined as: $\lceil \sigma \rceil = \alpha'_1 t_{\gamma_1}\ \alpha'_2 t_{\gamma_2}\ \ldots\ \alpha'_k t_{\gamma_k}$ where*

$$\alpha'_i = \left\lceil \sum_{1 \le j \le i | t_{\gamma_i} = t_{\gamma_j}} \alpha_j \right\rceil - \sum_{1 \le j < i | t_{\gamma_i} = t_{\gamma_j}} \alpha'_j$$

For example, for the sequence $\sigma_1 = 0.1\ t_1\ 0.8\ t_2\ 0.1\ t_1\ 0.2\ t_1\ 0.8\ t_2$ in the HAPN of Fig. 9.6 (a), the ceil sequence $\lceil \sigma_1 \rceil$ is defined as $\lceil \sigma_1 \rceil = 1\ t_1\ 1\ t_2\ 0\ t_1\ 0\ t_1\ 1\ t_2$.

Lemma 9.13 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ be an ordinary choice-free HAPN system with $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$. If σ is a fireable sequence in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ then $\lceil \sigma \rceil$ is fireable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$.*

Let us assume without loss of generality that $\sigma = \alpha_1 t_{\gamma_1} \ldots \alpha_k t_{\gamma_k}$ and $0 < \alpha_j \le 1$ for every $j \in \{1, \ldots, k\}$. Induction on the length of σ: $|\sigma| = k$.

- Base case ($|\sigma| = 1$). Let $\sigma = \alpha_1 t_{\gamma_1}$, then $\forall p \in {}^\bullet t_{\gamma_1}, m_0[p] \ge \alpha_1$ and given that $m_0[p] \in \mathbb{N}$, it holds that $m_0[p] \ge \lceil \alpha_1 \rceil$. Thus $\lceil \sigma \rceil = \lceil \alpha_1 \rceil t_{\gamma_1}$ can be fired in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$.
- Inductive step. Assume that the Lemma holds for $|\sigma| = k$. Let us consider the $k+1$ firing, i.e., $t_{\gamma_{k+1}}$ fires in α_{k+1}. Two cases can occur:

  a) $\alpha'_{k+1} = 0$. In this case, the Lemma trivially holds.

  b) $\alpha'_{k+1} = 1$. Let $\boldsymbol{m}_i$ and $\boldsymbol{\sigma}_i$ ($\boldsymbol{m}'_i$ and $\boldsymbol{\sigma}'_i$) be the marking and firing count vector obtained just after the firing of t_{γ_i} in an amount α_i (α'_i). If $t_{\gamma_{k+1}}$ fires in the HAPN system, it means that $\boldsymbol{m}_k[p] > 0$ for every $p \in {}^\bullet t$. Notice that, by definition of ceil sequence, after the k^{th} firing the following inequalities are satisfied: $\boldsymbol{\sigma}'_k[t] \ge \boldsymbol{\sigma}_k[t]$ and $\boldsymbol{\sigma}'_k[t^q] \ge \boldsymbol{\sigma}_k[t^q]$ for every $t^q \in {}^\bullet({}^\bullet t)$. Given that the net is choice-free, for every place p it holds that $|{}^\bullet p| = 0$ or $|{}^\bullet p| \ge |p^\bullet| = 1$. If for $p \in {}^\bullet t$, it holds that $|{}^\bullet p| \ge |p^\bullet| = 1$, then the previous inequalities ensure $\boldsymbol{m}'_k[p] \ge 1$. If p has no input transitions, then it must hold that $\boldsymbol{\sigma}'_{k+1}[t] \le m_0[p]$. Therefore $t_{\gamma_{k+1}}$ can fire from $\boldsymbol{m}'_k$ an amount of 1.

The second lemma states that if a certain sequence σ deadlocks an ordinary choice-free HAPN, then its firing count vector is in the naturals.

Lemma 9.14 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ be an ordinary choice-free HAPN system with $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$. If σ is a fireable sequence $\boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}$, such that $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ deadlocks at $\boldsymbol{m}$, then $\boldsymbol{\sigma} \in (\mathbb{N} \cup \{0\})^{|T|}$, where $\boldsymbol{\sigma}$ is the firing count vector of σ.*

Let us first prove that if $\boldsymbol{m}$ is a deadlock marking then for every transition t there exists $p \in {}^\bullet t$ such that $m[p] = 0$. Notice that just after the last firing of t in the sequence σ, which is necessarily discrete firing given that $\boldsymbol{\mu} \in \mathbb{N}^{|P|}$, at least one place $p \in {}^\bullet t$ becomes empty. Assume that after such a firing, a transition $t' \in {}^\bullet p$ fires. If the firing of t' is discrete then t would become enabled again; if it is continuous then t' is sufficiently enabled to fire also as discrete what would enable t. Hence, after the last firing of t, no transition $t' \in {}^\bullet p$ can fire and p remains empty.

Assume that $\boldsymbol{\sigma}[t] > 0$ is not a natural number and that $m[p] = 0$ for a given $p \in {}^\bullet t$. Then, there exists $t' \in {}^\bullet p$ such that $\boldsymbol{\sigma}[t']$ is not a natural number and $\boldsymbol{\sigma}[t'] \leq \boldsymbol{\sigma}[t] - m_0[p]$. Notice that there also exists $p' \in {}^\bullet t'$ such that $m[p'] = 0$, hence $t'' \in {}^\bullet p'$ exists such that $\boldsymbol{\sigma}[t'']$ is not a natural number and $\boldsymbol{\sigma}[t''] \leq \boldsymbol{\sigma}[t'] - m_0[p'] \leq \boldsymbol{\sigma}[t] - m_0[p] - m_0[p']$. This reasoning can be repeated until a transition t^* is found such that it deadlocked with $\boldsymbol{\sigma}[t^*] < 1$. Contradiction since natural thresholds do not allow $\boldsymbol{\sigma}[t^*]$ to be less than 1.

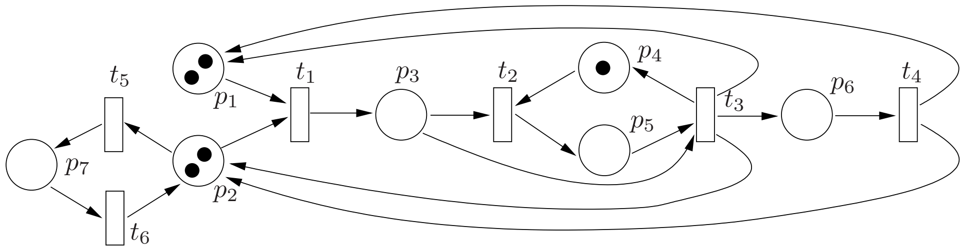


Figure 9.15: Ordinary PN system that is deadlock-free as DPN, but it deadlocks as HAPN with $\boldsymbol{\mu} = \mathbf{1}$.

Therefore, because of Lemmas 9.13 and 9.14, if a deadlock marking $\boldsymbol{m}$ is reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ when σ is fired, the same deadlock marking $\boldsymbol{m}'$ is reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$, when $\lceil \sigma \rceil$ is fired. Thus, if $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is deadlock-free, then $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ is deadlock-free too.

Theorem 9.15 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ be an ordinary choice-free and deadlock-free discrete system. Then, the HAPN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ is deadlock-free for any $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$.*

If non ordinary or non choice-free PN are considered, Theorem 9.15 does not hold. For example, consider the PN in Fig. 9.15. It is an ordinary PN which is not choice-free. It is deadlock-free as DPN. However, considered as HAPN with thresholds $\boldsymbol{\mu} = \mathbf{1}$, it can reach the deadlock marking $\boldsymbol{m}_d = (0.5, 0, 0.5, 0, 1, 0, 0.5)$ by firing the firing sequence $\sigma = 0.5t_5 0.5t_1 1t_1 1t_2$ from the initial marking $\boldsymbol{m}_0 = (2, 2, 0, 1, 0, 0, 0)$ shown in the figure. Hence, deadlock-freeness property is not preserved by the HAPN with $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$.

The following corollary is straightforwardly obtained from Theorems 9.11 and 9.15.

Corollary 9.16 *Let $\mathcal{N}$ be an ordinary choice-free net. $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_D$ is deadlock-free iff $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ is deadlock-free with $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$.*

Preservation of deadlock-freeness in a S^{3}PR example

Let us consider the Example 9.17, in which the PN in Fig. 9.16 (a) is considered, which is a System of Simple Sequential Processes (S^{3}PR) [ECM95].

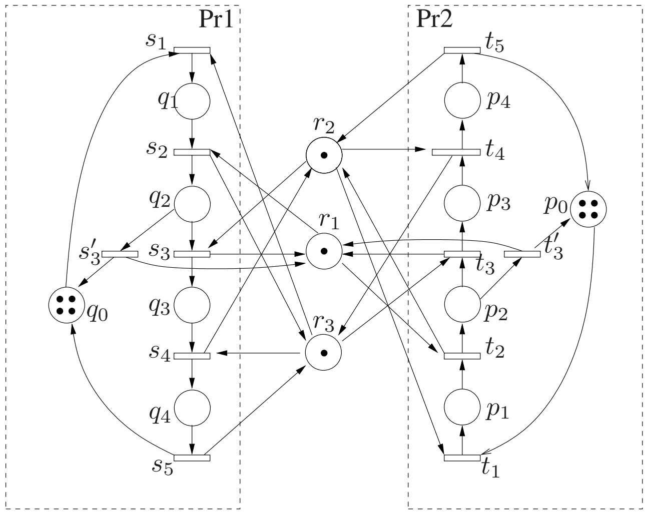


Figure 9.16: A S[3]PR system [GV99] that is deadlock-free as discrete, but it deadlocks as continuous ($\boldsymbol{m}_d = (3, 0, 0, 1, 0, 3, 0, 0, 1, 0, 0, 0, 1)$). It is deadlock-free as HAPN with $\boldsymbol{\mu} = \mathbf{1}$.

Example 9.17 *The PN in Fig. 9.16 models a system in which two processes Pr_1 and Pr_2 share resources r_1, r_2 and r_3. The PN system is live as discrete. However, when the PN system is fluidified, i.e., transitions can be fired in non negative real amounts, the system can reach a deadlock: from the initial marking $\boldsymbol{m}_0$, where $m_0[r_1] = m_0[r_2] = m_0[r_3] = 1$, $m_0[q_0] = m_0[p_0] = 4$ and the other places are empty, the firing sequence σ = $\frac{1}{2}s_1\frac{1}{2}t_1\frac{1}{2}s_2\frac{1}{2}t_2\frac{1}{2}s_3\frac{1}{2}t_3\frac{1}{2}s_1\frac{1}{2}t_1\frac{1}{2}s_2\frac{1}{2}t_2\frac{1}{2}s_3\frac{1}{2}t_3$ can be fired, reaching the deadlock marking $\boldsymbol{m}_d$, where $m_d[r_1] = m_d[q_3] = m_d[p_3] = 1$, $m_d[q_0] = m_d[p_0] = 3$ and the other places are empty. The continuous PN system does not preserve the deadlock-freeness of the discrete PN system. However, this property will be preserved by the HAPN system with appropriate thresholds (in this case, with $\boldsymbol{\mu} = \mathbf{1}$).*

An important issue in HAPN is to choose adequate thresholds $\boldsymbol{\lambda}$ in a given HAPN. An interesting criterion is to identify which transitions are involved in a *bad behaviour* that might happen in the CPN but not in the DPN. Then, the thresholds of the transitions related to such bad behaviour should be higher than 0, to preserve some discrete behaviour, while the rest of the transitions can be continuous (i.e., threshold equal to 0).

A *bad behaviour* can be the existence of a spurious deadlock which becomes reachable by the emptying of a trap (see [Sil+11]), such as in the PN considered in the Example 9.17. In that case, the undesired deadlocks occur in the limit, when the firings of the transitions can be as small as desired. This can be avoided by setting any positive threshold for the transitions, as it is illustrated in the example.

Appropriate thresholds have been selected for the examples considered in this work. However, a general method to obtain the thresholds for any net system requires more investigation.

9.5 Conclusions

The formalism of HAPN, introduced for timed PN in [YLL09], has been generalized to the untimed context in this chapter. In contrast with HPN, in which the partition among *continuous* and *discrete* is *static*, HAPN adapts its behaviour *dinamically* with the load of the syhstem. An *adaptive* transition has two different modes: *continuous* and *discrete*, continuous mode for high transition load (in this case, enabling degree higher than a given threshold) and discrete in other case.

HAPN can be seen as a conceptual framework which includes discrete, continuous and hybrid PN. Moreover, by selecting the appropriate thresholds, HAPN offer the chance to preserve important properties of discrete event systems, as deadlock-freeness,that are not always retained by fully continuous approximations.

Reachability and deadlock-freeness property of hybrid adaptive nets has been compared with those of DPN and CPN. For a rather general class of nets, an inclusion relationship was proved for the reachability sets of the discrete, hybrid adaptive and continuous nets. With respect to deadlock-freeness, although this property is not preserved in general for arbitrary real thresholds, it was shown that it is necessary and sufficient for deadlock-freeness of choice-free nets with arbitrary natural thresholds.

10 Reachability analysis of hybrid adaptive Petri nets

"*All failure is failure to adapt, all success is successful adaptation*".
Max McKeown

The aim of this chapter is the characterization of the set of reachable markings of an untimed *Hybrid Adaptive Petri Net* (HAPN) system. An algorithm is proposed to achieve it, which computes not only the set of reachable markings, but also its reachability graph. The algorithm is illustrated with the computation of the reachability graph and reachability set of some examples.

Due to the fact that HAPN is a general formalism which includes discrete, continuous and hybrid PN, the techniques developed here can be also used for them. In this chapter, the algorithm to calculate the reachability graph of HAPN is simplified to be applied to hybrid PN, as a particular case of the algorithm. It is also applied to some examples, and it is used to verify some system properties.

