University "Alexandru Ioan Cuza" of Iaşi
Faculty of Computer Science

T E C H N I C A L R E P O R T

Time Constraints in Workflow Net Theory

Geanina Ionela Macovei

TR 09-03, December 2009

ISSN 1224-9327

Universitatea "Alexandru Ioan Cuza" Iaşi
Facultatea de Informatică
Str. Berthelot 16, 6600-Iaşi, Romania
Tel. +40-32-201090, email: bibl@infoiasi.ro

Preface

The aim of the workflow management is to help business goals to be achieved with high efficiency by means of sequencing work activities and invoking appropriate resources associated to these activities. Therefore, workflow management is both a key technology in supporting business processes and an effective means to realize full or partial automation of a business process. Despite many workflow management systems developed for different types of workflow processes, the lack of rigorous theoretic foundation and effective model verification and analysis methods in current workflow management systems is considered a serious problem.

Correctness verification plays an important role in implementing successful workflow management. The objective of correctness analysis of process control is to avoid the deadloks or structural conflicts in the execution of a workflow model. Formal semantics, local state-based system description, and abundant analysis techniques are good reasons to use Petri net based workflow management systems.

Time management in workflow processes is crucial in determining and controlling the life cycle of business activities. Time management involves the assignment of dead-lines, checking of timing inconsistencies, or the calculation of the overall process duration.

In this thesis there are discussed two main time extensions of workflow nets. Four types of timed workflow nets are introduced, L, E, S, and A, each of them based on the corresponding type of timed Petri net. The soundness property is studied for each type of them. In case of L- and E-timed workflow nets, the soundness property is equivalent with boundedness and liveness properties of their closures, which are L- and E-timed Petri nets, respectively. Boundedness and liveness properties for L-timed Petri nets can be reduced to the same properties for classical Petri nets. Petri nets with zero tests on bounded places are introduced and boundedness and liveness of E-timed Petri nets are reduced to the same properties of Petri nets with zero tests on bounded places. The soundness property for S- and A-timed workflow nets is studied through the halting problem of deterministic counter machines: deterministic counter machine can be simulated by S- and A-timed workflow nets and the halting problem of deterministic counter machines can be reduce to the quasi-liveness property of S- and A-timed workflow nets. As a conclusion, our thesis closes the decidability status of the soundness problem for timed workflow nets with discrete time durations based on L-, E-, S-, and A-timed Petri nets [56, 22].

I would like to thank all the persons who assisted me with my thesis: my supervisor, Prof. Dr. Ferucio Laurenţiu Ţiplea, for his confidence in me and for helping me to prepare this thesis; my former supervisor, Prof. Dr. Dumitru Todoroi, and Prof. Dr. Toader Jucan for their encouraging words; my friend Aurora Ţiplea for her support; and all the committee members for carefully reading my thesis.

Iaşi, October 15, 2007
Geanina Ionela Macovei

Introduction

A *workflow management system* is a system that completely defines, manages, and executes workflows through the execution of software whose order of execution is driven by a computer representation of the workflow logic [59]. Workflows are *cased-based* and the goal of workflow management is to handle cases as efficiently as possible. Cases are handled by executing *tasks* in a specific order. To do that, and to model states between tasks as well, *conditions* are added. A condition may hold or not. Each task has *pre-conditions* and *post-conditions*. Pre-conditions should hold before the task is executed, and post-conditions should hold after the task is executed.

Petri nets model workflow systems in a rather straightforward manner: tasks are modeled by transitions, conditions are modeled by places, and cases are modeled by tokens. Moreover, Petri nets have a well-defined semantics, provide a graphical language, are expressive, and many analysis techniques are now available for them. All these facts have led to an increasing interest in using Petri nets for workflow modeling [2].

For many real workflow processes temporal restrictions are essential. For instance, if a process does not take place on a given time interval then its execution on a further time can be insignificant. Therefore, incorporating time in workflow systems is a necessity.

Our thesis focuses on the two main time extensions of workfow nets: time workflow nets and timed worflow nets. For types of timed workflow nets behaviors (L, E, A, and S) are introduced and the soundness property is studied for each of them.

The thesis is structured as follows. Chapter 1 recalls the main notation and terminology on classical Petri nets which will be used throughout the thesis. Since many text-book on Petri nets are available we did not go into details. The main decision problems for classical Petri net are mentioned and two extensions of Petri nets, conditional Petri nets and Petri nets with zero tests, are presented. In the end, a short review of deterministic counter machines is made.

Chapter 2 begins with a short argumentation of the necessity of introducing time specifications in Petri net theory. Then we survey two classes of such time specifications: time Petri nets and timed Petri nets. Different variants of firing rules for time Petri nets are presented and their reachability, boundedness, and liveness properties are studied. A review of the firing strategies considered for timed Petri nets is made. We introduce slight modified firing rules for the four types of timed Petri nets with discrete durations (L, E, A, and S) that have been defined in [56] and [22]. We prove that reachability, boundedness, and liveness properties are decidable for LT_dPN, LT_dPN with auto-concurrency, and ET_dPN. In order to do that we show that this properties for LT_dPN and LT_dPN with auto-concurrency can be reduced to the same properties for classical Petri nets. In the case of ET_dPN, we introduce Petri nets with zero-tests on bounded places and we show that reachability, boundedness, and liveness properties for ET_dPN can be reduce to same properties for them. We also prove that reachability, coverability, boundedness, and quasi-liveness are all undecidable for ST_dPN and AT_dPN by reducing the halting problem for deterministic counter machines to each of these decision problems.

Chapter 3 presents the basic concepts related to classical workflow nets. One of the most important correctness criterion for workflow nets, the soundness property, is studied thoroughly. The two major approaches used to decide the soundness property for workflow nets, reducing the soundness problem to the liveness and boundedness of Petri nets and reducing the soundness problem to the "home marking" problem, are discussed. The complexity of the algorithms developed for deciding liveness and boundedness of Petri nets is given.

Chapter 4 deals with time extensions of workflow nets which are build on the skeleton of time extensions for Petri nets presented in Chapter 2. Four types of timed workflow nets with discrete duration based on L-, E-, S-, and A-timed Petri nets [56, 22] are introduced. This chapter will focus on studying the soundness property for time workflow nets and for the four types of timed workflow nets we introduced [53, 54, 55]. We show that the soundness property of time wokflow nets can be characterized in the same manner as for classical workflow nets and we shall present two classes of time workflow nets whose soundness is equivalent with the soundness of their underlying nets, which are classical workflow nets. We also prove that the soundness of L- and E-timed workflow nets can be reduce to the boundedness and liveness properties for L- and E-timed Petri nets. Since these last properties are decidable for L- and E-timed Petri nets, the soundness of L- and E-timed workflow nets is decidable. Another result is that soundness is undecidable for type S- and A-timed workflow nets. We prove it by transforming the type S- and A-timed Petri nets associated to a deterministic counter machine into type S- and A-timed workflow nets, respectively, and showing that the reachability, coverability, boundedness, and quasi-liveness are all undecidable for them as well.

The thesis is self-contained and unitary. The chapters follow a natural order to achieve our goal of studying the soundness property for time and timed workflow nets.

Contents

Chapter 1

Petri Net Theory

In this chapter we establish basic terminology, notation, and results in Petri net theory that will be used in our thesis. For further details, the reader is referred to [43, 35, 24].

1.1 Petri Nets

The set of *nonnegative integers* or *natural numbers* (*nonnegative rational numbers*, *nonnegative real numbers*, respectively) is denoted by $\mathbf{N}$ ($\mathbf{Q}_0^+$, $\mathbf{R}_0^+$, respectively) and the set of *positive integers* (*positive rational numbers*, *positive real numbers*, respectively) is denoted by $\mathbf{N}^+$ ($\mathbf{Q}^+$, $\mathbf{R}^+$, respectively). We denote by $\overline{\mathbf{N}}$ the set $\mathbf{N} \cup \{\omega\}$, where ω is a symbol which does not belong to $\mathbf{N}$. We extend the relation $<$ and the operations $+$ and $-$ to the set $\overline{\mathbf{N}}$ as follows:

- $n < \omega$, for all $n \in \mathbf{N}$;
- $\omega + n = n + \omega = \omega$, for all $n \in \mathbf{N} \cup \{\omega\}$;
- $\omega - n = \omega$, for all $n \in \mathbf{N}$.

Set inclusion is denoted by $\subseteq$ and *strict inclusion* by $\subset$. $|A|$ stands for the cardinality of the set A.

Given a function $\rho : A \to B$, its restriction to $C \subseteq A$ is denoted by $\rho|_C$.

If ρ is a function from a set A into $\mathbf{N}$ and $r \in \mathbf{N}$, then $\rho - r$ denotes the function $(\rho - r)(a) = \rho(a) - r$, for all $a \in A$, and $\rho \dot{-} r$ denotes the function $(\rho \dot{-} r)(a) = max\{\rho(a) - r, 0\}$, for all $a \in A$.

For a matrix I we denote by I^t the *transpose* of I. We denote by $\mathbf{0}$ the vector whose components are all 0.

A *multiset over a set* A is a function f from A into $\mathbf{N}$. The number $f(a)$ is called the *number of occurrences of* a *into* f. f is called *finite* if $f(a) = 0$ for all but finitely many $a \in A$. The *cardinality* of a finite multiset f over A, denoted $|f|$, is $\sum_{a \in A} f(a)$. It is usually written $a \in f$ if $f(a) > 0$. The basic operations with sets are defined for multisets too. For instance, if f is a multiset over A and $a \in A$, then $f \cup \{a\}$ is the multiset $(f \cup \{a\})(a) = f(a)+1$ and $(f \cup \{a\})(a') = f(a')$ for any $a' \neq a$.

For an alphabet V (i.e., a nonempty finite set), V^* denotes the free monoid generated by V under the operation of concatenation and λ denotes the unity of V^*. The elements of V^* are called *words* over V. λ is called the *empty word*. $|w|$ denotes the length of w. A *language* over V is any subset of V^*.

A *directed graph* is a pair (V, E), where V is a nonempty and finite set of *nodes* and E is a set of ordered pairs of nodes from V called *arcs*.

A *tree* is a directed graph for which there is a distinct node $v_0 \in V$, called the *root* of the tree, such that for every node $v \in V$ there exists a unique path from v_0 to v. If $v \in V$ we denote by v^+ the set of direct successors of v.

Let A and B be two nonempty sets. An (A, B)*-labeled tree* is a 4-tuple $\mathcal{A} = (V, E, l_1, l_2)$, where (V, E) is a tree, $l_1 : V \to A$ is the labeling function of the nodes, and $l_2 : E \to B$ is the labeling function of the arcs. By $d_{\mathcal{A}}(v, v')$ we denote the path (unique, if it exists) from v to v' in $\mathcal{A}$. We write $v'' \in d_{\mathcal{A}}(v, v')$ if v'' is a node on the path from v to v'.

A (finite) *Petri net* is a 4-tuple $\Sigma = (S, T, F, W)$, where:

- S and T are two finite nonempty sets (of *places* and *transitions*, respectively) such that $S \cap T = \emptyset$;
- $F \subseteq (S \times T) \cup (T \times S)$ is the *flow relation*;
- $W : (S \times T) \cup (T \times S) \to \mathbf{N}$ is the *weight function* of Σ verifying $W(x, y) = 0$ iff $(x, y) \notin F$.

When $W(x, y) \leq 1$ for all $(x, y) \in F$, we will simplify the 4-tuple (S, T, F, W) to the 3-tuple (S, T, F).

For each $x \in S \cup T$ we denote:

- ${}^\bullet x = \{y \mid (y, x) \in F\}$ and call it the *pre-set* of x, and

- $x^\bullet = \{y \mid (x,y) \in F\}$ and call it the *post-set* of x.

These notations are extended to subsets $X \subseteq S \cup T$ by union.

For a transition t we denote by Δt the function given by $\Delta t(s) = W(t,s) - W(s,t)$, for all places $s \in S$.

$T^{\pm}$ stands for the set $\{t^+|t \in T\} \cup \{t^-|t \in T\}$.

Two transitions t_1 and t_2 are *in conflict* if ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$.

A *marking* of Σ is any function M from S into $\mathbf{N}$, usually denoted as an S-indexed vector $M \in \mathbf{N}^{|S|}$. As the operations with markings are operations with functions or vectors, they are componentwise defined. Given two markings M_1 and M_2, we write $M_1 \leq M_2$ whenever $M_1(s) \leq M_2(s)$, for any place s. Moreover, if $M_1(s) < M_2(s)$ for some place s, then we write $M_1 < M_2$.

For a place s of Σ we denote by M_s the marking given by $M_s(s) = 1$ and $M_s(s') = 0$, for all $s' \neq s$.

A *marked Petri net* is a pair $\gamma = (\Sigma, M_0)$, where Σ is a Petri net and M_0, *the initial marking of* γ, is a marking of Σ.

Pictorially, Petri nets are represented as follows: places are represented by circles, transitions by boxes, the flow relation is represented by drawing an arc between x and y whenever (x,y) is in the relation, and the weight function labels the arcs whenever their weights are greater than 1. A marking M of a Petri net is represented by drawing $M(s)$ black tokens into circle representing the place s, for each place s.

The *firing rule* of a Petri net Σ states that a transition t is *enabled* at a marking M (or M *enables* t), denoted by $M[t\rangle_\Sigma$, if $M(s) \geq W(s,t)$, for all $s \in S$. Given $T' \subseteq T$, $T'(M)$ stands for the set of all transitions in T' which are enabled at the marking M. If t is enabled at M, then it can *fire* yielding a new marking M' given by $M'(s) = M(s) - W(s,t) + W(t,s)$, for all $s \in S$; we denote this by $M[t\rangle_\Sigma M'$.

The firing rule is extended usually to sequences of transitions by:

- $M[\lambda\rangle M$, for any marking M;
- $M[wt\rangle M'$ whenever there is a marking M'' such that $M[w\rangle M''$ and $M''[t\rangle M'$, where M and M' are markings, $w \in T^*$, and $t \in T$.

When there is a sequence $w \in T^*$ such that $M[w\rangle_\Sigma M'$ we say that M' is *reachable* (from M) in Σ and w is a *firing sequence* (from M) in Σ. $[M\rangle_\Sigma$ stands for the set of all reachable markings (from M) in Σ.

The notation $[\cdot\rangle_x$ will be simplified to $[\cdot\rangle$ whenever no confusion may arise.

Let Σ be a Petri net and $x_i, x_k \in S \cup T$. A *path* from x_1 to x_k is a sequence $x_1, x_2, \ldots, x_k$ such that $(x_i, x_{i+1}) \in F$, for all $1 \leq i \leq k-1$. The path $x_1, x_2, \ldots, x_k$ is called *elementary* if $x_i \neq x_j$ whenever $i \neq j$. The path $x_1, x_2, \ldots, x_k$ is called a *circuit* if $x_1 = x_k$.

For a path $c = x_1, x_2, \ldots, x_k$ we denote by $\alpha(c) = \{x_1, x_2, \ldots, x_k\}$.

A Petri net is called *weakly connected* (or just *connected*) if every pair of nodes x and y satisfies $(x, y) \in (F \cup F^{-1})^*$. It is called *strongly connected* if every pair of nodes x and y satisfies $(x, y) \in F^*$.

A Petri net Σ is called an *S-graph* if for all $t \in T$ it holds $|{}^\bullet t| = |t^\bullet| = 1$.

A net $\Sigma' = (S', T', F', W')$ is a *subnet* of Σ if $S' \subseteq S$, $T' \subseteq T$, $F' = F \cap ((S' \times T')(T' \times S'))$, and $W' = W|_{(S' \cup T') \times (T' \cup S')}$. Σ' is a *partial subnet* of Σ if $S' \subseteq S$, $T' \subseteq T$, $F' \subseteq F \cap ((S' \times T')(T' \times S'))$, and $W' = W|_{F'}$.

A subnet Σ' of Σ is an *S-component* of Σ if Σ' is a strongly connected S-graph and $T' = {}^\bullet S' \cup S'^\bullet$.

A Petri net $\Sigma = (S, T, F, W)$ is *covered by S-components* if there exists a set of S-components $\Sigma_i = (S_i, T_i, F_i, W_i)$, $i = 1, \ldots, n$, of Σ such that $S = \bigcup_{i=1,\ldots,n} S_i$, $T = \bigcup_{i=1,\ldots,n} T_i$, and $F = \bigcup_{i=1,\ldots,n} F_i$.

Assume that the sets S and T of places and transitions of a Petri net are $S = \{s_1, \cdots, s_m\}$ and $T = \{t_1, \cdots, t_n\}$. Moreover, we assume that these two sets are totally ordered by:

- $s_1 < \cdots < s_m$;
- $t_1 < \cdots < t_n$.

The *incidence matrix* of Σ is an $m \times n$ matrix I_Σ defined by:

$$I_\Sigma(i, j) = W(t_j, s_i) - W(s_i, t_j), \text{ for all } 1 \leq i \leq m \text{ and } 1 \leq j \leq n.$$

We denote by $Rank(I_\Sigma)$ the rank of the incidence matrix of Σ.