Introduction

This chapter studies in depth the reachability of HAPN, which is a basic topic of major importance, highly related to properties as deadlock-freeness, liveness, reversibility, etc. Based in some basic reachability concepts related to DPN and CPN, the set of reachable markings of a HAPN system are characterized. An algorithm to compute the *Reachability Graph* (RG) of HAPN is proposed in this chapter. The *Reachability Set* (RS) of a HAPN can be directly obtained by the computed RG. Moreover, due to the fact that HAPN contains other formalisms such as HPN, the algorithm can be simplified to compute the RG of HPN. The algorithm to compute the RG is simplified for HPN, obtaining a particular characterization of the RG of HPN, in which the firing of discrete transitions are explicitly shown in the arcs of the graph, while the firing of continuous transitions are implicitly included in the nodes. The obtained RG is compared with other methods from the literature.

The rest of the chapter is organized as follows. In Section 10.1, some concepts related to reachability of DPN and CPN such as RS and FS are recalled, those concepts are defined for HAPN, and the formal definition of the RG of HAPN is given. The algorithm to compute the RG of HAPN is proposed in Section 10.2. The definition of the RG and the algorithm to compute it are simplified for HPN in Section 10.3. The conclusions are summarized in Section 10.4

10.1 Previous concepts. Reachability in hybrid adaptive Petri nets

The Reachability Set (RS) of a PN system is the union of all the markings which are reachable by the system from the initial one.

The RS of a DPN system are *disjoint* points in the $\mathbb{N}^{|P|}$ space. By contrast, the RS of a CPN system is a convex set in $\mathbb{R}_{\geq 0}^{|P|}$ [RTS99]. Because HAPN combines discrete and continuous firings of transitions, its RS is a union of certain *disjoint* sets.

These sets are defined in Chapters 2 and 4 for discrete and continuous PN. Let us recall them and define it for HAPN (here, t_{γ_i} denotes the i-th transition of the sequence σ):

Definition 10.1

- $RS_D(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} \mid \exists\sigma,\ \sigma = t_{\gamma_1} \dots t_{\gamma_k},\ \textit{s.t.}\ \boldsymbol{m}_{i-1} \xrightarrow{t_{\gamma_i}} \boldsymbol{m}_i\ \forall i \in \{1..k\}\ \textit{and}\ \boldsymbol{m}_k = \boldsymbol{m}\}$
- $RS_C(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} \mid \exists\sigma,\ \sigma = \alpha_1 t_{\gamma_1} \dots \alpha_k t_{\gamma_k},\ \textit{s.t.}\ \boldsymbol{m}_{i-1} \xrightarrow{\alpha_i t_{\gamma_i}} \boldsymbol{m}_i,\ \alpha_i \in \mathbb{R}_{>0}\ \forall i \in \{1..k\},\ \textit{and}\ \boldsymbol{m}_k = \boldsymbol{m}\ \}$
- $RS_A(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{m} \mid \exists\ \sigma,\ \sigma = \alpha_1 t_{\gamma_1} \dots \alpha_k t_{\gamma_k},\ \textit{s.t.}\ \boldsymbol{m}_{i-1} \xrightarrow{\alpha_i t_{\gamma_i}} \boldsymbol{m}_i,\ \alpha_i \in \mathbb{R}_{>0}\ \textit{if}\ mode(t_{\gamma_i}, \boldsymbol{m}_{i-1}) = C\textit{; and}\ \alpha_i = 1\ \textit{if}\ mode(t_{\gamma_i}, \boldsymbol{m}_{i-1}) = D,\ \forall i \in \{1..k\},\ \textit{and}\ \boldsymbol{m}_k = \boldsymbol{m}\}$

An interesting concept which is used in the characterization of the set of markings due to continuous firings is denoted Fireable Set (FS), and it is a set composed of the sets of transitions for which a continuous firing sequence exists. As explained in Chapter 6, the FS is defined for a CPN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$ [JRS03]. It can be defined and computed similarly for a HAPN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$, $\text{FS}_A(\mathcal{N}, \boldsymbol{m}_0)$ is the set of sets of transitions which are enabled as continuous in $\boldsymbol{m}_0$, and those which are exactly at its threshold at $\boldsymbol{m}_0$ but can become continuous by the firing of other transition in continuous mode. $\text{FS}_C(\mathcal{N}, \boldsymbol{m}_0) \subseteq 2^T$ and $\text{FS}_A(\mathcal{N}, \boldsymbol{m}_0) \subseteq 2^T$, where 2^T denote the set of all possible subsets of T, are defined as follows:

Definition 10.2

- $FS_C(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{\theta} \mid \exists\, \sigma$ *fireable from* $\boldsymbol{m}_0$, *such that* $\boldsymbol{\theta} = ||\boldsymbol{\sigma}||\ \}$
- $FS_A(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{\theta} \mid \exists \sigma = \alpha_1 t_{\gamma_1} \ldots\ \alpha_k t_{\gamma_k}$ *s.t.* $\boldsymbol{m}_{i-1} \xrightarrow{\alpha_i t_{\gamma_i}} \boldsymbol{m}_i$, $\alpha_i \in \mathbb{R}_{>0}$, $\forall i \in \{1..k\}$ $mode(t_{\gamma_i}, \boldsymbol{m}_{i-1}) = C$ *and* $enab(t_{\gamma_i}, \boldsymbol{m}_0) \geq \mu_{\gamma_i}$, *and* $\boldsymbol{\theta} = ||\boldsymbol{\sigma}||\}$

Intuitively, $\text{FS}_A(\mathcal{N}, \boldsymbol{m}_0)$ can be computed by firing transitions which are enabled as continuous in $\boldsymbol{m}_0$, or that are in the threshold and will be enabled as continuous by the continuous firing of other transitions. It can be computed with Algorithm 6.

Algorithm 3: FireableSet$_A$($\mathcal{N}$,$\boldsymbol{m}_0$):FS_A

Input: PN ($\mathcal{N}$), initial marking ($\boldsymbol{m}_0$)
Output: Fireable Set (FS_A)

1 $V = \{\ t_j \mid t_j \in T, enab(t_j, \boldsymbol{m}_0) > \mu_j\ \}$ % *transitions enabled as continuous at* $\boldsymbol{m}_0$
2 $FS_A = \{v | v \subseteq V\}$ % *all the subsets of V* **while** *not every element of* FS_A *has been taken* **do**
3   Take $f \in FS_A$ that has not been taken yet
4     $V = \{t_j\ | enab(t_j, \boldsymbol{m}_0) = \mu_j \wedge (\forall p \in {}^\bullet t_j,\ \frac{m_0[p]}{Pre[p,t_j]} > \mu_j \vee {}^\bullet p \cap f \neq \emptyset)\ \}$
5     $FS_A = FS_A \cup \{f \cup v | v \subseteq V\}$
6 **end**

To illustrate how the set $\text{FS}_A(\mathcal{N}, \boldsymbol{m}_0)$ is computed, consider the HAPN system in Fig. 10.8, with threshold vector $\boldsymbol{\mu} = (1, 1, 1, 1)$ and initial marking $\boldsymbol{m}_0 = (2, 0, 1, 2)$. Its FS_A is, after step 2, $\text{FS}_A(\mathcal{N}, \boldsymbol{m}_0) = \{\emptyset, \{t_4\}\}$, because t_4 is the only transition enabled as continuous. At line 5 of Algorithm 6, with f = $\{t_4\}$ the set V results $\{t_1, t_3\}$, thus three new sets can be added to the set: $\text{FS}_A(\mathcal{N}, \boldsymbol{m}_0) = \{\emptyset, \{t_4\}, \{t_1, t_4\}, \{t_3, t_4\}, \{t_1, t_3, t_4\}\}$. No more transition is enabled, and the algorithm stops.

A general characterization of the set of reachable markings in CPN is presented in Chapter 6:

Theorem 10.3 *Given* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$, $\boldsymbol{m} \in RS_C(\mathcal{N}, \boldsymbol{m}_0)$ *iff there exists a vector* $\boldsymbol{\sigma}$ *s.t.*

- $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma},\ \boldsymbol{m} \geq \mathbf{0},\ \boldsymbol{\sigma} \geq \mathbf{0}$
- $||\boldsymbol{\sigma}|| \in FS_C(\mathcal{N}, \boldsymbol{m}_0) \cap FS_C(\mathcal{N}^{-1}, \boldsymbol{m})$

Let us now define a *homogeneous region* as a set of markings for which the enabling degree of each transition is either over (or equal to) the threshold for every marking, or below (or equal to) the threshold for every marking.

Definition 10.4 *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$ be a HAPN system. A set $R \subseteq \mathbb{R}_{\geq 0}^{|P|}$ is a* homogeneous region *of markings if:*

- $R \subseteq RS_A(\mathcal{N}, \boldsymbol{m}_0)$.
- *For each $t_j \in T$, $(\forall \boldsymbol{m} \in R,\ enab(t_j, \boldsymbol{m}) \leq \mu_j) \vee (\forall \boldsymbol{m} \in R,\ enab(t_j, \boldsymbol{m}) \geq \mu_j)$.*

Finally, the Reachability Graph of HAPN is defined. It consists of nodes, that correspond with homogeneous regions obtained by the continuous firings of transitions, and arcs connecting the nodes. In this case, the arcs are of two different types, corresponding either with the fact that a transition mode has changed (ε-arcs) or with the discrete firing of a transition (D-arcs). Given a HAPN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_A$, we define $\text{RG}_A(\mathcal{N}, \boldsymbol{m}_0)$ as a labelled directed graph with two types of arcs, whose nodes are homogeneous regions defined over $RS_A(\mathcal{N}, \boldsymbol{m}_0)$:

Definition 10.5 *$RG_A(\mathcal{N}, \boldsymbol{m}_0) = \langle setofNodes, setofEarcs, setofDarcs \rangle$, where:*

- *Each $R_i \in setofNodes$ is a homogeneous region, $R_i \subseteq RS_A(\mathcal{N}, \boldsymbol{m}_0)$.*
  It holds that $\cup_{i=1}^{|setofNodes|} R_i = RS_A(\mathcal{N}, \boldsymbol{m}_0)$.
- *Each $A_k^{\varepsilon} \in setofEarcs$ is defined as a tuple $A_k^{\varepsilon} = \langle R_i, R_j, t_z \rangle$:*
  - *$R_i, R_j \in setofNodes$ are the source and target nodes of the arc.*
  - *$t_z \in T$ indicates the transition whose mode changes when the marking moves from R_i to R_j.*
  - *It holds that $\forall \boldsymbol{m}_j \in R_j, \exists \boldsymbol{m}_i \in R_i$ s.t. $enab(t_z, \boldsymbol{m}_i) = \mu_z$, a fireable sequence σ exists s.t. $\boldsymbol{m}_i \xrightarrow{\sigma} \boldsymbol{m}_j$, and σ contains only continuous firings.*
- *Each $A_k^{D} \in setofDarcs$ is defined as a tuple $A_k^{D} = \langle R_i, R_j, guard, t_z \rangle$:*
  - *$R_i, R_j \in setofNodes$ are the source and target nodes of the arc*
  - *$guard \subseteq \mathbb{R}_{\geq 0}^{|P|}$ gives a set of conditions over R_i.*
  - *$t_z \in T$ is the transition which is fired as discrete to move from R_i to R_j.*
  - *It holds that $\forall \boldsymbol{m}_j \in R_j, \exists \boldsymbol{m}_i \in R_i \cap guard$, a fireable sequence σ exists s.t. $\boldsymbol{m}_i \xrightarrow{1t_z \cdot \sigma} \boldsymbol{m}_j$, and σ contains only continuous firings.*

$R_0 \in setofNodes$ *is the initial node of the graph, where* $\boldsymbol{m}_0 \in R_0$.

The set of markings R_i can be described with linear inequalities over $\mathbb{R}^{|P|}$, as it can be seen along the examples.

In contrast with the RG of discrete PN systems, in which every firing of a transition is explicitly denoted, the RG of HAPN compacts the firings of the continuous transitions inside the nodes. It can be identified as two levels of description, in which nodes are not states, but sets of markings inside which transitions can be fired. Indeed, two systems with different behaviours may have the same underlying graph. In general, some behaviour can be directly seen in the RG, but some evolution of the marking is done inside the node and it is not explicitly seen in the RG.

10.2 On the computation of a Reachability Graph for hybrid adaptive Petri nets

An algorithm to compute the RG of a HAPN is proposed in this section. The RG will be computed recursively, each call to the algorithm will return a node which contains a set of markings, together with its corresponding arcs in the RG.

Algorithm 6 initializes some global variables ($setofNodes$, $setofEarcs$, $setofDarcs$), and calls the recursive function $explore_A$ (Algorithm 20) with the initial marking of the system. As in the case of HPN, $explore_A$ is a recursive function that, given a marking or a set of markings, calculates the markings reachable from it due to continuous firings, and calculates its adjacent nodes (reached with ε-arcs or D-arcs). Function $explore_A$ will be recursively called until the full RG has been calculated.

Function $explore_A$ takes a homogeneous region R as input parameter. After checking if it has been considered before or the stopping condition holds (line 1), the markings which are reachable from R due to continuous firings (i.e., considering only firings of the transitions of the FS) are computed, with the condition that no transition traverses its threshold. This set of markings can be mathematically characterized with (10.1), obtained from Theorem 10.3 where $\text{FS}_A(\mathcal{N}, \boldsymbol{m}_0)$ is calculated as specified in the previous subsection. This function is called in line 2 in Algorithm 20:

$$continuousM_A(\mathcal{N}, R) = \{\boldsymbol{m}| \quad \begin{aligned}[t] & \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma},\ \boldsymbol{m}_0 \in R, \\ & \boldsymbol{m} \geq \boldsymbol{0},\ \boldsymbol{\sigma} \geq \boldsymbol{0}, \\ & ||\boldsymbol{\sigma}|| \in \text{FS}_A(\mathcal{N}, \boldsymbol{m}_0) \cap \text{FS}_A(\mathcal{N}^{-1}, \boldsymbol{m}), \\ & \forall t_i \in T,\ enab(t_i, \boldsymbol{m}) \geq \mu_i \Leftrightarrow enab(t_i, R) \geq \mu_i\} \end{aligned} \tag{10.1}$$

where $enab(t_i, R)$ is the enabling degree of t_i at an arbitrary marking $\boldsymbol{m}'$ in the homogeneous region R.

Then, the ε-arcs, those due to mode changes, are calculated (lines 4-8). This is done in three steps: First, the set of markings in which transition t_j changes its *mode* is characterized.

This set is characterized with (10.2), which just adds an additional constraint to (10.1), and which is used in line 5:

$$\begin{aligned} frontierM_A(\mathcal{N}, R, t_j) = \quad \{\boldsymbol{m}| \quad & \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C}\cdot\boldsymbol{\sigma},\ \boldsymbol{m}_0 \in R, \\ & \boldsymbol{m} \geq \boldsymbol{0},\ \boldsymbol{\sigma} \geq \boldsymbol{0}, \\ & ||\boldsymbol{\sigma}|| \in \mathrm{FS}_A(\mathcal{N}, \boldsymbol{m}_0) \cap \mathrm{FS}_A(\mathcal{N}^{-1}, \boldsymbol{m}), \\ & enab(t_j, \boldsymbol{m}) = \mu_j, \\ & \forall t_i, enab(t_i, \boldsymbol{m}) \geq \mu_i \Leftrightarrow enab(t_i, R) \geq \mu_i\} \end{aligned} \tag{10.2}$$

Second, once the $frontier$ markings f are calculated, the recursive algorithm $explore_A$ is called with f. Third, an ε-arc is created from the current node to the node obtained by $explore_A$ (the second and the third steps are done in line 7). This ε-arc is labeled with the transition t_j which changes its mode, which indicates that this ε-arc can only be taken from the markings of R which hold $enab(t_j) = \mu_j$.

Finally, the D-arcs from node R are created (lines 10-18)), considering the possible discrete firing of each transitions from R. A transition t_j can only be fired as discrete if $1 \leq enab(t_j, \boldsymbol{m}_0) \leq \mu_j$. In principle, the set of markings reached by the discrete firing of t_j an amount equal to 1 is:

$$dFM = \{\boldsymbol{m}|\boldsymbol{m} = \boldsymbol{m}_0 + C[P, t_j], \boldsymbol{m}_0 \in R, 1 \leq enab(t_j, \boldsymbol{m}_0) \leq \mu_j\}$$

The preliminary idea would be to consider dFM as the target node of the D-arc. Unfortunately, dFM might not be a homogeneous region, in fact the discrete firing of t_j can modify the mode of the transitions whose input places are in ${}^\bullet t_j$ (because its marking is decreased) or $t_j{}^\bullet$ (because its marking is increased). In the algorithm, this transitions are denoted as neighbor transitions: $neiT = ({}^\bullet t_j)^\bullet \cup (t_j{}^\bullet)^\bullet$ (line 11).

The nodes of the RG have to be *homogeneous* regions. Hence, the algorithm has to *partitionate* dFM before adding new nodes to the RG. Notice that in each target homogeneous region R_j, for every $t_n \in neiT$ it must hold $\forall \boldsymbol{m}, enab(t_n, \boldsymbol{m}) \geq \mu_n$ or $\forall \boldsymbol{m}, enab(t_n, \boldsymbol{m}) \leq \mu_n$.

In order to take into account every possible combination of transitions t_n *over* and *below* the threshold, we define T_{cn} as the subset of transitions whose enabling degree is forced to be higher or equal to the threshold ("$\geq$") in a given homogeneous region, $T_{cn} = \{t_{cn} \in neiT | enab(t_{cn}, R) \geq \mu_{cn}\}$. The rest of transitions, $t_{dn} \in neiT \setminus T_n$, are forced to have its enabling degree lower or equal to μ_{dn}.

Then, to consider all the possibilities, the algorithm iterates over $T_{cn} \in 2^{neiT}$. Given $neiT$ and a subset T_{cn}, the set of markings d which are reached from R with the discrete firing of t_j is obtained with (10.3), see line 13:

$$\begin{aligned} discreteM_A(\mathcal{N}, R, t_j, neiT, T_{cn}) = \quad \{\boldsymbol{m}| \quad & \boldsymbol{m} = \boldsymbol{m}_0 + C[P, t_j], \\ & \boldsymbol{m}_0 \in R, \boldsymbol{m} \geq \boldsymbol{0}, \\ & 1 \leq enab(t_j, \boldsymbol{m}_0) \leq \mu_j, \\ & \forall t_{cn} \in T_{cn}, enab(t_{cn}, \boldsymbol{m}) \geq \mu_{cn}, \\ & \forall t_{dn} \in neiT \setminus T_{cn}, enab(t_{dn}, \boldsymbol{m}) \leq \mu_{dn}\} \end{aligned} \tag{10.3}$$

Once d is obtained, the arc from R to d is computed (line 15), where the guard $[\boldsymbol{m}_0|\boldsymbol{m}_0 = \boldsymbol{m} - C[P, t], \boldsymbol{m} \in d]$ is specified. If the set of markings dFM has been divided in several *homogeneous regions* dealing to different nodes, then there will be several D-arcs from R with different guards and with different target nodes (see Example 10.6).

Algorithm 4: calculateRG$_A(\mathcal{N}, \boldsymbol{m}_0)$: $reachGraph$

Input: PN ($\mathcal{N}$), initial marking ($\boldsymbol{m}_0 \in \mathbb{R}^{|P|}_{\geq 0}$)
Output: Reachability graph ($reachGraph$)

```
setofNodes = ∅
setofDarcs = ∅
setofEarcs = ∅
R0 = exploreA(N, {m0})
RGA = ⟨setofNodes, setofDarcs, setofEarcs⟩
return RGA
```

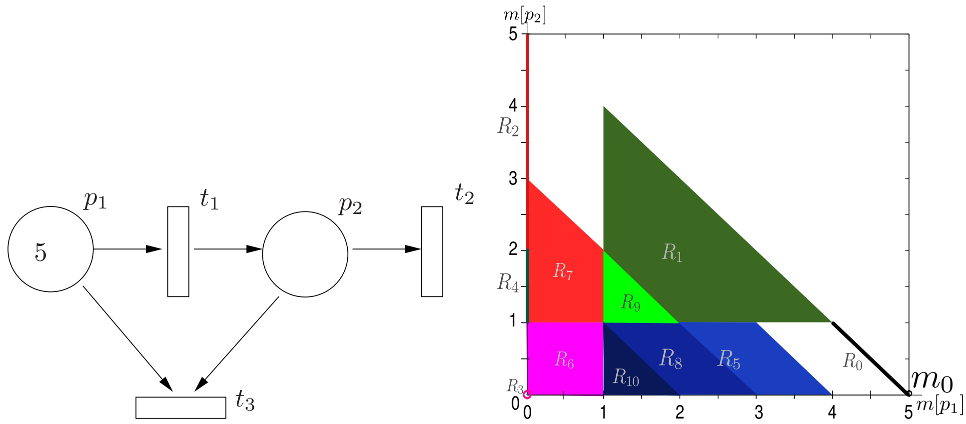
Example 10.6 *Let us illustrate the algorithm with the PN in Fig. 10.1(a), with thresholds* $\boldsymbol{\mu} = (1, 1, 5)$ *and initial marking* $\boldsymbol{m}_0 = (5, 0)$*. Notice that with this* $\boldsymbol{m}_0$*, the marking of* p_1 *will be never higher than 5, thus for any reachable marking* $\boldsymbol{m}$*,* $enab(t_3, \boldsymbol{m}) = min\{m[p_1], m[p_2]\} \leq 5 = \mu_3$*, and hence* t_3 *will be always in discrete mode.*

At the initial marking $\boldsymbol{m}_0 = (5, 0)$*, only transition* t_1 *is enabled and it is in continuous mode (*$m[p_1] > \mu_1$*). When* t_1 *is fired as continuous (and* t_2, t_3 *remain discrete and not enabled) all the markings contained in the segment from* $(5, 0)$ *to* $(4, 1)$ *can be reached. This segment is the starting set of the RG (see node* R_0 *in Fig. 10.2).*

Algorithm 5: explore$_A(\mathcal{N}, R) : n)$

Input: PN ($\mathcal{N}$), markingSet ($R \subseteq \mathbb{R}^{|P|}_{\geq 0}$)
Output: set of markings (n)

if $R \notin setofNodes \wedge$ *not* $stoppingCond(R, setofNodes)$ **then**
    $n = continuousM_A(\mathcal{N}, R)$
    $setofNodes = setofNodes \cup n$
    for $t_j \in T$ **do**
        $f = frontierM_A(\mathcal{N}, R, t_j)$
        if $f \neq \emptyset$ **then**
            $setofEarcs = setofEarcs \cup \langle n, explore_A(\mathcal{N}, f), t_j \rangle$
        end
    end
    for $t_j \in T$ **do**
        $neiT = ({}^{\bullet}t_j)^{\bullet} \cup (t_j{}^{\bullet})^{\bullet}$
        for *each* $T_{cn} \in 2^{neiT}$ **do**
            $d = discreteM_A(\mathcal{N}, n, t_j, neiT, T_{cn})$
            if $d \neq \emptyset$ **then**
                $setofDarcs = setofDarcs \cup$
                $\langle n, explore_A(\mathcal{N}, d), [\boldsymbol{m}_0 | \boldsymbol{m}_0 = \boldsymbol{m} - C[P, t_j], \boldsymbol{m} \in d], t_j \rangle$
            end
        end
    end
end
return n



(a) (b)

Figure 10.1: (a) Example of a HAPN system. (b) Its Reachability Set, with $\boldsymbol{\mu} = (1, 1, 5)$ and $\boldsymbol{m}_0 = (5, 0)$. It is represented in the axes $m[p_1]$ and $m[p_2]$, because $m[p_3]$ is linearly dependent, more precisely $m[p_3] = 10 - 2\, m[p_1] - m[p_2]$.

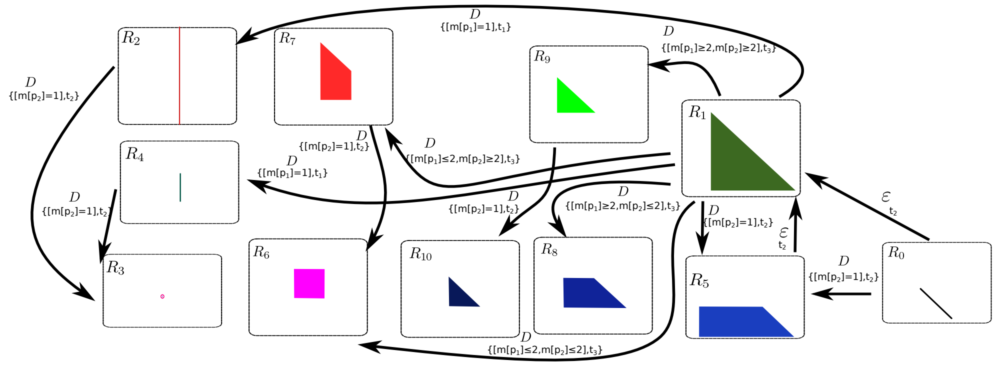


Figure 10.2: RG of the HAPN system in Figure 10.1(a), with $\boldsymbol{\mu} = (1, 1, 5)$ and $\boldsymbol{m}_0 = (5, 0)$. The homogeneous regions of the nodes correspond to the ones depicted in Fig. 10.1(b).

Let us show how ε-arcs are created (lines 4-8). When p_2 (see R_1 in Fig. 10.2) reaches a marking higher to 1, the enabling degree of t_2 is higher than its threshold, and t_2 becomes continuous ($enab(t_2, m) > \mu_2$). This mode change is represented in the RG as an ε-arc, in which no transition firing is considered. The transition which changes its node, t_2, is indicated in the ε-arc. This arc can only be taken from $R_1 \cap [enab(t_2) = \mu_2]$, which corresponds to the mode change.

In the example, $frontierM_A(\mathcal{N}, R_0, t_2)$ calculates the set $\{(4, 1)\}$, that is the marking *from R_0 at which the mode of t_2 changes. Given this set, an ε-arc is created (see Fig. 10.2) labelled with t_2, from R_0 to R_1. The node R_1 (see Fig. 10.1(b) and 10.2) is created from $\boldsymbol{m} = (4, 1)$ in the same way as before: considering the continuous markings of t_1, t_2 from (4,1), i.e., by computing $continuousM_A(\mathcal{N}, \{(4, 1)\})$.*

In order to illustrate the creation of the D-arcs, let us consider node R_1, in which transition t_3 is enabled as discrete for every marking $\boldsymbol{m} \in R_1$, because $1 \leq enab(t_3, \boldsymbol{m}) < 5 = \boldsymbol{m}_3$. The firing of t_3 from R_1 would reach the triangle obtained by the union of $R_6 \cup R_7 \cup R_8 \cup R_9$. However, it would not be a homogeneous region, for example $\forall \boldsymbol{m} \in R_6, enab(t_1, \boldsymbol{m}) \leq \mu_1$, while $\forall \boldsymbol{m} \in R_8, enab(t_1, \boldsymbol{m}) \geq \mu_1$. In this case, $neiT$ = $\{t_1, t_2, t_3\}$, but t_3 will never be in continuous mode, as explained before. Four nodes R_6, R_7, R_8, R_9 are obtained, which are reached from R_1 through D-arcs (lines 12-17 in Algorithm 20). For example, when $T_{cn} = \emptyset$, node R_6 is obtained, which is reached with the D-arc $\langle R_1, R_6, [m[p_1] \leq 2, m[p_2] \leq 2], t_3 \rangle$. It means that when transition t_3 is fired from $R_1 \cap [m[p_1] \leq 2, m[p_2] \leq 2]$, the homogeneous region R_6 is reached, at which the modes of all the transitions are discrete (forced by $T_{cn} = \emptyset$). Node R_7 (resp. R_8 and R_9) is obtained when $T_{cn} = \{t_2\}$ (resp. $\{t_1\}$ and $\{t_1, t_2\}$).