An m-dimension vector J is called an *S-invariant* of Σ if it satisfies:

$$J^t \cdot I_\Sigma = \mathbf{0}.$$

An n-dimension vector J is called a *T-invariant* of Σ if it satisfies:

$$I_\Sigma \cdot J = \mathbf{0}.$$

Let $\gamma = (\Sigma, M_0)$ be a marked Petri net. A $(\overline{\mathbf{N}}^{|S|}, T)$-labeled tree $\mathcal{T}(\gamma) = (V, E, l_1, l_2)$ is called the *coverability tree* of γ if it satisfies the following properties:

1. $l_1(v_0) = M_0$, where v_0 is the root of $\mathcal{T}$;

2. for every node $v \in V$ we have:

   - $|v^+| = 0$, if $T(l_1(v)) = \emptyset$ or there exists a node $v' \in d_{\mathcal{T}}(v_0, v)$, $v' \neq v$, such that $l_1(v) = l_1(v')$;
   - $|v^+| = |T(l_1(v))|$, otherwise;

3. for every node $v \in V$ with $|v^+| > 0$ and every $t \in T(l_1(v))$ there exists a node $v' \in V$ such that:

   (a) $(v, v') \in E$;

   (b) for every $s \in S$ we have:

   - $l_1(v')(s) = \omega$, if there exists $v'' \in d_{\mathcal{T}}(v_0, v)$ such that $l_1(v'')(s') \leq l_1(v)(s') + W(t, s') - W(s', t)$, for all $s' \neq s$, and $l_1(v'')(s) < l_1(v)(s) + W(t, s) - W(s, t)$;
   - $l_1(v')(s) = l_1(v)(s) + W(t, s) - W(s, t)$;

   (c) $l_2(v, v') = t$.

1.2 Basic Decision Problems for Petri Nets

The analysis of Petri nets as concurrent system models lead to the necessity of solving some decision problems. In the sequel, we shall recall the basic decision problems concerning Petri nets [15].

The *reachability* problem for marked Petri nets is to decide whether a given marking M of a marked Petri net $\gamma = (\Sigma, M_0)$ is reachable form M_0.

The reachability problem is decidable for marked Petri nets [31, 27].

For a marking M of γ and $S' \subset S$, $M|_{S'}$ is a *submarking* of M.

The *submarking reachability* problem for marked Petri nets is to decide whether a given submarking of a marking M of a marked Petri net $\gamma = (\Sigma, M_0)$ is reachable from M_0.

The submarking reachability problem is recursively equivalent to the reachability problem [15], therefore it is decidable for marked Petri nets.

We say that a marking M is *coverable* in a marked Petri net $\gamma = (\Sigma, M_0)$ if there exists a marking $M' \in [M_0\rangle$ such that $M \leq M'$.

The *coverability* problem for marked Petri nets is to decide whether a given marking M of a marked Petri net γ is coverable.

The coverability problem is decidable for marked Petri nets [25].

A Petri net γ is *bounded with respect to a marking* M if there exists a natural number n such that $M'(s) \leq n$, for every markings M' reachable from M and every place s. If $n = 1$ then γ is called *safe* with respect to M.

The *boundedness* problem for marked Petri nets is to decide whether a given marked Petri net $\gamma = (\Sigma, M_0)$ is bounded (with respect to the initial marking M_0).

The boundedness problem is decidable for marked Petri nets [25].

A Petri net γ is called *live with respect to a marking* M if for every marking M' reachable from M and every transition t, there exists a marking M'' reachable from M' such that $M''[t\rangle$.

The *liveness* problem for marked Petri nets is to decide whether a given marked Petri net $\gamma = (\Sigma, M_0)$ is live (with respect to the initial marking M_0).

Hack showed in [15] that the liveness problem is recursively equivalent to the reachability problem.

1.3 Conditional Petri Nets

Many real systems can not be properly modeled by classical Petri nets. This fact led to considering extensions of Petri nets in order to better fit real necessities. In [52], $\mathcal{L}$-conditional Petri nets were introduced. These nets are obtained by imposing a restriction on the firing rule.

Let $\mathcal{L}$ be an arbitrary family of languages. An $\mathcal{L}$-*conditional Petri net* is a pair $\gamma = (\Sigma, \varphi)$, where:

- Σ is a Petri net;
- φ, the $\mathcal{L}$-*conditioning function* of γ, is a function from T into $\mathcal{P}(T^*) \cap \mathcal{L}$.

A *marked conditional Petri net* is a 3-tuple $\gamma = (\Sigma, \varphi, M_0)$, where (Σ, φ) is a conditional Petri net and M_0 is a marking of Σ called the *initial marking* of γ.

The *firing rule* of a conditional net γ states that if M is a marking of γ and $u \in T^*$, the transition t is *enabled* at (M, u), denoted by $(M, u)[t\rangle_{\gamma,c}$, if $W(s, t) \leq M(s)$, for any place s, and $u \in \varphi(t)$. If $(M, u)[t\rangle_{\gamma,c}$, then t can *fire* yielding a pair (M', v), denoted by $(M, u)[t\rangle_{\gamma,c}(M', v)$, where $M[t\rangle_\Sigma M'$ and $v = ut$.

As for Petri nets, the transition rule can be be extended to sequences of transitions.

Reachability and *liveness* are similarly defined for $\mathcal{L}$-conditional Petri nets as for ordinary Petri nets, and these properties are decidable for $\mathcal{L}_3$ conditional Petri nets ($\mathcal{L}_3$ stands for the family of regular languages) [51].

1.4 Petri Nets with Zero Tests

Another extension of Petri nets that we shall use in this thesis is that of a Petri net with zero-tests.

A *Petri net with zero-tests* (PN_{0t}), also called an *inhibitor Petri Net* [4], is a pair $\gamma = (\Sigma, J)$, where Σ is a Petri net and J is a subset of $S \times T$ such that $F \cap J = \emptyset$.

A transition t is *enabled* at a marking M in γ, denoted by $M[t\rangle_\gamma$, if $M[t\rangle_\Sigma$ and $M(s) = 0$, for all $s \in pr_1(J)$ $(pr_1(J) = \{s \mid \exists t \in T : (s, t) \in J\})$. If t is enabled at M in γ, then it can *fire* yielding a new marking M' given by $M'(s) = M(s) - W(s, t) + W(t, s)$, for all $s \in S$; we denote this by $M[t\rangle_\gamma M'$.

Pictorially, a pair $(s, t) \in J$ of a Petri net with zero-tests is represented by an arc as in Figure 1.1.

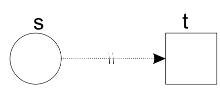


Figure 1.1: Representation of a pair $(s, t) \in J$

A *marked* PN_{0t} (mPN_{0t}) is a 3-tuple $\gamma = (\Sigma, J, M_0)$, where (Σ, J) is a PN_{0t} and M_0 is a marking of Σ called the *initial marking* of γ.

1.5 Counter Machines

A *deterministic counter machine* (*DCM*) is a 6-tuple $A = (Q, q_0, q_f, C, x_0, I)$, where:

1. Q is a finite nonempty set of *states*, $q_0 \in Q$ is the *initial state*, and $q_f \in Q$ is the *final state*;

2. C is a finite nonempty set of *counters*. Each counter can store any natural number, and $x_0 : C \to \mathbf{N}$ is the initial content of the counters;

3. I is a finite set of *instructions*. For each non-final state there is exactly an instruction that can be executed in that state; for q_f there is no instruction. An instruction for a state q is of the one of the following forms:

   - *increment instruction* - $I(q, c, q')$

     q : begin
     $\quad\quad c := c + 1$;
     $\quad\quad$ go to q'
     $\quad$ end

   - *test instruction* - $I(q, c, q', q'')$

     q : if $c = 0$ then go to q'
     $\quad\quad\quad\quad$ else begin
     $\quad\quad\quad\quad\quad\quad c := c - 1$;
     $\quad\quad\quad\quad\quad\quad$ go to q''
     $\quad\quad\quad\quad\quad$ end

Let $A = (Q, q_0, q_f, C, x_0, I)$ be a DCM. A *configuration* of A is a pair (q, x), where $q \in Q$ and $x : C \to \mathbf{N}$. A configuration (q, x) is called *initial* when $q = q_0$ and $x = x_0$; (q, x) is called *final* when $q = q_f$.

Let $A = (Q, q_0, q_f, C, x_0, I)$ be a DCM. We define the binary relation $\vdash_A$ on the configurations of A by:

$$(q, x) \vdash_A (q', x') \quad \text{if one of the following holds:}$$

1. there is an increment instruction $I(q, c, q')$ such that $x'(c) = x(c) + 1$ and $x'(c') = x(c')$, for all $c' \in C - \{c\}$;

2. there is a test instruction $I(q, c, q_1, q_2)$ such that
   (a) if $x(c) = 0$, then $q' = q_1$ and $x' = x$;
   (b) if $x(c) \neq 0$, then $q' = q_2$, $x'(c) = x(c) - 1$ and $x'(c') = x(c')$, for all $c' \in C - \{c\}$.

$\vdash^{+}_{A}(\vdash^{*}_{A})$ stands for the transitive (reflexive and transitive) closer of $\vdash_A$. If $(q_0, x_0) \vdash^{*}_{A} (q, x)$, then (q, x) is called *reachable*.

The *halting problem* for deterministic counter machines is to decide whether a given DCM reaches a final configuration.

It is well-known that this problem is undecidable [34].