The RG of an unbounded DPN has an infinite number of nodes, while it is finite for bounded DPN. An interesting question is to consider if it would be also true with the number of nodes in the RG of HAPN. Example 10.7 shows a HAPN which is bounded (i.e., $\forall p \in P$, $\exists b \in \mathbb{R}$ s.t. $\forall \boldsymbol{m} \in \mathrm{RG}_H(\mathcal{N}, \boldsymbol{m}_0), m[p] \leq b$) and whose RG has an infinite number of nodes. In the cases in which the RG has an infinite number of steps, the algorithm has to detect this situation and stop the computation of those nodes. This stopping condition is motivated with Example 10.7 and discussed after the example.

Example 10.7 ***A bounded HAPN with infinite reachability graph***

Consider the HAPN in Fig. 10.3, with $\mu_1 = \mu_2 = \infty$ (i.e., t_1 and t_2 are discrete) and $\mu_3 = \mu_4 = 0$ (i.e., t_3 and t_4 are continuous). Actually it is a HPN.

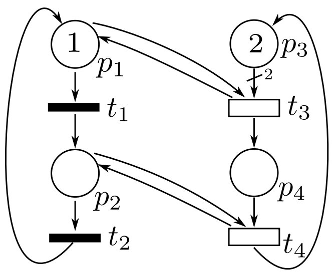


Figure 10.3: HAPN whose RG has an infinite number of nodes.

Let us consider the computation of its RG from $\boldsymbol{m}_0 = (1, 0, 2, 0)$. At $\boldsymbol{m}_0$, transitions t_1 and t_3 are enabled. The first node, R_0, is obtained considering every possible firing of continuous transitions (in this case, any firing of t_3) from $\boldsymbol{m}_0$. It corresponds to lines 2-3 in

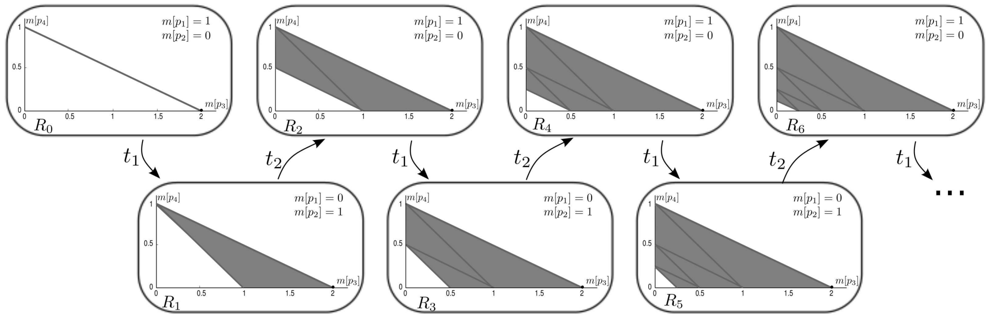


Figure 10.4: RG of the PN in Figure 10.3. It has an infinite number of nodes.

Algorithm 11, and it results the segment between $(1,0,2,0)$ *and* $(1,0,0,1)$*, see node* R_0 *in Fig.10.4.*

From R_0*, the firings of discrete transitions are considered to calculate its adjacent nodes. Consider the firing of* t_1*, which is the only enabled discrete transition. First, the markings obtained by the firing of* t_1 *from the markings in* R_0 *is calculated,* d *in line 5 in Algorithm 11. If* t_1 *is fired,* $m[p_3]$ *and* $m[p_4]$ *do not change,* $m[p_1] = 0$ *and* $m[p_2] = 1$*, hence* d *is a segment between* $(0,1,2,0)$ *and* $(0,1,0,1)$*. Once calculated* d*, it is recursively* $explored$*, and the arc from* R_0 *to* R_1 *is obtained (line 7 in Algorithm 11). In the recursive call to* $explore_H(\mathcal{N}, \mathrm{d})$*, the process is repeated: first the firings of the enabled continuous transition* t_4 *are considered (and the triangle in* R_1 *is obtained), and then the possible arcs are calculated.*

The marking of p_3 *and* p_4 *(consider* $m[p_3] + m[p_4]$*) is decreasing when* t_3*,* t_4 *are successively fired. Notice that transitions* t_1 *and* t_2 *have to be fired every time that the marking of* p_3*,* p_4 *is divided by 2 (for example, to move from* $(1,0,0,1)$ *to* $(1,0,0,0.5)$*). It means that after firing successively* t_1 *and* t_2 *the marking would approach marking* $(1,0,0,0)$*, but it would require an infinite number of firings of the discrete transitions, which creates an infinite number of nodes in the RG.*

Stopping condition. As discussed in Example 10.7, the RG of a HAPN system may have an infinite number of nodes. This can be due to two cases: (a) the marking of a place tends to infinity (so, the net system is unbounded) or (b) the difference among the new node and one of the previous nodes. Case (a) is not possible because structurally bounded nets are considered. Case (b) can occur when the difference among the new node and one of the previous nodes tends asymptotically to 0, which would generate infinite nodes, even in bounded net systems. The occurrence of this situation is checked by the *stopping condition* when a new region R is going to be explored (line 1 in Algorithm 20 If it holds, R is not explored further to avoid an infinite execution of the procedure. In this situation, R contains the same markings than a previously created node R' except a small difference (an increment, decrement of markings, or both) which tends to 0. Sets R and R' are very similar (hence, $R \cap R' \neq \emptyset$), and the difference between them (which is $(R \cup R') \setminus (R \cap R')$) is as small as desired, measured as its n-dimensional volume. The n-dimensional volume of a set S, denoted $\lambda(S)$, is a generalization of the concept of *area* in $\mathbb{R}^2$ or *volume* in $\mathbb{R}^3$ to a n-dimensional $S \subseteq \mathbb{R}^n$ (here, $n = |P|$). Hence,the condition to be checked is: $\exists R' \in setofNodes$ s.t. $R \cap R' \neq \emptyset$ and $\lambda((R \cup R') \setminus (R \cap R')) < \varepsilon$, where ε is a very small value, e.g., $\varepsilon = 10^{-3}$. This stopping condition, included in Algorithm 20 (line 1), is expressed as follows:

$$stoppingCond(R, setofNodes) = (\exists R' \in setofNodes \text{ s.t. } (R \cap R' \neq \emptyset \wedge \lambda((R \cup R') \setminus (R \cap R')) < \varepsilon) \quad (10.4)$$

Reachability set computation. The RS of a HAPN system, $RS_A(\mathcal{N}, \boldsymbol{m}_0)$, is the union of all the homogeneous regions in $\text{RG}_A(\mathcal{N}, \boldsymbol{m}_0)$ which have been computed with Algorithms 6 and 20. The RS of the HAPN in Fig. 10.1(a) is depicted in Fig. 10.1(b).

Example 10.8 ***A signal transduction network modeled with HAPN***
Fig. 10.5 presents a signal transduction network *that can be appropriately modelled by HAPN. In such a network, three different reactions can occur (Fig. 10.5 (a)). This behaviour is sketched in Fig. 10.5 (b). If the places (transitions) with the same label are merged into one place (transition), the net system in Fig. 10.5 (c) is obtained.*

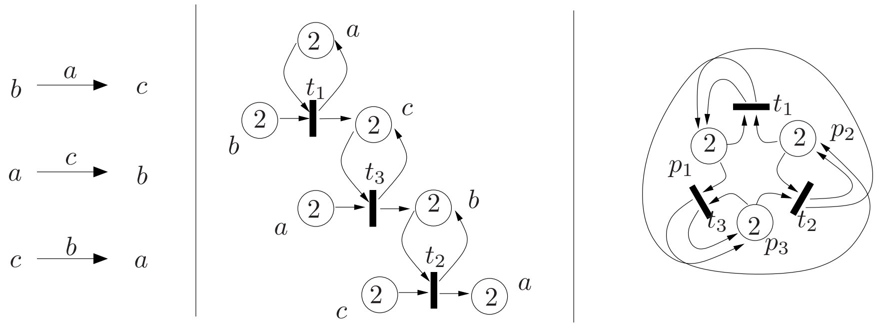


Figure 10.5: (a) Signal transduction network. (b) Sketch of its behaviour. (c) PN modeling the signal transduction network.

This PN system is live as discrete (see its RS in Fig. 10.6(a)). However, when considered as continuous, it reaches in the limit a marking[1] *that is a deadlock (see its RS in Fig. 10.6(b)). Given the initial marking $\boldsymbol{m}_0$ = (2,2,2); the following infinite firing sequence can be fired in the CPN: $\sigma = t_2t_3t_2t_3 1t_1 1t_2 1t_2 1t_3 0.5t_1 0.5t_2 0.5t_2 0.5t_3 0.25t_1 0.25t_2 0.25t_2 0.25t_3 \ldots$ When σ is fired, the deadlock marking $\boldsymbol{m}_d$ = (0, 6, 0) is reached in the limit (see more examples in [RTS99]).*

This undesired deadlock does not occur if instead of a fully continuous model, a HAPN with thresholds $\boldsymbol{\mu} = \mathbf{1}$ is considered. Such HAPN system behaves as continuous while the enabling degree of the places is greater than 1, but it avoids the potential deadlocks, $\boldsymbol{m}_d = (0, 6, 0)$, $\boldsymbol{m}'_d = (6, 0, 0)$ and $\boldsymbol{m}''_d = (0, 0, 6)$, as it can be seen in its RG (and in its RS).

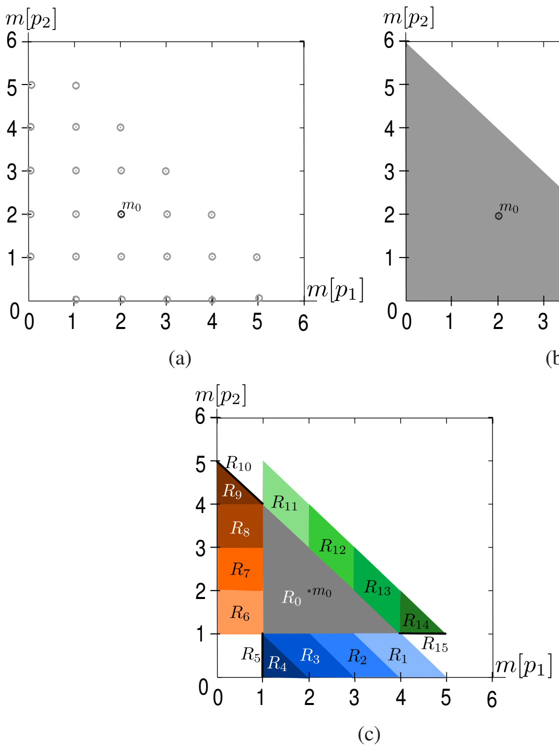


Figure 10.6: RS of the PN system in Fig. 10.5 with $\boldsymbol{m}_0 = (2, 2, 2)$. It is represented in the axes $m[p_1]$ and $m[p_2]$, because $m[p_3]$ is linearly dependent on $m[p_1]$ and $m[p_2]$, more precisely $m[p_3] = 6 - m[p_1] - m[p_2]$. (a) Considered as DPN. (b) Considered as CPN. (c) Considered as HAPN with $\boldsymbol{\mu} = \mathbf{1}$.

The RS of the HAPN system with $\boldsymbol{\mu} = \mathbf{1}$ and $\boldsymbol{m}_0 = (2, 2, 2)$ is shown in Fig.10.6(c). Its RG, computed with Algorithm 6, is depicted in Fig. 10.8. At the initial marking, the three transitions are in continuous mode, and they can fire as continuous within the homogeneous region R_0.

Consider the markings of R_0 at which $m[p_1] = 1$. From those markings, t_3 is enabled as discrete, and it can fire, emptying p_1. Exploring the resulting set of markings, the obtained homogeneous region is R_{18} (the trapezoid composed by the union of R_7, R_8 and R_9 in Fig. 10.6(c)). At R_{18} (indeed, at any region R_6-R_9 and R_{19}), the only transition which is enabled as continuous is t_2. When t_2 is fired, the marking of p_1 increases, and it can reach $m[p_1] = 1$ again. At the markings of R_{18} in which $m[p_1] = 1$, two things can happen: either the marking moves to node R_0 again, through the ε-arc $\langle R_{18}, R_0, t_3 \rangle$ (also with the analogous arc ε-arc

[1]Notice that it is a *spurious* marking: a solution of the state equation which is not reachable by the DPN system.

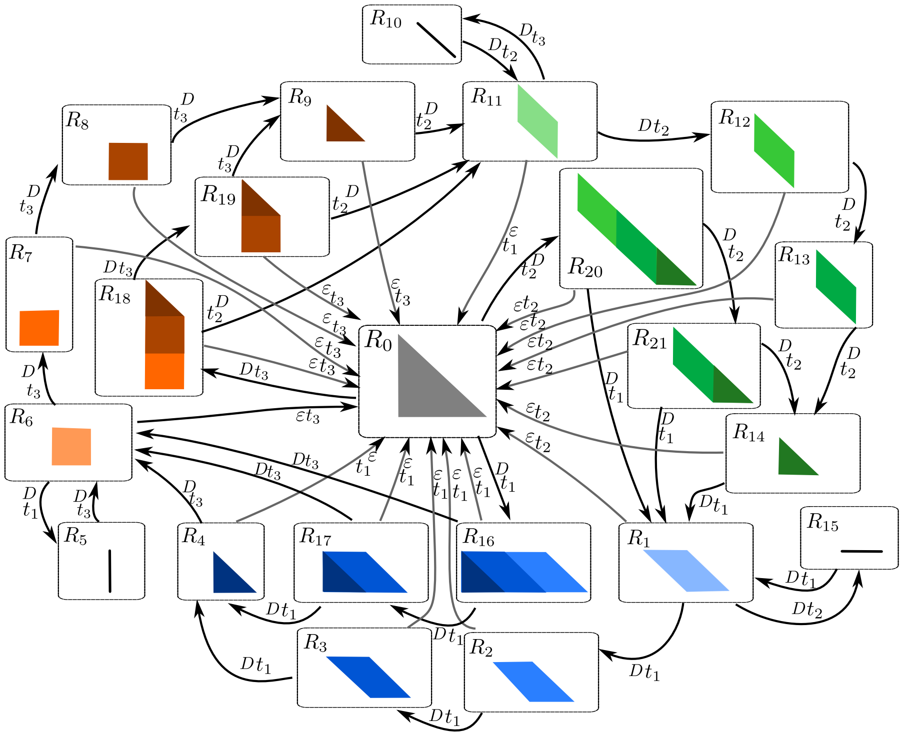


Figure 10.7: RG of the PN system in Fig. 10.5 as HAPN, with thresholds $\boldsymbol{\mu} = \mathbf{1}$.