Chapter 2

Time Constraints in Petri Net Theory

The concept of time is neither explicit nor implicit in the original definition of Petri nets because associating timing constraints to the activities represented in Petri nets may prevent certain transitions from firing, destroying the assumption that all possible behaviors of a real system are represented by the structure of the Petri nets.

In [35] it was remarked that

> *"The addition of timing information might provide a powerful new feature for Petri nets, but may not be possible in a manner consistent with the basic philosophy of PNs".*

Indeed, when dealing with the fundamental properties of Petri net models and systems, with their analysis techniques and the associated computational complexity, and with the equivalence between Petri nets and other models of parallel computation, timing is not relevant.

In the last years, Petri nets were recognized as possible models of real concurrent systems, capable of coping with all aspects of parallelism and conflict in asynchronous activities with multiple actors. In this case, timing is not important when considering only the logical relationships between the entities that are part of the real system. The concept of time becomes instead of paramount importance when the interest is driven by real applications whose efficiency is always a relevant design goal. In areas like hardware and computer architecture design, communication protocols, and software system analysis, timing is crucial even to define the logical aspects of the dynamic operations.

When introducing time into Petri net models and systems, it would be extremely useful not to modify the basic behavior of the underlying untimed model. By so doing, in the study of timed Petri nets it is possible to exploit the properties of the basic untimed model as well as the available theoretical results. The addition of temporal specifications therefore must not modify the unique and original way of expressing synchronization and parallelism that is peculiar to Petri nets.

Different ways of incorporating timing information into Petri nets were proposed by many researchers during the last two decades; the different proposals are strongly influenced by the specific application fields. Nowadays, two main directions in using the time are recognized. One of them, timed Petri nets, considers the time as a duration [41, 40, 17] and are used for performance evaluation of the systems. The other one, time Petri nets, considers time intervals [32, 33, 6] and are suitable to design and analyse communication protocols and study problems with recoverability aspects. Other extensions, as timing constraints Petri nets [50, 28], are inspired by both, timed and time Petri nets.

2.1 Time Petri Nets

In 1974 P.M. Merlin has introduced time Petri nets [32] to study the recoverability of computer systems. Later, in [33], time Petri nets have been used to design and analyze several practical computer communication protocols.

Merlin's time Petri nets extend classical Petri nets by associating a time interval $(I_1(t), I_2(t))$ to each transition t.

Definition 2.1 A *time Petri net* (TPN) is a tuple $\gamma = (\Sigma, I)$, where:

1. $\Sigma = (S, T, F, W, M_0)$ is a marked Petri net, called the *the underlying net of* γ;

2. $I : T \rightarrow \mathbf{R}^+ \times \mathbf{R}^+$ is the *time function* of γ, where for each $t \in T$, $I(t) = (I_1(t), I_2(t))$ with $I_1(t) < I_2(t)$.

$I(t)$ is called the *static interval* of t. $I_1(t)$ and $I_2(t)$ are called *static earliest firing time* of t ($seft(t)$) and *static latest firing time* of t ($slft(t)$), respectively, and are relative to the moment at which t was last enabled.

In order to fire, a transition of a time Petri net must be continuously enabled from the moment it became enabled until the moment it fires. When a transition t fires, the transitions which are disabled are those not enabled after t removes tokens from the corresponding input places and before t adds tokens to the corresponding output places. They include the transition t. When an enabled transition is disabled by the firing of another one, its enabling is interrupted. It is considered a newly enabled transition if it is enabled again after the transition that fired adds tokens to the corresponding output places. If a transition is continuously enabled for a period of time equal to its static latest firing time, then it is forced to fire. Firing a transition takes no time.

The firing rule in [33] is defined as follows:

1. a transition t *may fire* if the input places of t holds enough tokens for a period of time equal or greater then $I_1(t)$. The firing of t removes the corresponding tokens from each input place and adds the corresponding tokens to each output place;

2. if the input places of the transitions t holds enough tokens for a period of time equal to $I_2(t)$, then t *fires*.

Formal definitions of the firing rule have been introduced later, in two distinct ways, and we shall discuss them further.

Example 2.1 Timing values like time-outs used to implement recovery from losses of messages are very important in specifying communication protocols. Time Petri nets are suitable approaches for verifying that these time-outs are correctly set. The alternating bit protocol is an example of a process modeled by a TPN.

This protocol transmits messages between two entities, allowing only messages in transit at a time, over an unreliable transmission medium. Hypotheses on the behavior of the transmission medium are that the messages or acknowledgments may be lost in transit. Recovery form losses is done using a time-out and retransmitting: each sender records the time at which it sends a message and if an acknowledgment of its delivery does not return within a given time, the message is retransmitted.

The selected mechanism must be sufficient for recovering from losses and for preventing the acceptance of duplicate messages: upon reception of a message, the receiver must be able to decide whether this is a new message or a duplicate. This problem is solved by numbering the messages, prior to transmission, with modulo-2 sequence numbers and, for every packet received, an acknowledgment is sent that carries the sequence number of the received packet.

Figure 2.1 shows the TPN that model this protocol. Losses of messages and acknowledgments are represented as transitions with no output places. Estimates for duration of all elementary actions of the protocol are provided. Thus, retransmission of messages occur at a time comprised between 5 and 6 time units after the message has been sent. The interval estimated for losses and receptions of messages and acknowledgments is $(0, 1)$ and no time constraint (the intervals $(0, \infty)$) are given for sending the numbered messages.

The meaning of the transitions are:

t_1 Send packet 0
t_2 Resent packet 0
t_3 Receive acknowledgment 0
t_4 Send packet 1
t_5 Resent packet 01
t_6 Receive acknowledgment 1
t_7 Receive and release packet 0
t_8 Send acknowledgment 0
t_9 Receive and reject packet 0
t_{10} Receive and release packet 1
t_{11} Send acknowledgment 1
t_{12} Receive and reject packet 1
t_{13} Lose packet 0
t_{14} Lose acknowledgment 0
t_{15} Lose packet 1
t_{16} Lose acknowledgment 1

According to the time constraints, packet 0 can be sent at any moment by firing transition t_1. In the time interval $(0, 1)$ after the message was sent this can be lost (transition t_{13} fires) or received to destination (transition t_7 fires). If the message was lost, the only transition that can be applied is t_2 which means that the sender will resend the message in at least 5 and at most 6 time units after the previous transmission. If the receiver gets the message, it will send an acknowledgment (transition t_8 fires) in at most two units of time after the message was received. As the message, the acknowledgment can be lost (transition t_{14} fires) or received (transition t_3 fires) by the sender. If it

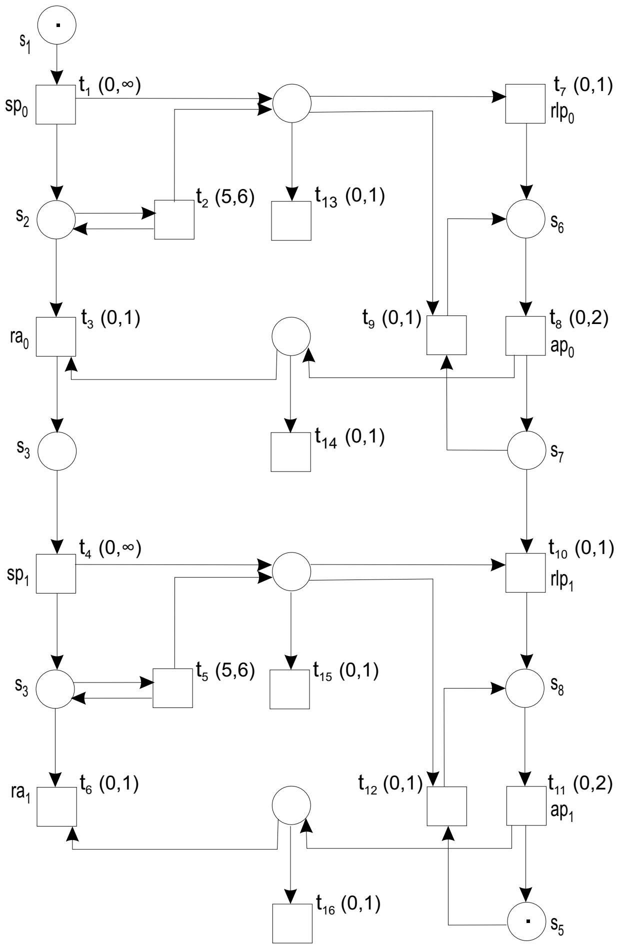


Figure 2.1: The alternating bit protocol

was lost, the sender will resend the packet between the interval 5 and 6 units of time after the packet 0 was last sent. If the acknowledgment is received, the packet 1 can be sent (transition t_4 can fire). The acknowledgment is received in at most 3 time units after the corresponding message was sent preventing the sender to retransmit the same packet again.