$\langle R_{18}, R_0, t_1\rangle$, although it is not shown in the figure for simplicity); or transition t_3 is fired as discrete, leading to R_{19} (see the D-arc $\langle R_{18}, R_{19}, [m[p_1] = 1], t_3\rangle$). Notice that the guards in the D-arcs are not shown in the figure for simplicity.

Moreover, in region R_{18} (and also in R_{19}, R_9 and R_{10}), from the markings in which $m[p_3] = 1$ (the segment from (0,5) to (1,4)) transition t_2 is fireable as discrete, as indicated by the D-arc $\langle R_{18}, R_{11}, [m[p_3] = 1], t_2\rangle$. This behaviour is analogous for the three places.

Consider this PN system with a parametrized initial marking $\boldsymbol{m}_0 = (c, c, c)$ where $c \in \mathbb{N}_{>0}$. If considered as a discrete system, the number of reachable markings can be calculated as $\sum_{i=1}^{3c+1} i - 3 = \frac{(3c+1)(3c+2)}{2} - 3 = \frac{9}{2}c^2 + \frac{9}{2}c - 2$, i.e., quadratic with respect to c. If considered as a continuous system, the set of markings that can be reached with a finite or infinite firing sequence is characterized by the single convex $\{\boldsymbol{m} | \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C}\boldsymbol{\sigma} \geq \mathbf{0}\}$. Unfortunately, this set contains deadlock markings that are not reachable by the original discrete system. Let us finally consider the system as HAPN with $\boldsymbol{\mu} = \mathbf{1}$. For $c = 1$, the system behaves as discrete, and the number of reachable markings is 7. For $c > 1$, the system exhibits some continuous behaviour, and the number of homogeneous regions computed by Algorithm 6 is $1 + 3\big((3c-2) + (3c-3)\big) = 18c - 14$, i.e., the number of reachable *regions grows linearly with respect to c. Thus, the HAPN system represents an interesting trade-off between the quality of the approximation (it avoids the spurious deadlocks) and the computational cost of state space exploration.*

10.3 On the computation of a Reachability Graph for hybrid Petri nets

Due to the fact that the HAPN are a formalism that includes hybrid PN, thus discrete and continuous, the methods proposed for the HAPN can be also used for the other formalisms. In this section, the algorithm proposed in Section 10.2 will be simplified and adapted for the computation of the Reachability Graph of a hybrid PN system. First, some reachability concepts on hybrid PN are recalled. Then, the algorithm to compute the RG is presented, it is explained with an example and it is compared with other technique from the literature. Then, it is used to obtain the RG of a HPN which models a production system, and it is used to check some properties.

10.3.1 Basic reachability concepts of hybrid Petri nets

As in HAPN, HPN combines discrete and continuous transitions, its RS is a union of certain *disjoint* sets (due to the combination of discrete and continuous firings).

Definition 10.9 $RS_H(\mathcal{N}, \boldsymbol{m}_0)$ = $\{\boldsymbol{m} \mid \exists\sigma,\ \sigma = \alpha_1 t_{\gamma_1} \ldots \alpha_k t_{\gamma_k}$, *s.t.* $\boldsymbol{m}_{i-1} \xrightarrow{\alpha_i t_{\gamma_i}} \boldsymbol{m}_i$, *where* $\alpha_i \in \mathbb{R}_{>0}$ *if* $\mu_{\gamma_i} = 0$, *while* $\alpha_i = 1$ *if* $\mu_{\gamma_i} = \infty$, $\forall i \in \{1..k\}$, *and* $\boldsymbol{m}_k = \boldsymbol{m}\ \}$

The Fireable Set FS_H of HPN can be defined in a similar way to the FS_A of HAPN.

Definition 10.10 $FS_H(\mathcal{N}, \boldsymbol{m}_0) = \{\boldsymbol{\theta} \mid \exists\ \sigma$ *fireable from* $\boldsymbol{m}_0$, *such that* $\boldsymbol{\theta} = ||\boldsymbol{\sigma}||$, *and* $\forall t_i \in ||\boldsymbol{\sigma}||$, $\mu_i = 0\ \}$

Consider the PN example in Fig. 10.8(a). It is a HPN in which t_1, t_2 are discrete transitions and t_3, t_4 are continuous ($\boldsymbol{\mu} = \{\infty, \infty, 0, 0\}$). Its Fireable Set is $\text{FS}_H(\mathcal{N}, \boldsymbol{m}_0) = \{\emptyset, \{t_4\}, \{t_4, t_3\}\}$.

Finally, a RG for a HPN is defined. It is a directed graph which consists of nodes, and arcs connecting the nodes. The nodes correspond to the sets of markings obtained by the continuous firings of transitions, while the arcs represent the firing of a discrete transition in an amount equal to 1.

Given a HPN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_H$, its $\text{RG}_H(\mathcal{N}, \boldsymbol{m}_0)$ is defined as:

Definition 10.11 $RG_H(\mathcal{N}, \boldsymbol{m}_0) = \langle setofNodes, setofArcs \rangle$ *is a directed graph of nodes* $setofNodes = \{R_1, R_2, \ldots R_i, \ldots\}$ *and directed arcs* $setofArcs = \{A_1, A_2, \ldots A_k, \ldots\}$*:*

- *Each* $R_i \in setofNodes$ *is a set which belongs to the RS, i.e.,* $R_i \subseteq RS_H(\mathcal{N}, \boldsymbol{m}_0)$*. It holds that* $\cup_{i=1}^{|setofNodes|} R_i = RS_H(\mathcal{N}, \boldsymbol{m}_0)$*.*

- *Each* $A_k \in setofArcs$ *is defined as a tuple* $A_k = \langle R_i, R_j, t\rangle$*:*
  - $R_i, R_j \in setofNodes$ *are the source and target nodes of the arc.*
  - $t \in T$ *is the discrete transition which is fired to move from* R_i *to* R_j*.*
  - *It holds that* $\forall \boldsymbol{m}_j \in R_j$, $\exists \boldsymbol{m}_i \in R_i$ *and a continuous sequence* σ *such that* $\boldsymbol{m}_i \xrightarrow{1t\cdot\sigma} \boldsymbol{m}_j$*, and* $\forall t_z \in ||\boldsymbol{\sigma}||$, $\mu_z = 0$.

$R_0 \in setofNodes$ *is the initial node of the graph, where* $\boldsymbol{m}_0 \in R_0$*.*

10.3.2 Algorithm to compute the Reachability Graph

In this section, the algorithm presented in Section 10.2 is simplified, and it can be used to compute the RG of HPN.

The general idea of the algorithm is to build the RG of the HPN system recursively in two steps: given a reachable marking or set of markings (initially, $\boldsymbol{m}_0$); (1) calculate the set of markings which are reachable considering only continuous firings, which constitute a node of the RG; and (2) consider the discrete firing of each discrete transition, which generates an arc of the RG from that node to a new one, which is explored in the same way.

Algorithm 1 initializes the set of nodes and the set of arcs, and calls the recursive algorithm $explore_H$ (Algorithm 2) with the initial marking, which is the first set to be “explored”.

Algorithm 2 takes a set of markings, R, and calculates the RG from it (unless R was already considered, or a stopping condition holds, as explained after Example 10.7). First, the markings which are reachable considering only continuous firings (in any possible fireable amount) are calculated, which constitute a set. This is done with the function $continuousM_H$ (see Theorem 10.3):

$$
\begin{aligned}
continuousM_H(\mathcal{N}, R) = \quad \{\boldsymbol{m} \mid \quad & \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \\
& \boldsymbol{m}_0 \in R, \boldsymbol{m} \geq \boldsymbol{0}, \boldsymbol{\sigma} \geq \boldsymbol{0}, \\
& ||\boldsymbol{\sigma}|| \in \mathrm{FS}_H(\mathcal{N}, \boldsymbol{m}_0) \cap \mathrm{FS}_H(\mathcal{N}^{-1}, \boldsymbol{m})\}
\end{aligned}
\tag{10.5}
$$

Then, for every possible discrete transition t_j, i.e., $\mu_j = \infty$, the markings due to its discrete firing in an amount equal to 1 are calculated with the function $discreteM_H$:

$$
\begin{aligned}
discreteM_H(\mathcal{N}, R, t_j) = \quad \{\boldsymbol{m} \mid \quad & \boldsymbol{m} = \boldsymbol{m}_0 + C[P, t_j], \\
& \boldsymbol{m}_0 \in R, enab(t_j, \boldsymbol{m}_0) \geq 1\}
\end{aligned}
\tag{10.6}
$$

Once the markings due to the discrete firing, *d*, are calculated, an arc of the RG is created. This arc goes from the current node which is being explored to a new node which will be created when $explore_H$ will be recursively called with the new set of markings *d*.

Stopping condition. As discussed for HAPN, the RG of a HPN system may also have an infinite number of nodes. The same stopping condition $stoppingCond(R, setofNodes)$

Algorithm 6: $calculateRG_H$

Input: PN ($\mathcal{N}$), initialMarking ($\boldsymbol{m}_0 \in \mathbb{R}^{|P|}_{\geq 0}$)
Output: Reachability Graph (RG)

$setofNodes = \emptyset$
$setofArcs = \emptyset$
$R_0 = \text{explore}_H(\mathcal{N}, \{\boldsymbol{m}_0\})$
$RG = \langle setofNodes, setofArcs \rangle$
return RG

Algorithm 7: $explore_H$

Input: PN ($\mathcal{N}$), setOfMarkings ($R \subseteq \mathbb{R}^{|P|}_{\geq 0}$)
Output: setOfMarkings (n)

if $R \notin setofNodes \wedge not\ stoppingCond(R, setofNodes)$ **then**
    $n = continuousM_H(\mathcal{N}, R)$
    $setofNodes = setofNodes \cup n$
    for $t_j \in T\ s.t.\ \mu_j = \infty$ **do**
        $d = discreteM_H(\mathcal{N}, n, t_j)$
        if $d \neq \emptyset$ **then**
            $setofArcs = setofArcs \cup \langle n, explore_H(\mathcal{N}, d), t_j \rangle$
        end
    end
end
return n

is checked in the HPN algorithm.

Reachability set computation. The RS of a HPN system, $RS_H(\mathcal{N}, \boldsymbol{m}_0)$, is the union of all the homogeneous regions included in $\text{RG}_H(\mathcal{N}, \boldsymbol{m}_0)$ which have been computed with Algorithms 5 and 11.

Example 10.12 ***A simple HPN***
Let us first apply the algorithm to an example, and then compare the algorithm proposed here with other technique proposed in the literature for the computation of the RG of HPN. The HPN depicted in Fig. 10.8(a) is taken from [DA10], in which transitions t_1 and t_2 are discrete *and t_3 and t_4 are* continuous *(i.e., $\boldsymbol{\mu} = (\infty, \infty, 0, 0)$) and the initial marking is $\boldsymbol{m}_0 = (2, 0, 0, 3)$.*

First, $calculateRG_H(\mathcal{N}, \boldsymbol{m}_0)$ is called. Then, in the algorithm, $explore_H$ is called with this initial marking: $explore_H(\mathcal{N}, \boldsymbol{m}_0)$. The RG obtained by the algorithm is depicted in Fig. 10.8(c).

The union of all the sets of the RG gives the RS (see Fig. 10.8(b)). Its RS is represented in the axes $m[p_1], m[p_3]$, because the marking of the other places can be calculated from them: $m[p_2] = 2 - m[p_1]$ and $m[p_4] = 3 - m[p_3]$.

The RG obtained with Algorithm 5 (Fig. 10.8(c)) has three nodes, each of them containing a set defined in $\mathbb{R}^{|P|}_{\geq 0}$. The firings of the discrete transitions make the marking move from one set to another. Notice that, as specified by $continuousM_H(\mathcal{N}, \boldsymbol{m}_0)$, an arc labeled with discrete transition t_1 can be taken from those markings at which t_1 is enabled. Moreover, the firings of the continuous transitions are codified inside the reachable sets.

Using the technique proposed by David and Alla [DA10], the RG of this HPN (Fig. 4.21, page 140) consists of 9 nodes together with the interactions of discrete and continuous transitions among those nodes. An arc labeled with a discrete transition in the graph represents that it has been fired (an amount of 1). However, the firing of continuous transitions is not measured explicitly in the arcs of the RG. In our method, the firing of discrete transitions is also explicit, but the firing of continuous transitions is compacted inside the markings sets.

The computation of the RG of HPN system which models a production system is considered in Example 10.13.

Example 10.13 ***A production system***
The HPN in Fig. 10.9 is a simplified model of an example in [DA10], in which p_1, p_2, p_3 model one production line, and p_4, p_5, p_6 model another one. Both lines share a resource, modeled by p_8 (marked when the resource is idle), p_7 and p_9.

From the initial marking $\boldsymbol{m}_0 = (500, 0, 0, 700, 0, 0, 0, 1, 0)$, continuous transitions t_1 and t_4 are enabled; considering their possible firings, the set of markings represented in node R_0 is obtained (see Fig. 10.10). From R_0, discrete transitions t_7 or t_8 can be fired when they are enabled. If the firing of t_7 is considered, the arc from R_0 to R_1 is created. R_1 contains the set of markings reachable after firing t_7, considering the possible firing of continuous transitions. The algorithm is recursively called from the new created nodes, and the complete

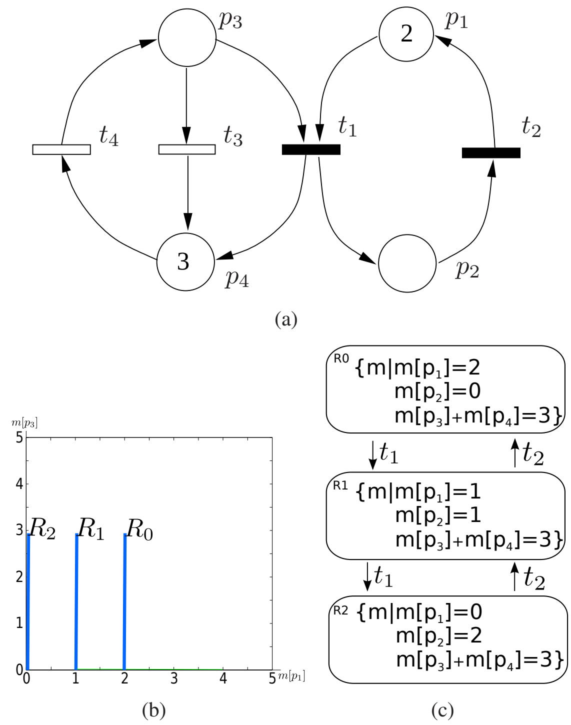


Figure 10.8: (a) HPN system [DA10]. (b) Its Reachability Set. (c) Its Reachability Graph.

RG is obtained, which is depicted in Fig. 10.10. In the figure, m_i *represents* $m[p_i]$*, markings are non negative values, and markings which equal to 0 are not shown. Every arc is labeled with the discrete transition which is fired.*

Exploiting the Reachability Graph. Some general considerations can be done on the obtained RG. It can be seen that the nodes computed by the algorithm are not *disjoint* (in contrast with the RG of DPN, in which each node corresponds to a marking). For instance, $R_0 \cap R_3 \neq \emptyset$ in Example 10.13 (Fig. 10.10). Moreover, the nodes of the RG of the HPN do not correspond directly to nodes of a subyacent discrete RG.

Because of the representation of the RG, in which the continuous behaviour is contained inside of the nodes but it is not explicitly seen in the arcs, the RGs from two markings $\boldsymbol{m}$ and $\boldsymbol{m}'$ mutually reachable can be different, although they describe the same *reachability sets*. For instance, the RG in Fig. 10.10 is obtained from $\boldsymbol{m}_0 = (500, 0, 0, 700, 0, 0, 0, 1, 0)$, but the RG obtained from $\boldsymbol{m}'_0 = (0, 0, 500, 0, 0, 700, 0, 1, 0)$ would start from an initial node $\{m_3 = 500, m_6 = 700, m_8 = 1\}$, which is not a node itself in Fig. 10.10 but it is included in node R_{11}. Both RG describe the same reachable markings. As a consequence, the reachability of a marking $\boldsymbol{m}'$ from a marking $\boldsymbol{m} \in RS_H(\mathcal{N}, \boldsymbol{m}_0)$ cannot be "directly" checked over

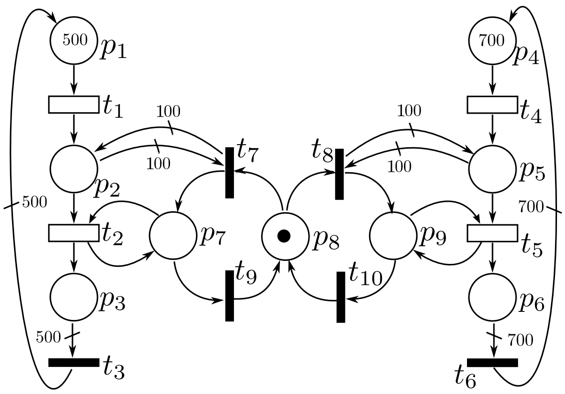


Figure 10.9: HPN which models a production system in which two production lines share a resource.

$\text{RG}_H(\mathcal{N}, \boldsymbol{m}_0)$, but it should be checked over $\text{RG}_H(\mathcal{N}, \boldsymbol{m})$.

Fortunately, the computed RG can be exploited to check "directly" certain properties of $\langle\mathcal{N}, \boldsymbol{m}_0\rangle_H$, by checking every obtained node R_i (if node R_j is a subset of R_i, checking R_j is not needed):

- The bound of a place, defined as $B(p) = max\{m[p] | \boldsymbol{m} \in RS_H(\mathcal{N}, \boldsymbol{m}_0)\}$ can be easily computed by calculating the following LPP in each node R_i: $z_i = max\ m[p]$ s.t. $\boldsymbol{m} \in R_i$, where $B(p) = max\{z_i\}$. For instance, $B(p_2) = 500$ in the example.

- Mutual exclusion property can also be checked by examining each node of the RG. In the example, places p_7, p_8 and p_8 are in mutual exclusion.

- In order to check deadlock-freeness, the deadlock condition (no transition is enabled)[Sil+11] has to be checked in each node. In the example, the marking $\boldsymbol{m}_d = (0, 50, 450, 0, 50, 650, 0, 1, 0)$ is a deadlock (no transition is enabled at $\boldsymbol{m}_d$) and it is reachable in node R_{16}.

- The deviation bound [SC88] of two transitions t and t', $DB(t, t')$, denotes the maximal amount that t can be fired without firing t'; and it can be checked in the RG if $t' \in T^d$. In case $t \in T^d$, it can be checked by looking at the arcs of the RG, as in discrete RGs. In case $t \in T^c$, $DB(t, t')$ can be calculated with the following LPP over the nodes R_z which are reachable from R_0 without considering the arcs labeled by t': $max\ \boldsymbol{\sigma}(t)$ s.t. $\boldsymbol{m} + \boldsymbol{C} \cdot \boldsymbol{\sigma} \geq 0, \boldsymbol{m} \in R_z$; and adding the resulting values. Considering the deviation bounds between two discrete transitions, B-fairness property [SC88] can be checked.

The RG obtained with the technique proposed in [DA10] can also be used to check deadlock-freeness, boundedness, mutual exclusion, and deviation bound properties in an

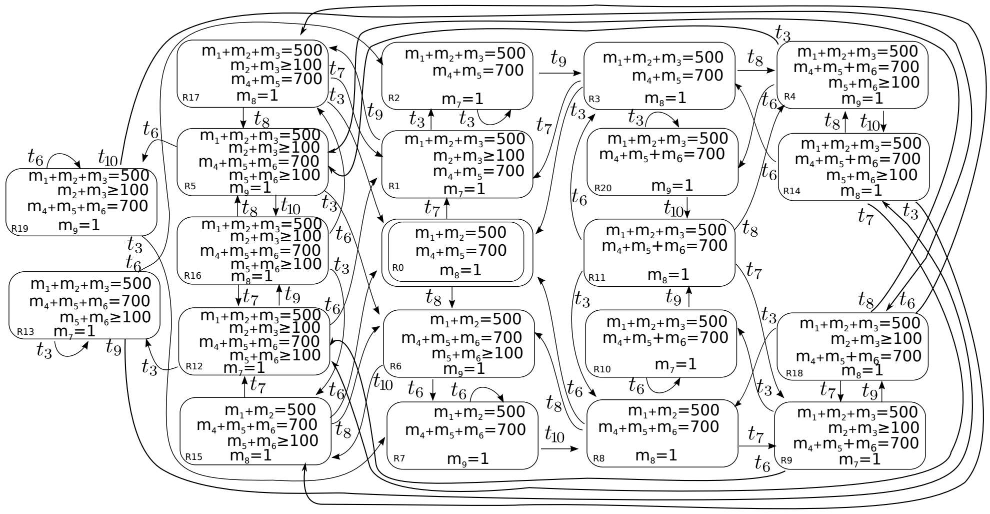


Figure 10.10: RG of the HPN system in Fig. 10.9; m_i represents $m[p_i]$, zero markings are not shown.

analogous way, by checking every node of the RG. However, the basic reachability property (given a marking $\boldsymbol{m}$, is it reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_H$?) requires more computation.

10.4 Conclusions

The formalism of *Hybrid Adaptive Petri Nets* (HAPN) proposes the partial relaxation of transitions, which *adapt* its behaviour to the workload of the system.

This chapter has focused on reachability of autonomous HAPN. A recursive algorithm to compute a RG for autonomous HAPN has been proposed. The algorithm compacts the markings which are due to continuous firings of transitions into nodes of the RG, while the firings of discrete transitions are explicitly represented with arcs connecting the nodes.

The nodes of the RG for HAPN are *homogeneous regions*, what means that for each transition, its enabling degree is either higher or equal to its threshold, or lower or equal to its threshold. The RG has two types of arcs: ε-arcs, due to mode changes; and D-arcs, due to discrete firings of transitions. Given a marking, the proposed recursive algorithm characterizes the markings reachable due to continuous firings, and the possible ε-arcs and D-arcs from it. These algorithms integrate some reachability concepts and definitions about DPN, CPN, and HPN.

It has been seen that the RG of a HAPN or a HPN can be infinite. In HPN, some transitions are continuous and the rest of the transitions remain discrete. In HAPN, a threshold is defined for each transition, such that it behaves as continuous when its enabling degree is *higher* than its threshold and as discrete otherwise.

Moreover, the algorithm is simplified to be applied to HPN, a formalism which is included in HAPN, and it is compared with an existing method. The algorithm compacts the markings which are due to continuous firings of transitions into nodes of the RG, while the firings of discrete transitions are explicitly represented with arcs connecting the nodes. The RS of either HPN or HAPN can be straightforwardly obtained from its RG.

11

Conclusions and perspectives

"*After climbing a great hill, one only finds that there are many more hills to climb*".
Nelson Mandela

Fluidization of *Petri Nets* (PN) is a relaxation technique to tackle the *state explosion problem* inherent to *Discrete PN* (DPN). Fluidization results in *Continuous PN* (CPN), which were proposed almost 30 years ago in the conference ICATPN'1987 in Zaragoza. They have been studied both as a formalism to make easier the study of a original DPN and as a modelling formalism on itself.

This thesis contributes to the study and understanding of the fluidization process that transforms a given DPN to a CPN. It also considers the properties of the CPN as a standalone formalism. Moreover, it proposes some improvements on the fluidization process, such as the partial fluidization of the system in which transitions adapt dynamically its behaviour to the workload of the system, among others. These improvements are specially indicated when the population of the system is neither very low (because fluidization would not be needed) nor very high (because fluidified approximation is good enough), but between them, where low and high populations are combined.

The main contributions of this thesis can be summarized as follows:

- Transformations of the original discrete PN system which do not modify its behaviour, but they obtain a continuous PN system to be more faithful to the original one.
- Establishment of conditions under which a property such as boundedness, B-fairness, deadlock-freeness, liveness or reversibility is preserved by the fluidization of an untimed DPN.
- Improvement of the upper and lower bounds of the computational complexity of checking properties in continuous PN.

- Analysis of the throughput approximation of stochastic Petri nets. The *bound reaching problem* has been identified as a particular situation in which the approximation may be particularly bad. A particular construction based in *Infinite Servers Semantics* (ISS), and denoted as ρ-$semantics$, has been proposed to deal with this situation.
- A formalism which considers partial fluidization of transitions, *Hybrid Adaptive Petri Nets* (HAPN), has been formally defined for untimed systems. They can be seen as a conceptual framework which includes discrete, continuous and hybrid PN. Deadlock-freeness preservation has been taken into account, and an algorithm to compute its *Reachability Graph* (RG) and *Reachability Set* (RS) has been proposed.

Fluidization of discrete PN systems removes the integrality constraint of the discrete PN formalism. One of the consequences is that, under general assumptions (non existence of empty siphons at the initial marking and consistency), all the *spurious* markings of the system become (lim-)reachable by the continuous relaxation. One of the contributions of this thesis is the proposal of some techniques to remove such spurious markings, specially when they may be deadlocks. A preliminary technique is recalled from the literature [STC96; Sil+11], which removes those *spurious* markings which are due to the emptying of a trap. It proposes the addition of an *implicit* place which forces a marking invariant (token conservation law). Moreover, four more kinds of *implicit* places to remove *spurious* solutions have been proposed (see Chapter 3): (a) places which avoid *empty siphons* which have been previously identified as non reachable; (b) *vertex cutting* places, which cut a non natural vertex of the polytope defined by the solutions of the state equation; (c) *marking truncation* places, which avoid those solutions in which the marking of a place is above its natural structural bound; and (d) *enabling truncation* places do not modify the set of reachable markings, but they may modify the behaviour of the continuous system when it is fluidified. The elimination of these markings provides an improvement on the fluidization of both untimed and timed PN. Other results from discrete PN systems such as the *Linear Enabling Functions* (LEF) have been recalled. They are used to establish (homothetic) deadlock-freeness of a discrete PN system or to add a *representative* place which represent the discrete enabling of *join* transition.

Due to the fact that *spurious* deadlocks of the DPN can become reachable in the CPN, a PN system which is deadlock-free as discrete can reach a deadlock as continuous, considering finite or infinite firing sequences. Other contribution of this thesis has been the determination of the conditions under which a given property is preserved by the fluidization of the untimed system. It has been established (see Chapter 5) that given a synchronic property (boundedness, B-fairness, etc) in which there are not empty siphons at the initial marking, structural boundedness of the net, boundedness of the discrete system and (lim-)boundedness of the continuous system coincide. Moreover, it has been stated that considering a property Π among deadlock-freeness, liveness or reversibility, the following holds: (a) lim-Π in a continuous PN system implies homothetic Π in the discrete one; and (b) a homothetic property Π in a discrete PN system is preserved by its continuous counterpart, which is also Π (but it may be not lim-Π). The reverses are not true.