Time Petri nets are used in [6] in order to model the behavior and analyse the properties of time systems. A formal definition of the firing rule was adopted. The definition of a time Petri net slightly differs from the one in [33] in sense that, from technical reasons, the static earliest and the static latest firing time of the transitions are nonnegative rational numbers. Thus, the time function of γ is defined as $I : T \to \mathbf{Q}_0^+ \times (\mathbf{Q}_0^+ \cup \{\infty\})$ satisfying:

$$I(t) = (I_1(t), I_2(t))$$

where $I_1(t)$ and $I_2(t)$ are rationals such that:

- $0 \leq I_1(t) < \infty$;
- $0 \leq I_2(t) \leq \infty$;
- $I_1(t) \leq I_2(t)$ if $I_2(t) \neq \infty$.

We remark that $I_1(t) < I_2(t)$ if $I_2(t) = \infty$.

In practice, considering rational numbers for static intervals does not limit the modeling and the analysis power of the systems.

For simplicity, in [6] it is supposed that the TPNs are such that none of the transitions may become enabled more then once "simultaneously". Therefore, for any marking M and any transition t, it holds that there exists a place s such that $M(s) < 2 \cdot W(s, t)$.

Definition 2.2 Let $\gamma = (\Sigma, I)$ be a time Petri net. *A state* of γ is a pair (M, I) consisting of:

1. a marking M of Σ;
2. a vector I of possible firing times. Each entry in I is a firing interval and the number of entries in I is given by the number of transitions enabled at the marking M.

I will vary during the behavior of the net according to the number of transitions enabled at the current marking. The entry in I corresponding to a transition t defines the minimum and the maximum time between which the transition t is individually allowed to fire. The bounds of this entry are called the *dynamic earliest firing time* of t ($eft(t)$) and the *dynamic latest firing time* of t ($lft(t)$).

Definition 2.3 Let $\gamma = (\Sigma, I)$ be a time Petri net. The state (M_0, I_0), where M_0 is the initial marking of Σ and I_0 contains the static firing intervals of the transitions enabled at M_0 in an arbitrary but fixed order, is called the *initial state* of γ.

In [6] a global watch is used to keep the time τ elapsed since the first transition became enabled in the TPN.

Definition 2.4 A transition t is *firable* from the state (M, I) at time $\theta' = \tau + \theta$, denoted by $(M, I)[(t, \theta')\rangle_\gamma$, if the following conditions hold:

1. t is enabled in γ at the marking M, at time τ:

2. the *relative firing time* θ (relative to the *absolute enabling time* τ) is not smaller than $eft(t)$ and not greater than the smallest of the lft's of all transitions enabled at marking M:

$$eft(t) \leq \theta \leq min\{lft(t') \mid M[t'\rangle_\gamma\}$$

Condition 2 in Definition 2.4 should hold because the firing of t should not disable any transition currently enabled at M by passing their dynamic lft.

Definition 2.5 Let γ be a TPN. If a transition t is firable from the state (M, I) at time $\theta' = \tau + \theta$, then the state (M', I') reached from (M, I) by *firing* t at the relative time θ, denoted by $(M, I)[(t, \theta')\rangle_\gamma(M', I')$, is computed as follows:

1. the marking M' is computed like in classical Petri nets:

$$M'(s) = M(s) - W(s, t) + W(t, s), \text{ for all } s \in S;$$

2. I' is computed in three steps:

   (a) remove from I the intervals that corresponds to the transition not enabled at the marking M';

   (b) for all transition t' enabled at the marking M and not in conflict with t (transition which remains enabled) the intervals in I' are:

$$(max(0, eft(t') - \theta), lft(t') - \theta);$$

   (c) for the newly enabled transitions at M' and those which are enabled at M', were already enabled at M and in conflict with t in M, the intervals in I' are set to the static firing intervals of the corresponding transitions.

Definition 2.6 Let γ be a TPN.

1. A *firing schedule* is a sequence of pairs
$$\sigma = (t_1, \theta_1)(t_2, \theta_2) \dots (t_n, \theta_n).$$

2. A firing schedule σ is *feasible* from a state (M, I), denoted $(M, I)[\sigma\rangle_\gamma$, if there exists the states $(M_1, I_1), (M_2, I_2), \dots, (M_n, I_n)$ such that:
$$(M, I)[(t_1, \theta_1)\rangle_\gamma(M_1, I_1)[(t_2, \theta_2)\rangle_\gamma(M_2, I_2) \dots (M_{n-1}, I_{n-1})$$
$$[(t_{n-1}, \theta_{n-1})\rangle_\gamma(M_n, I_n).$$

3. A state (M', I') is *reachable* from (M, I) if there exists a feasible firing schedule σ such that $(M, I)[\sigma\rangle_\gamma(M', I')$.

$[(M, I)\rangle_\gamma$ stands for the set of all reachable states in γ (from (M, I)).

Let us assume that a transition t, with firing interval $(eft(t), lft(t))$, is enabled at the current marking and time θ has elapsed since it was enabled. If a different transition t' is fired at time θ, the current firing interval for t becomes $(max(0, eft(t) - \theta), lft(t) - \theta)$. If the firing of t' makes t twice enabled, then t has two related firing intervals:

- $(max(0, eft(t) - \theta), lft(t) - \theta)$ for the first time it was enabled, and
- $(I_1(t), I_2(t))$ for the second time it was enabled.

There are several interpretations for multiple enabledness and the firing rule will depend on the chosen one. In a general interpretation, transitions enabled several times simultaneously can be considered as independent occurrences of the same transition. Therefore, the occurrence that is related to the oldest interval has to fire first. The restriction that none of the transitions may become enabled more then once simultaneously is considered because the multiple enabledness complicates the notation. Thus, notation like $t_{enabled(k)}$ for transition t multiple enabled k times should be used.

The set of states of a time Petri nets may be infinite because of the very high or infinite number of time values that can be selected to fire a transition at a given marking. State classes are proposed in [6] to define a finite enumerative analysis method for characterizing the behavior of a TPN.

Definition 2.7 Let γ be a TPN. A *state class* is a pair (M, D), where:

1. M is a marking of γ;

2. D, the *firing domain* of the class, is a set of firing domains.

The intuition is that a state class (M, D) defines a set of states, all of them having the same marking M. D is then the set of all firing domains of these states.

Usually, D is the solution set of a system of inequalities in which variables are associated to the transitions enabled by the marking M:

$$D = \{t \mid A \cdot t \geq b\}$$

where A is a matrix, b a vector of constants and t is a vector of variable depending on the enabled transitions.

State classes capture the idea of all possible firing times that may occur at a given marking.

In what follows, we assume that all transitions associated to the variables in t are ordered. $t(i)$ will refer to the ith transition enabled at M.

Definition 2.8 Let γ be a TPN and (M, D) a state class of it. A transition t_i is *firable* from the class (M, D), denoted $(M, D)[t_i\rangle_\gamma$, if the following conditions hold:

1. t_i is enabled at the marking M;

2. the variable $t(i)$ associated to the transition t_i must satisfy the system of inequalities:
$$\begin{cases} A \cdot t \geq b \\ t(i) \leq t(j), \quad \text{for all } j \neq i \end{cases}$$
where $t(j)$ is the variable associated to the jth transition enabled at M. The second inequality enforce that the firing of t_i must occur before the minimum of all lft of all enabled transitions at M.

Definition 2.9 Let γ be a TPN and (M, D) a state class of it such that

$$D = \{t \mid A \cdot t \geq b\}$$

The class (M', D') reached from class (M, D) by firing the transition t_f associated to the variable $t(f)$, denoted $(M, D)[t_f\rangle_\gamma(M, D')$, is computed as follows:

1. the marking M is computed by the classical rule:

$$M'(s) = M(s) - W(s, t_f) + W(t_f, s), \text{ for all } s \in S;$$

2. D' is computed in three steps:

   (a) add to the system that define domain D, the firability condition for the transition t_f, that is:

   $$A \cdot t \geq b$$
   $$t(f) \leq t(j), \text{ for all } j,\ j \neq f.$$

   All the times related to the variables $t(j)$, $j \neq f$, are expressed as the sum of the time of the fired transition t_f and the new variable $t''(j)$ with

   $$t(j) = t(f) + t''(j)$$

   The variable $t(f)$ is eliminated from the system. The resulting system may be written (see Lemma 2.1):

   $$A'' \cdot t'' \geq b''$$
   $$0 \leq t''$$

   where A'', b'' are computed from A, b, the equations that define the new variables, and eliminating the variables $t(f)$.

   (b) as for $t(f)$, all variables corresponding to the transitions disabled when t_f fires are eliminated from the system obtained after step (a) (these transitions are those enabled at M and not enabled at $M(\cdot) - W(\cdot, t_f)$).

   (c) add to the system obtained after step (b) the inequalities corresponding to the new variable associated to the newly enabled transitions (these transitions are those enabled at M, not enabled at $M(\cdot) - W(\cdot, t_f)$ and enabled at M'). These variables must belong to the static firing interval of the corresponding transitions.

   The new system of inequalities may be written as (see Lemma 2.1):

$$A' \cdot t' \geq b'.$$

This system has as many variables as there are transitions enabled at M' and its solution set defines D'.