Once the fluidization process has been performed, it is also interesting to consider the formalism of continuous PN itself. From early works in continuous PN [RTS99], an interesting issue has been to determine the decidability and computational complexity of the reachability of a marking $\boldsymbol{m}$ in a given $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle_C$, or the analysis of properties such as boundedness or deadlock-freeness. Some previous works have provided efficient methods for reachability and boundedness for consistent CPN systems in which no siphon is empty at $\boldsymbol{m}_0$. In this thesis (see Chapter 6), it has been proved that both problems ((lim-)reachability and (lim-)boundedness) are polynomial time even in the general case (non consistent nets, and allowing empty siphons). Moreover, a characterization of reachability set inclusion has been proposed, which is undecidable in *discrete* PN systems. Hence, fluidization does not only reduce the complexity of checking some properties, but also some properties which were not decidable in the DPN become decidable in the CPN. Finally, lower bounds have been established for those problems. Reachability and boundedness problems in CPN are *PTIME-complete*. Moreover, (lim-)deadlock-freeness, and (lim-)liveness are *coNP-complete*, even considering consistency of $\mathcal{N}$ and no empty siphons at $\boldsymbol{m}_0$. It has been proved that reversibility is *coNP-hard*.

By the addition of time, *Timed Continuous Petri Nets* (TCPN) are obtained. In this thesis, the focus has been on the approximation of the steady state throughput of discrete *Stochastic Petri Nets* (SPN). It is a known result that TCPN provide a good approximation of the throughput of SPN when the population of the system is large. In this work (see Chapter 8), a particular situation has been identified, in which the approximation is specially bad. It is the case in which the bound of place p is equal to k and there is an arc from p to t whose weight is equal to k; and, in general, the case in which the structural enabling bound of a transition t equal to 1. In this situation, called *Bound Reaching Problem* (BRP), transition t can be fired in the SPN only when $m[p] = k$ (i.e., its bound is reached), while t can be fired in the TCPN.

A particular way to handle the firing of the transitions, based in TCPN under ISS has been proposed to tackle the BRP, denoted as ρ-$semantics$. It is the abbreviation of a structure of ISS and immediate transitions, in which the flow of transition t is defined as a piecewise function which depends on parameter ρ. Transition t does not fire if $m[p] < k - \rho$, and it only fires if $m[p] \geq k - \rho$. This semantics is inspired in the fact that the SPN transition has to wait (i.e. it is not fire) during certain period of time (or certain degree of marking) before firing. It preserves some properties from ISS such as homothetic monotonicity of the flow w.r.t. the firing rates. The ρ-$semantics$ gives specially good results when the BRP appears in a transition t such that ${}^\bullet t = \{p\}$ and $|{}^\bullet p| = |p^\bullet| = 1$. Moreover, it has been generalized for the case in which $|{}^\bullet t| > 1$, by means of the *representative* places proposed for discrete systems.

Among the methods to improve the fluidization process of discrete PN, a formalism for the partial fluidization of transitions, has been considered, the hybrid adaptive Petri nets (see Chapters 9 and 10). HAPN allow the transitions to adapt its behaviour to the workload of the system: a threshold μ is associated to each transition t, such that t behaves as continuous if

$enab(t, m) > \mu$, and as discrete in other case. Depending on the value of the thresholds, the transitions can behave as discrete, as continuous or as adaptive. Consequently, HAPN can be also seen as a conceptual framework which contains DPN, CPN and *Hybrid Petri Nets* (HPN).

HAPN were considered in [YLL09] for the partial fluidization of timed systems. The contribution of this thesis (see Chapter 9) is to formally characterize the HAPN in the untimed context. Reachability set inclusion among DPN, CPN and HAPN has been studied. It has been proved that $RS_D(\mathcal{N}, \boldsymbol{m}_0)$ is included in $RS_A(\mathcal{N}, \boldsymbol{m}_0)$ for ordinary PN with natural thresholds ($\boldsymbol{\mu} \in \mathbb{N}^{|T|}$). However, it is not true if not ordinary nets or not natural thresholds are considered. Moreover, it has been proved that $RS_A(\mathcal{N}, \boldsymbol{m}_0)$ is included in $RS_C(\mathcal{N}, \boldsymbol{m}_0)$ for any net system and for any threshold vector $\boldsymbol{\mu} \in \mathbb{R}^{|T|}$. The preservation of deadlock-freeness property has also been studied. Given an ordinary choice-free DPN which is deadlock-free, its HAPN counterpart is also deadlock-free for any $\boldsymbol{\mu} \in \mathbb{N}^{|T|}$. However, if non-ordinary or non choice free DPN are considered, it is not true. It has been proved that deadlock-free ordinary HAPN systems which natural thresholds are also deadlock-free as DPN.

As in the case of CPN, an interesting issue related to the definition of HAPN is the formal characterization of its RS. Moreover, the potential evolutions of the system can be captured in its RG. Here, a definition of a particular RG has been proposed (see Chapter 10). The nodes of the graph contain convex sets of markings which are reachable with continuous firings, and the arcs of the graph represent discrete firings of transitions. The particularity of this RG is that the continuous firings are not explicit in the graph, while the discrete firings are explicit. The main contribution of Chapter 10 is the algorithm to build the RG of HAPN.

Moreover, due to the fact that HAPN are a general formalism which includes HPN, the proposed algorithm can be used for the reachability analysis of HPN. The algorithm has been simplified to provide a compact method which can be directly applied to HPN. Because some continuous behaviour is collapsed inside the nodes, it can obtain a RG smaller than other methods from the literature, such as the one informally described in [DA10].

Considering the different investigations which have been carried out during this thesis, and the different topics related to fluidization of PN which have been investigated, some of the lines which remain open or which would require more research are enumerated below:

- One of the kinds of *implicit* places proposed in Chapter 3 removes *spurious* deadlocks by avoiding a siphon from been emptied. However, we need to know *a priori* if the marking is spurious or not. Otherwise, we could *provide deadlock-freeness* to a system which deadlocks as discrete. This topic was considered in [GV99], but it requires more investigation.

- The existence of spurious solutions in different PN subclasses has been studied in Chapter 5. Considering other net subclasses or structural conditions would also be interesting.

- The computational complexity of some known and well defined properties in CPN has been studied in Chapter 6. A possible extension of this work would be to consider also other properties, which could be defined with a temporal logic.

- Determining the decidability and computational complexity of analysing some properties of HPN (or even HAPN) could be also considered. The whole formalism or some PN subclasses could be taken in account.

- The ρ-$semantics$ has been proposed in Chapter 8 to handle the BRP in those synchronizations which have an arc weight equal to k. However, the proposed formula to compute the value of ρ does not modify the behaviour of a transition whose input arc weights are equal to 1. Different ways to compute ρ can be proposed for different net structures.

- A first method to apply the ρ-$semantics$ to *join* transitions has been proposed. However, more research would improve the selection of the appropriate value of ρ or it would improve the formula for the flow of the transition itself.

- Steady state throughput approximation of SPN has been considered in Chapters 7 and 8. However, the approximation of the transient behaviour can be also considered. Moreover, the fluidization of not only Markovian PN but also firing semantics with different probabilistic functions would be interesting.

- Considering HAPN, an interesting open issue is the selection of the thresholds of a given PN system to preserve deadlock-freeness property. This topic is briefly considered in Chapter 9, but it requires further investigation.

- In this work, HAPN in an untimed framework have been studied. It would be very interesting to extend the results to the timed HAPN. A first step would be to define the firing semantics of a transition t, considering three different situations for the enabling degree of a transition: over the threshold ($enab(t_j, \boldsymbol{m}) > \mu_j$), below the threshold ($enab(t_j, \boldsymbol{m}) < \mu_j$) or equal to the threshold ($enab(t_j, \boldsymbol{m}) = \mu_j$).

List of Acronyms

B Boundedness

BRP Bound Reaching Problem

CPN Continuous Petri net

DB Deviation Bound

DES Discrete Event System

DF Deadlock-freeness

DPN Discrete Petri net

DSSP Deterministically Synchronized Sequential Processes

(DS)*SP Multi-level Deterministically Synchronized Sequential Processes

EQ Equal Conflict net

EXPTIME Complexity class. It refers to ”Exponential time”

HAPN Hybrid Adaptive Petri Net

HPN Hybrid Petri Net

IPP Integer Programming Problem

ISS Infinite Servers Semantics

JF Join Free net

L Liveness

LPP Linear Programming Problem

MTS Mono-T-Semiflow net

NC Complexity class. It refers to ”Nick's Class”

NP Complexity class. It refers to ”Nondeterministic Polynomial time”

PN Petri net

PTIME Complexity class. It refers to ”Polynomial time”

RG Reachability Graph

RS Reachability Set

S^{3}PR Systems of Simple Sequential Processes with Resources

SB Structural Boundedness, Structural Bound

SEB Structural enabling bound

SPN Stochastic Petri net

TCPN Timed Continuous Petri net

Bibliography

[AD98] H. Alla and R. David. “Continuous and Hybrid Petri Nets”. In: *Journal of Circuits, Systems, and Computers* 8.1 (1998), pp. 159–188.

[AF92] D. Avis and K. Fukuda. “A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra”. In: *Discrete & Computational Geometry* 8.1 (1992), pp. 295–313.

[AFP02] D. Avis, K. Fukuda, and S. Picozzi. “On canonical representations of convex polyhedra”. In: *Mathematical Software, Proceedings of the First International Congress of Mathematical Software*. Ed. by A. M. Cohen, X.-S. Gao, and N. Takayama. World Scientific Publishing, 2002, pp. 350–360.

[Alf+05] A. Alfonsi et al. “Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems”. In: *ESAIM: Proc.* 14 (2005), pp. 1–13.

[AM+95] M. Ajmone-Marsan et al. *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1995.

[Bal+87] G. Balbo et al. “Generalized Stochastic Petri Nets for the Performance Evaluation of FMS”. In: *Invited Sessions: Petri Nets and Flexible Manufacturing. IEEE Int. Conf. on Robotics and Automation*. Ed. by M. Silva and T. Murata. Raleigh, NC, USA, Apr. 1987, pp. 1013–1018.

[Bal+96] E. Balas et al. “Gomory cuts revisited”. In: *Operations Research Letters* 19.1 (1996), pp. 1 –9.

[BC94] J. L. Briz and J. M. Colom. “Implementation of Weighted P/T Nets based on Linear Enabling Functions”. In: *Application and Theory of Petri Nets 1994*. Ed. by R. Valette. Vol. 815. Lecture Notes in Computer Science. Springer, 1994, pp. 99–118.

[BCP96] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Philadelphia, PA: SIAM, 1996.

[BCS94] J. L. Briz, J.M. Colom, and M. Silva. “Simulation of Petri nets and linear enabling functions”. In: *Systems, Man, and Cybernetics, 1994. Humans, Information and Technology., 1994 IEEE International Conference on*. Vol. 2. 1994, 1671–1676 vol.2.

[BGM00] F. Balduzzi, A. Giua, and G. Menga. “First-Order Hybrid Petri nets: A model for optimization and control”. In: *IEEE Trans. on Robotics and Automation* 16.4 (2000), pp. 382–399.

[BGS01] F. Balduzzi, A. Giua, and C. Seatzu. “Modelling and Simulation of Manufacturing Systems with First-Order Hybrid Petri Nets”. In: *Int. J. of Production Research* 39.2 (2001). Special Issue on Modelling, Specification and analysis of Manufacturing Systems, pp. 255–282.

[BHZ05] R. Bagnara, P. M. Hill, and E. Zaffanella. “Not necessarily closed convex polyhedra and the double description method”. In: *Formal Aspects of Computing* 17.2 (2005), pp. 222–257.

[Bra83] G. W. Brams. *Réseaux de Petri: Théorie et Pratique*. Masson, 1983.

[Cab+10] M. P. Cabasino et al. “Fault diagnosis of manufacturing systems using continuous Petri nets”. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey*. IEEE, 2010, pp. 534–539.

[CCS91] J. Campos, G. Chiola, and M. Silva. “Ergodicity and Throughput Bounds of Petri Net with Unique Consistent Firing Count Vector”. In: *IEEE Trans. on Software Engineering* 17.2 (1991), pp. 117–125.

[CCS92] J. Campos, J.M. Colom, and M. Silva. “Improving Throughput Upper Bounds for Net Based Models of Manufacturing Systems”. In: *Robotics and Flexible Manufacturing Systems*. Ed. by J.C. Gentina and S.G. Tzafestas. Amsterdam, The Netherlands: Elsevier Science Publishers B.V. (North-Holland), 1992, pp. 281–294.

[CLP01] B. Codenotti, M. Leoncini, and F. P. Preparata. “The Role of Arithmetic in Fast Parallel Matrix Inversion”. In: *Algorithmica* 30.4 (2001), pp. 685–707.

[Com+71] F. Commoner et al. “Marked Directed Graphs”. In: *Journal on Computer Systems Science* 5 (1971), pp. 72–79.

[Cor12] G. Cornuejols. “The Ongoing Story of Gomory Cuts”. In: *Documenta Mathematica - Optimization Stories* (2012), pp. 221–226.

[DA10] R. David and H. Alla. *Discrete, Continuous and Hybrid Petri Nets*. Revisited 2^{nd} edition. (First edition, 2004). Berlin: Springer, 2010.

[DA87] R. David and H. Alla. “Continuous Petri Nets”. In: *Proc. of the 8th European Workshop on Application and Theory of Petri Nets*. Zaragoza, Spain, 1987, pp. 275–294.

[DE95] J. Desel and J. Esparza. *Free Choice Petri Nets*. Cambridge Tracts in Theoretical Computer Science 40, 1995.