Definition 2.10 Let γ be a TPN. γ is called *T-bounded* if there exists a natural number k such that none of its transitions may be enabled more than k times simultaneously by any reachable marking:

$$(\exists k \in \mathbf{N})(\forall M \in [(M_0, I_0)\rangle_\gamma)(\forall t \in T)(\exists s \in S)(M(s) < (k+1) \cdot W(s,t))$$

In the particular case that $k = 1$, γ is called *T-safe*.

Lemma 2.1 (*General form.*) The firing domain D of classes for any T-safe TPN can be expressed as solution sets of systems of inequalities of the following form:

$$\alpha_i \leq t(i) \leq \beta_i, \text{ for all } i$$
$$t(j) - t(k) \leq \gamma_{jk}, \text{ for all } j, k, k \neq j.$$

Proof We shall prove that this is true for the initial class and that this form is conserved by transition's firing.

The initial class is (M_0, D_0), where M_0 is the initial marking of γ and D_0 is defined as a set of inequalities of the form:

$$\alpha_i^s \leq t(i) \leq \beta_i^s,$$

for all variables $t(i)$ associated with the transition t_i enabled at the initial marking, where $\alpha_i^s = I_1(t_i)$, $\beta_i^s = I_2(t_i)$.

By definition, $\gamma_{jk} = \beta_j^s - \alpha_k^s$. These inequalities are redundant and do not affect the solution set of the system $\alpha_i^s \leq t(i) \leq \beta_i^s$.

So, the initial firing domain fulfills the general form.

Let us assume that in the state class (M, D), D is the solution of a system in general form and (M', D') is the state reached form (M, D) by the firing of the transition t_f. We shall show that the newly computed system has the general form, too.

We shall follow the three steps in which D' is computed.

(a) After the step (a) of Definition 2.9 we shall have:

- α_i becomes $max(0, -\gamma_{fi}, \alpha_i - \beta_f)$;
- β_i becomes $min(\gamma_{if}, \beta_i - \alpha_f)$;
- γ_{jk} becomes $min(\gamma_{jk}, \beta_j - \alpha_k)$.

By definition, $\alpha_i \geq 0$.

All inequalities containing variable $t(f)$ must disappear from the system. $t(f)$ appears in:

$$\alpha_f \leq t(f) \leq \beta_f \tag{2.1}$$

$$t(i) - t(f) \leq \gamma_{if} \tag{2.2}$$

$$t(f) - t(i) \leq \gamma_{fi} \tag{2.3}$$

and does not appear in

$$\alpha_i \leq t(i) \leq \beta_i \tag{2.4}$$

$$t(j) - t(k) \leq \gamma_{jk} \tag{2.5}$$

From 2.4 and $t(i) = t(f) + t''(i)$, it comes $\alpha_i - t(f) \leq t''(i) \leq \beta_i - t(f)$ and from 2.1 we have

$$\alpha_i - \beta_f \leq t''(i) \leq \beta_i - \alpha_f.$$

From 2.2 it comes $t''(i) + t(f) - t(f) \leq \gamma_{if}$.

From 2.3 we have $t(f) - t''(i) - t(f) \leq \gamma_{fi}$, giving $t''(i) \geq -\gamma_{fi}$.

From 2.5 we have $t''(j) + t(f) - t''(k) - t(f) \leq \gamma_{jk}$.

The new relationships between $t(j)$ and $t(k)$ come through $t(f)$ as:

$\alpha_j \leq t''(j) + t(f) \leq \beta_j$
$\alpha_j - t''(j) \leq t(f) \leq \beta_j - t''(j)$
and $\alpha_k - t''(k) \leq t(f) \leq \beta_k - t''(k)$
giving $t''(k) - t''(j) \leq \beta_k - \alpha_j$
and $t''(j) - t''(k) \leq \beta_j - \alpha_k$

(b) In the step (b) of Definition 2.9 we must successively eliminate from the system the variables associated with the transitions disabled when t_f fires. Each elimination, for instance of variable $t(e)$, corresponds to the following transformation of the system:

– α_i becomes $max(\alpha_i, \alpha_e - \gamma_{ei})$;
– β_i becomes $min(\beta_i, \beta_e + \gamma_{ie})$;
– γ_{jk} becomes $min(\gamma_{jk}, \gamma_{je} + \gamma_{ek})$.

$t(e)$ appears in:

$$\alpha_e \leq t(e) \leq \beta_e \tag{2.6}$$
$$t(e) - t(k) \leq \gamma_{ek} \tag{2.7}$$
$$t(j) - t(e) \leq \gamma_{je} \tag{2.8}$$

and does not appear in

$$\alpha_i \leq t(i) \leq \beta_i \tag{2.9}$$
$$t(j) - t(k) \leq \gamma_{jk} \tag{2.10}$$

The proof is identically to step (a). All inequalities containing variable $t(e)$ disappear from the system.

(c) This step corresponds to the introduction in the system of the variables and inequations relative to the newly enabled transitions. Each introduction of variable $t(n)$ yields the following transformation of the system:

– α_i, β_i, γ_{jk}, for all i,j, k distinct from n, do not change.
– The new variable $t(n)$, when is introduced, is constrained by:

$$\alpha_n^s \leq t(n) \leq \beta_n^s$$

where $\alpha_n^s = I_1(t_n)$, $\beta_n^s = I_2(t_n)$ and t_n is the transition associated to $t(n)$.
– The inequalities which establish the relations between $t(n)$ and the others are:

$$t(n) - t(k) \leq \gamma_{nk}, \text{ for all } k,\ k \neq n$$
$$t(j) - t(n) \leq \gamma_{jn}, \text{ for all } j,\ j \neq n.$$

We shall chose values for γ_{nk} and γ_{jn} such that these inequalities are redundant:

$$\gamma_{nk} = \beta_n^s - \alpha_k$$
$$\gamma_{jn} = \beta_j - \alpha_n^s.$$

□

Using the firing rule between classes, a tree can be built. Its root is the initial class and there is an arc labeled with t from class (M, D) to class (M', D') if t is firable from (M, D) and its firing yields (M', D').

In this tree each class have a finite numbers of successors, at most one for each transition enabled at the marking of the class.

Any sequence of transitions firable in γ will be a path in this tree of classes. The existence of a path labeled ω between two classes (M, D) and (M', D') of the tree is interpreted as follows: there exists two feasible firing schedule $(t_1, \theta_1)(t_2, \theta_2) \dots (t_k, \theta_k)$ and $(t_{k+1}, \theta_{k+1})(t_{k+2}, \theta_{k+2}) \dots (t_n, \theta_n)$ such that

$$(M_0, I_0)[(t_1, \theta_1)(t_2, \theta_2) \dots (t_k, \theta_k)\rangle(M, I)$$
$$[(t_{k+1}, \theta_{k+1})(t_{k+2}, \theta_{k+2}) \dots (t_n, \theta_n)\rangle(M', I')$$

where $\omega = t_{k+1}t_{k+2} \dots t_n$, (M, I) belong to class (M, D) and (M', I') belong to class (M', D').

The *reachability graph* associated to the net is obtained from this tree by grouping equal classes into the same one.

Definition 2.11 Two state classes (M, D) and (M', D') *are equal* if $M = M'$ and $D = D'$.

In what follows we shall show that the reachability and the boundedness are undecidable for time Petri nets.

We shall present the proof given in [23] where it is shown that time Petri nets can simulate deterministic counter machines.

Let $A = (Q, q_0, q_f, C, x_0, I)$ be a DCM. We associate to it a time Petri net $\gamma' = (\Sigma, I)$ as follows:

- for each instruction $I(q, c, q')$ we define a structure as in Figure 2.2.
- for each instruction $I(q, c, q', q'')$ we define a structure as in Figure 2.3.

A configuration $\sigma = (q, x)$ of A is simulated by the state (M_σ, I_σ) where:

$$M_\sigma(s) = \begin{cases} 1, & \text{if } s = s_q, \\ 0, & \text{if } s = s'_q, \ \forall q' \in Q - \{q\}, \\ x(c) & \text{if } s = s_c, \ \forall c \in C, \end{cases}$$

and I_σ contains the static firing intervals of the transitions enabled at M_σ.

We call $\gamma = (\gamma', (M_{\sigma_0}, I_{\sigma_0}))$ the TPN *associated to* A.

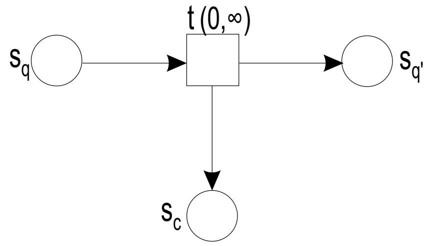


Figure 2.2: The structure associated to the instruction $I(q, c, q')$

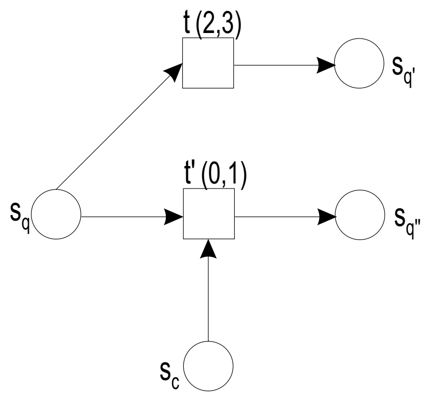


Figure 2.3: The structure associated to the instruction $I(q, c, q', q'')$

Lemma 2.2 Let $A = (Q, q_0, q_f, C, x_0, I)$ be a DCM, γ be its associated time Petri net, and $\sigma = (q, x)$ and $\sigma' = (q', x')$ be two configurations of A. Then, the following properties holds:

1. if $(q, x) \vdash_A (q', x')$ then, there exists a transition t and a time θ such that in γ we have:

$$(M_\sigma, I_\sigma)[(t, \theta)\rangle_\gamma (M_{\sigma'}, I_{\sigma'});$$

2. if $(M_\sigma, I_\sigma)[(t, \theta)\rangle_\gamma (M_{\sigma'}, I_{\sigma'})$ for some $t \in T$ and $\theta \in \mathbf{Q}^+$ then

$$(q, x) \vdash_A (q', x').$$

Proof (1) Let τ be the moment M_σ was reached.

If $(q,x) \vdash_A (q',x')$ by an increment instruction $I(q,c,q')$, then in γ, the only transition enabled at M_σ is t. t can fire at any time $\theta = \tau + \theta'$, with $\theta' \geq 0$, reaching the state $(M_{\sigma'}, I_{\sigma'})$.

If $(q,x) \vdash_A (q',x')$ by a test instruction $I(q,c,q_1,q_2)$ and $x(c) = 0$ then $q' = q_1$. According to the structure in Figure 2.3, t is the only transition enabled at the marking M_σ in γ and it can fire at any time between 2 and 3 units of time once it became enabled, yielding the state $(M_{\sigma'}, I_{\sigma'})$.

If $(q,x) \vdash_A (q',x')$ by a test instruction $I(q,c,q_1,q_2)$ and $x(c) > 0$ then $q' = q_2$. At the marking M_σ there are two enabled transitions t and t'. But t' must fire at most one unit of time after it become enabled, disabling t which must wait at least two units of time before firing. The state reached after t' fires at any time $\theta = \tau + \theta'$, with $0 \leq \theta' \leq 1$, is $(M_{\sigma'}, I_{\sigma'})$.

(2) In the state (M_σ, I_σ) of γ there is one token in the place s_q and $x(c)$ token in each place s_c. If s_q has one output transition t then, in A the instruction for the state q is an increment one $I(q,c,q')$. The firing of t at time θ yields the new state $(M_{\sigma'}, I_{\sigma'})$ with one token in $s_{q'}$, $x(c)+1$ tokens in s_c and $x(c')$ tokens in s'_c, $c \neq c'$. This state corresponds to the configuration $\sigma' = (q', x')$ reached by executing the instruction $I(q,c,q')$ in A

If s_q has two output transitions t and t' then, in A the instruction for the state q is a test instruction $I(q,c,q_1,q_2)$. The firing of (t,θ) at (M_σ, I_σ) corresponds to the situation in which the counter c is 0 and the firing of (t',θ) at (M_σ, I_σ) corresponds the situation in which $x(c) > 0$. In both cases, the state $(M_{\sigma'}, I_{\sigma'})$ reached by firing (t,θ) or (t',θ) at (M_σ, I_σ) is associated to the configuration $\sigma' = (q', x')$ reached by executing the instruction $I(q,c,q_1,q_2)$ in A. $\square$

We define now two new time Petri nets obtained from the time Petri net γ associated to a DCM A:

- $\gamma_1 = (\Sigma_1, I_1, (M^1_{\sigma_0}, I^1_{\sigma_0}))$ is obtained from γ as described in Figure 2.4, where $t_1, \ldots, t_r$ are new transitions with time intervals $(0,1)$. The initial marking $M^1_{\sigma_0}$ marks all the places as M_{σ_0} does.

- $\gamma_2 = (\Sigma_2, I_2, (M^2_{\sigma_0}, I^2_{\sigma_0}))$ is obtained from γ as described in Figure 2.5, where $\tilde{s}$ is a new place. The initial marking is

$$M^2_{\sigma_0}(s) = \begin{cases} M_{\sigma_0}(s), & s \in S \\ 0, & s = \tilde{s} \end{cases}$$

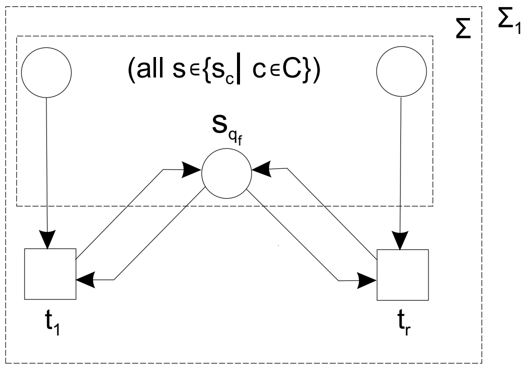


Figure 2.4: Undecidability of reachability for TPN

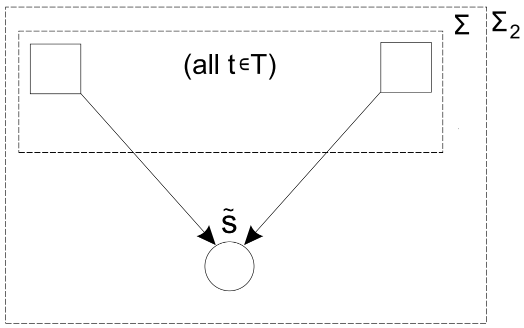


Figure 2.5: Undecidability of boundedness for TPN

Theorem 2.1 Let $A = (Q, q_0, q_f, C, x_0, I)$ be a DCM, γ be its associated time Petri net and γ_i, $i = 1, 2$, be the TPNs constructed as above.

1. the state (M_σ, I_σ) such that

$$M_\sigma(s) = \begin{cases} 1, & s = s_{q_f} \\ 0, & s \neq s_{q_f} \end{cases}$$

   is reachable in γ_1 if and only if A halts;

2. γ_2 is bounded if and only if A halts.

Proof (1) If A halts then $(q_0, x_0) \vdash_A (q_f, x)$. According to Lemma 2.2 in γ a state whose marking has a token in s_{q_f} and some tokens in some s_c is reachable from the initial state. This state is reached in γ_1 as well. At this state, we can fire the transitions t_i which remove tokens from places s_c yielding the state (M_σ, I_σ). Therefore (M_σ, I_σ) is reachable in γ_1.

Conversely, if (M_σ, I_σ) is reachable in γ_1 then, there exists a feasible firing schedule ω such that $(M_{\sigma_0}, I_{\sigma_0})[\omega\rangle_{\gamma_1}(M_\sigma, I_\sigma)$. According to Lemma 2.2, in A we have $(q_0, x_0) \vdash_A (q_f, x)$ which means that A halts.

(2) If A halts then the final state q_f is reached after applying a finite numbers of instructions. Thus, in all markings of γ_2 each place can hold a finite numbers of tokens. Therefore, γ_2 is bounded.

Conversely, let γ_2 be bounded. If we assume that A does not halt, then in A an infinite sequence of instructions can be applied. According to Lemma 2.2, an infinite sequence of transitions is firing, each transition adding a token in $\tilde{s}$. Thus, $\tilde{s}$ holds an infinite number of tokens contradicting the fact that γ_2 is bounded. □

Corollary 2.1 The reachability and the boundedness are undecidable for time Petri nets.

Proof Follows from Theorem 2.1. □

In what follows, we shall give two sufficient conditions for boundedness ([6]).

Lemma 2.3 Let $\gamma = (\Sigma, I)$ be a T-bounded TPN. If γ is unbounded then, there exists an infinite sequence of state classes $(M_0, D_0), (M_1, D_1), \ldots$ in which all classes are pairwise different, such that:

$$(M_0, D_0)[t_1\rangle_\gamma(M_1, D_1)[t_2\rangle_\gamma \ldots$$

where $t_i \in T$.

Proof If γ is unbounded then, for any $n \in \mathbf{N}$ there exists a state class (M, D) reachable from (M_0, D_0) such that $M(s) > n$, for some place $s \in S$. Since each state class has only a finite number of successors, from the definition of the firing rule, there exists an infinite sequence of state classes reachable from (M_0, D_0)

$$(M_1, D_1), (M_2, D_2), \ldots.$$

with $(M_i, D_i) \neq (M_j, D_j)$, for all i, j, $i \neq j$, and there exists $t_i \in T$ such that:

$$(M_0, D_0)[t_1\rangle_\gamma(M_1, D_1)[t_2\rangle_\gamma \ldots.$$

□

Theorem 2.2 $(SC1)$ A T-bounded TPN γ is bounded if there is no pair of state classes (M, D) and (M', D') reachable from the initial state class such that:

(a) (M', D') is reachable from (M, D);

(b) $M'(s) \geq M(s)$, for all $s \in S$.