[DiC+93] F. DiCesare et al. *Practice of Petri Nets in Manufacturing*. Chapman & Hall, 1993.

[Dun11] J. Dunkel. “The Gomory-Chvátal closure : polyhedrality, complexity, and extensions”. PhD thesis. Massachusetts Institute of Technology, 2011.

[ECM95] J. Ezpeleta, J. M. Colom, and J. Martínez. “A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems”. In: *IEEE Trans. on Robotics and Automation* 11.2 (1995), pp. 173–184.

[ECS93] J. Ezpeleta, J. M. Couvreur, and M. Silva. “A New Technique for Finding a Generating Family of Siphons, Traps and ST-Components. Application to Coloured Petri Nets”. In: *Advances in Petri Nets 1993*. Ed. by G. Rozenberg. Vol. 674. Lecture Notes in Computer Science. Springer, 1993, pp. 126–147.

[EK86] S. Ethier and T. Kurtz. *Markov Processes: Characterization and Convergence*. John Wiley, 1986.

[EN94] J. Esparza and M. Nielsen. “Decidability Issues for Petri Nets - a survey”. In: *Elektronische Informationsverarbeitung und Kybernetik* 30.3 (1994), pp. 143–160.

[FH13] E. Fraca and S. Haddad. “Complexity Analysis of Continuous Petri Nets”. In: *Application and Theory of Petri Nets and Concurrency*. Ed. by José-Manuel Colom and Jörg Desel. Vol. 7927. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 170–189. ISBN: 978-3-642-38696-1.

[FH15] E. Fraca and S. Haddad. “Complexity Analysis of Continuous Petri Nets”. In: *Fundamenta informaticae* 136.1 (2015), pp. 1–28.

[FJS s] E. Fraca, J. Júlvez, and M. Silva. “Improving deterministic fluid approximation of stochastic PN”. In: (– Submmited to an indexed journal –).

[FJS12] E. Fraca, J. Júlvez, and M. Silva. “Marking homothetic monotonicity and fluidization of untimed Petri nets”. In: *In Proc. of the 11th Int. Workshop on Discrete Event Systems, WODES*. 2012, pp. 21–27.

[FJS14a] E. Fraca, J. Júlvez, and M. Silva. “On the fluidization of Petri nets and marking homothecy”. In: *Nonlinear Analysis: Hybrid Systems* 12 (2014), pp. 3 –19.

[FJS14b] E. Fraca, J. Júlvez, and M. Silva. “The Bound Reaching Problem on the fluidization of timed PN”. In: *In Proc. of the 12th Int. Workshop on Discrete Event Systems, WODES*. 2014, pp. 142–148.

[FJS15] E. Fraca, J. Júlvez, and M. Silva. “Hybrid and Hybrid Adaptive Petri Nets: on the computation of a Reachability Graph”. In: *Nonlinear Analysis: Hybrid Systems* 16 (2015), pp. 24 –39.

[FN89] G. Florin and S. Natkin. “Necessary and Sufficient Ergodicity Condition for Open Synchronized Queueing Networks”. In: *IEEE Trans. Softw. Eng.* 15.4 (Apr. 1989), pp. 367–380. ISSN: 0098-5589.

[Fra+11] E. Fraca et al. “On reachability and deadlock-freeness of Hybrid Adaptive Petri nets”. In: *18th IFAC World Congress*. 2011.

[Gil77] D.T. Gillispie. “Exact stochastic simulation of coupled chemical reactions”. In: *The Journal of Physical Chemistry* 81 (1977), pp. 2340–2361.

[GMLMA12] B. Gudiño-Mendoza, E. López-Mellado, and H. Alla. “Modeling and simulation of water distribution systems using timed hybrid Petri nets”. In: *Simulation* 88.3 (2012), pp. 329–347.

[Gri+06] M. Griffith et al. “Dynamic partitioning for hybrid simulation of the bistable HIV-1 transactivation network”. In: *Bioinformatics* 22.22 (2006), pp. 2782–2789.

[GV99] F. García-Vallés. “Contributions to the Structural and Symbolic Analysis of Place/Transition Nets with Applications to Flexible Manufacturing Systems and Asynchronous Circuits”. PhD thesis. DIIS. Univ. Zaragoza, 1999.

[GVC99] F. García-Vallés and J.M. Colom. “Implicit Places in Net Systems”. In: *Petri Nets and Performance Models (PNPM’99)*. IEEE Computer Society Press, 1999, pp. 104–113.

[Hac72] M. H. T. Hack. “Analysis of Production Schemata by Petri Nets”. (Corrections in *Computation Structures Note* 17, 1974). MA thesis. Cambridge, MA, USA: M.I.T., 1972.

[Har02] J. M. Harrison. “Stochastic Processing Networks and Activity Analysis”. In: *Analytic Methods in Applied Probability*. Ed. by Y. Suhov. In Memory of Fridrik Karpelevich. American Mathematical Society, Providence, RI. 2002.

[HLEM01] S. Hennequin, D. Lefebvre, and A. El-Moudni. “Fuzzy multimodel of timed Petri nets”. In: *IEEE Transactions on Systems, Man, and Cyb., Part B* 31.2 (2001), pp. 245–251.

[JC03] L. Jiao and T.Y. Cheung. “Characterizing liveness monotonicity for weighted Petri nets in terms of siphon-based properties”. In: *International Journal of Foundations of Computer Science* 14.4 (2003), pp. 641–658.

[Jim+04] E. Jiménez et al. “Relaxed Continuous Views of Discrete Event Systems: Petri Nets, Forrester Diagrams and ODES”. In: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. The Hague, The Netherlands, Oct. 2004.

[JRS03] J. Júlvez, L. Recalde, and M. Silva. "On reachability in autonomous continuous Petri net systems". In: 24^{th} *Int. Conf. on Application and Theory of Petri Nets (ICATPN 2003), Lecture Notes in Computer Science* 2679 (2003). Ed. by W. van der Aalst and E. Best, pp. 221–240.

[JS02] J. Jacod and A. Shiryaev. *Limit Theorems for Stochastic Processes*. Springer, 2002.

[LL12] D. Lefebvre and E. Leclercq. "Piecewise constant timed continuous PNs for the steady state estimation of stochastic PNs". In: *Discrete Event Dynamic Systems* 22.2 (2012), pp. 179–196.

[Mar+04] M. Ajmone Marsan et al. *Modelling With Generalised Stochastic Petri Nets*. John Wiley and Sons, 2004. URL: http://www.di.unito.it/~greatspn/GSPN-Wiley/.

[Mol82] M. K. Molloy. "Performance Analysis Using Stochastic Petri Nets". In: *IEEE Trans. on Computers* 31.9 (1982), pp. 913–917.

[MRS09] C. Mahulea, L. Recalde, and M. Silva. "Basic server semantics and performance monotonicity of continuous Petri nets". In: *Discrete Event Dynamic Systems* 19.2 (2009), pp. 189–212. ISSN: 0924-6703.

[MRS10] C. Mahulea, L. Recalde, and M. Silva. "Observability of continuous Petri nets with infinite server semantics". In: *Nonlinear Analysis: Hybrid Systems* 4.2 (2010), pp. 219–232.

[MS12] A.L. Meyer and M. Silva. "Symmetry Reductions in Timed Continuous Petri Nets under Infinite Server Semantic". In: *Proceedings of the 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS'12)*. 2012, pp. 153–159.

[Mur89] T. Murata. "Petri Nets: Properties, Analysis and Applications". In: *Procs. of the IEEE* 77.4 (1989), pp. 541–580.

[OWW10] O. Oanea, H. Wimmel, and K Wolf. "New algorithms for deciding the siphon-trap property". In: *Proceedings of the 31st international conference on Applications and Theory of Petri Nets*. ICATPN 2010. Braga, Portugal: Springer-Verlag, 2010, pp. 267–286. ISBN: 3-642-13674-5, 978-3-642-13674-0.

[Pap94] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994. ISBN: 0201530821.

[Pet81] J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, 1981.

[PS98] C. H. Papadimitriou and K. Steigliz. *Combinatorial Optimization. Algorithms and Complexity*. Dover publications, second edition, 1998.

[Rec98] L. Recalde. “Structural Methods for the Design and Analysis of Concurrent Systems Modeled with Place/Transition Nets”. PhD thesis. DIIS. Univ. Zaragoza, 1998.

[RHS10] L. Recalde, S. Haddad, and M. Silva. “Continuous Petri Nets: Expressive Power and Decidability Issues”. In: *Int. Journal of Foundations of Computer Science* 21.2 (2010), pp. 235–256.

[RL+10] R. Ross-Leon et al. “Control of Metabolic Systems Modeled with Timed Continuous Petri Nets.” In: *ACSD/Petri Nets Workshops*. Vol. 827. CEUR Workshop Proceedings. 2010, pp. 87–102.

[RMS06] L. Recalde, C. Mahulea, and M. Silva. “Improving analysis and simulation of continuous Petri Nets”. In: *In Proc. of the 2nd IEEE Conference on Automation Science and Engineering, Shanghai, China.* Oct. 2006, pp. 7–12.

[RS01] L. Recalde and M. Silva. “Petri Nets Fluidification revisited: Semantics and Steady state”. In: *APII-JESA* 35.4 (2001), pp. 435–449.

[RTS01] L. Recalde, E. Teruel, and M. Silva. “Structure Theory of Multi-Level Deterministically Synchronized Sequential Processes”. In: *Theoretical Computer Science* 254.1-2 (2001), pp. 1–33.

[RTS98] L. Recalde, E. Teruel, and M. Silva. “Modeling and Analysis of Sequential Processes that Cooperate through Buffers”. In: *IEEE Trans. on Robotics and Automation* 14.2 (1998), pp. 267–277.

[RTS99] L. Recalde, E. Teruel, and M. Silva. “Autonomous Continuous P/T systems”. In: *Application and Theory of Petri Nets 1999*. Ed. by J. Kleijn S. Donatelli. Vol. 1639. Lecture Notes in Computer Science. Springer, 1999, pp. 107–126.

[SC88] M. Silva and J. M. Colom. “On the Computation of Structural Synchronic Invariants in P/T Nets”. In: *Advances in Petri Nets 1988*. Ed. by G. Rozenberg. Vol. 340. Lecture Notes in Computer Science. Springer, 1988, pp. 387–417.

[Sil+11] M. Silva et al. “On fluidization of discrete event models: observation and control of continuous Petri nets”. In: *Discrete Event Dynamic Systems: Theory and Applications* 21.4 (2011), pp. 427–497.

[Sil85] M. Silva. *Las Redes de Petri: en la Automática y la Informática*. Madrid: Ed. AC, 1985.

[Sil87] M. Silva. “Towards a Synchrony Theory for P/T Nets”. In: *Concurrency and Nets*. Ed. by K. Voss et al. Springer, 1987, pp. 435–460.

[SM92] Manuel Silva and Tadao Murata. “B-fairness and structural B-fairness in Petri net models of concurrent systems”. In: *Journal of Computer and System Sciences* 44.3 (1992), pp. 447 –477. ISSN: 0022-0000.

[SR02] M. Silva and L. Recalde. “Petri nets and integrality relaxations: A view of continuous Petri net models”. In: *IEEE Trans. on Systems, Man, and Cybernetics* 32.4 (2002), pp. 314–327.

[SR04] M. Silva and L. Recalde. "On fluidification of Petri net models: from discrete to hybrid and continuous models". In: *Annual Reviews in Control* 28 (2004), pp. 253–266.

[STC96] M. Silva, E. Teruel, and J.M. Colom. "Linear algebraic and linear programming techniques for the analysis of P/T net systems". In: *LNCS* 1491 (1996). Ed. by G. Rozenberg and W. Reisig, pp. 309–373.

[STC98] M. Silva, E. Teruel, and J. M. Colom. "Linear Algebraic and Linear Programming Techniques for the Analysis of Net Systems". In: *Lecture Notes in Computer Science* 1491 (1998), pp. 309–373.

[TCS93] E. Teruel, J. M. Colom, and M. Silva. "Linear Analysis of Deadlock-Freeness of Petri Net Models". In: *Procs. of the* 2^{nd} *European Control Conf.* Vol. 2. North-Holland, 1993, pp. 513–518.

[TCS97] E. Teruel, J. M. Colom, and M. Silva. "Choice-free Petri Nets: A Model for Deterministic Concurrent Systems with Bulk Services and Arrivals". In: *IEEE Trans. on Systems, Man, and Cybernetics* 27.1 (1997), pp. 73–83.

[TS96] E. Teruel and M. Silva. "Structure Theory of Equal Conflict Systems". In: *Theoretical Computer Science* 153.1 (1996), pp. 271–300.

[VS12] C. R. Vázquez and M. Silva. "Stochastic Continuous Petri Nets: An Approximation of Markovian Net Models". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part A* 42.3 (2012), pp. 641–653.

[Váz+10] C. Renato Vázquez et al. "Hybrid Petri Net Model of a Traffic Intersection in an Urban Network". In: *Proceedings of the IEEE International Conference on Control Applications, CCA 2010*. Yokohama, Japan, 2010, pp. 658–664.

[Wei00] K. Weihrauch. *Computable Analysis. An Introduction*. Springer, 2000.

[YLL09] H. Yang, C. Lin, and Q. Li. "Hybrid simulation of biochemical systems using hybrid adaptive Petri nets". In: *VALUETOOLS '09: International ICST Conference on Performance Evaluation Methodologies and Tools*. 2009.

[ZA90] N. Zerhouni and H. Alla. "Dynamic Analysis of Manufacturing Systems using Continuous Petri Nets." In: *Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA;* vol. 2. Los Alamitos, CA, USA, 1990, pp. 1070–1075.

[ZS10] M. Zarnay and M. Silva. "From coloured Petri nets to continuous P/T net models". In: *Technical report, DISC: Distributed Supervisory Control of Large Plants* (2010).