Proof Assume that γ is unbounded and consider the infinite sequence

$$\omega = (M_0, D_0), (M_1, D_1), \ldots$$

defined in Lema 2.3. Since γ is T-bounded, there is only a finite number of firing variables associated to the enabled transitions at any marking. Hence, there is only a finite number of domains. As state classes are pairwise different, there exists an infinite subsequence

$$\omega' = (M_{i_0}, D_{i_0}), (M_{i_1}, D_{i_1}), \ldots$$

of ω such that $M_{i_j} \neq M_{i_k}$, for all i, j, $i \neq j$ and ω' contains two state classes (M, D) and (M', D') with $M'(s) \geq M(s)$, for all $s \in S$. $\square$

Theorem 2.3 $(SC2)$ A TPN γ is bounded if no pair of states classes (M, D) and (M', D') reachable from the initial state class holds the following conditions:

(a) (M', D') is reachable from (M, D);

(b) $M'(s) \geq M(s)$, for all $s \in S$;

(c) $D' = D$;

(d) for all $s \in \{s \in S \mid M'(s) > M(s)\}$ we have $M(s) > max\{W(s,t) \mid t \in T\}$.

Proof Assume that γ is unbounded and consider the infinite sequence ω' from Theorem 2.2. Since the number of distinct subsets of S is finite ω' must contain an infinite subsequence which has two state classes (M, D) and (M', D') such that $D = D'$, $M' \geq M$ and $\forall s \in \{s \in S \mid M'(s) > M(s)\}$ we have $M(s) > k(s)$, where k is any mapping associating an integer with any place. Because a transition's firing can remove maximum $max\{W(s,t) \mid t \in$

$T\}$ tokens, k may be chosen to be greater then all related transition weights for any place s: $k(s) = max\{W(s,t) \mid t \in T\}$. $\square$

It is clear that $(SC2) \Rightarrow (SC1)$.

The second type of firing rule for time Petri nets have been introduced in [36]. The time Petri nets are defined like in [6].

Because it is easier to study the behavior of TPNs whose $seft$ and $slft$ are natural numbers it is shown that for each TPN can be found another one which have naturals $seft$ and $slft$.

Definition 2.12 Let $\gamma = (\Sigma, I)$ and $\gamma' = (\Sigma', I')$ be two time Petri nets, γ and γ' are *time equivalent* if the following properties holds:

1. $\Sigma = \Sigma'$;
2. There exists a constant $c \neq 0$ such that, for each $t \in T$ we have:
   (a) $I_2(t) = \infty$ if and only if $I'_2(t) = \infty$;
   (b) $I_1(t) = 0$ if and only if $I'_1(t) = 0$ and $I_2(t) = 0$ if and only if $I'_2(t) = 0$;
   (c) $\frac{I_1(t)}{I'_1(t)} = c$ if $I'_1(t) \neq 0$;
   (d) $\frac{I_2(t)}{I'_2(t)} = c$ if $I'_2(t) \neq 0$.

Theorem 2.4 Let γ be a TPN. Then, there exists a TPN γ' with $I' : T \to \mathbf{N} \times (\mathbf{N} \cup \{\infty\})$ which is time equivalent with γ.

Proof We compute the least common multiple c of the denominators of all $I_1(t)$ and $I_2(t)$, $t \in T$. Then, we define $I'_1(t)$ and $I'_2(t)$ as the product of $I_1(t)$ and $I_2(t)$ with c. $\square$

All the TPNs used below, have the time function defined from T into $\mathbf{N} \times (\mathbf{N} \cup \{\infty\})$.

Before giving the definition of the firing rule we shall present the notion of state as defined in [36].

Definition 2.13 Let $\gamma = (\Sigma, I)$ be a time Petri net and $J : T \to \mathbf{Q} \cup \{\#\}$. A pair (M, J) is called *a state* of γ if:

- M is a reachable marking in Σ;

- $\forall t(t \in T \wedge W(s,t) \leq M(s)$, for all $s \in S \Rightarrow J(t) \leq I_2(t))$;
- $\forall t(t \in T \wedge \exists\, s \in S$ such that $W(s,t) > M(s) \Rightarrow J(t) = \#)$.

An interpretation of a state is as follows: in the net each transition t has a watch. The watch does not work ($J(t) = \#$) at the marking M if t is disabled at M. If t is enabled at M the watch of t shows the time ($J(t)$) elapsed since t was last enabled.

Definition 2.14 Let γ be a TPN. A state (M_0, J_0) is called the *initial state* of γ if

$$J_0(t) = \begin{cases} 0, & \text{if } W(s,t) \leq M(s), \text{ for all } s \in S; \\ \#, & \text{if } \exists\, s \in S \text{ such that } W(s,t) > M(s). \end{cases}$$

Unlike [6] where it is specified the time at which a transition fires, in [36] the time is explicitly passed. The firing rules are essentially the same, but there are defined completely different methods for building the reachability graph for a TPN.

Definition 2.15 A transition t is *ready to fire* at the state (M, J), denote by $(M, J)[t\rangle_\gamma$, if

1. $W(s,t) \leq M(s)$, for all $s \in S$;
2. $I_1(t) \leq J(t)$.

The transition t is ready to fire at the state (M, J) if t is enabled at the marking M and the watch of t shows its eft or a later time.

Definition 2.16 Let $\epsilon \in T \cup \mathbf{Q}$. We write $(M, J)[\epsilon\rangle_\gamma(M', J')$ if one of the following cases applies:

1. if $\epsilon = t$ then:
   (a) t is ready to fire at (M, J);
   (b) $M'(s) = M(s) - W(s,t) + W(t,s)$, for all $s \in S$ and

(c)

$$J'(t) = \begin{cases} \#, & \text{if } \exists\, s \in S \text{ such that } W(s,t) > M'(s); \\ J(t), & \text{if } W(s,t) \leq M(s) \wedge W(s,t) \leq M'(s), \text{ for all } s \in S \\ & \wedge\, {}^\bullet t \cap {}^\bullet t' = \emptyset, \text{ for all } t' \in T, t' \neq t; \\ 0, & \text{otherwise;} \end{cases}$$

2. if $\epsilon = (r)$ for some $r \in \mathbf{Q}$, then:
   (a) $\forall t(t \in T \wedge J(t) \neq \# \Rightarrow J(t) + r \leq I_2(t))$;
   (b) $M' = M$ and
   (c)

$$J'(t) = \begin{cases} \#, & \text{if } \exists\, s \in S \text{ such that } W(s,t) > M(s); \\ J(t) + r, & \text{if } W(s,t) \leq M(s), \text{ for all } s \in S. \end{cases}$$

We say that the state (M', J') is *reachable* from (M, J) in γ if there exists $\epsilon_1, \ldots, \epsilon_n \in T \cup \mathbf{Q}$ such that

$$(M, J)[\epsilon_1\rangle(M_1, J_1) \ldots (M_{n-1}, J_{n-1})[\epsilon_n\rangle(M', J').$$

where (M_i, J_i), $i = 1, \ldots, n-1$, are states of γ. $[(M, J)\rangle_\gamma$ stands for the set of all reachable states in γ (from (M, J)).

Definition 2.17 1. Let γ be a TPN, (M, J) and (M', J') two states of it, $w \in T^*$ and $\xi \in (\mathbf{Q}_0^+)^{|w|+1}$. The state (M, J) *changes into the state* (M', J') *by* w *and* ξ, denoted by $(M, J)[w_\xi\rangle(M', J')$, if

   a) $w = \lambda$ then $(M', J') = (M, J)$ and $(M, J)[\lambda\rangle(M, J)$;
   b) $(M, J)[w_\xi t_r\rangle(M', J')$ if there exists two states (M'', J'') and (M''', J'''), such that $(M, J)[w_\xi\rangle(M'', J'')$, $(M'', J'')[t\rangle(M''', J''')$ and $(M''', J''')[r\rangle(M', J')$.

2. Let γ be a TPN and (M, J) a reachable marking of it. The sequence $w \in T^*$ *can fire* at (M, J) in γ, $((M, J)[w.\rangle)$ if there exists a sequence of rational numbers ξ and a state (M', J') such that $(M, J)[w_\xi\rangle(M', J')$.

3. Let γ be a TPN and (M, J) a reachable marking of it. The sequence $w \in T^*$ *can integer–fire* at (M, J) in γ, $((M, J)[w._i\rangle)$ if there exists a sequence of integers ξ and a state (M', J') such that $(M, J)[w_\xi\rangle(M', J')$.

Definition 2.18 A TPN γ is said to be of *finite delay* if $I_2(t) < \infty$, for all $t \in T$.

Definition 2.19 The state (M, J) is called an *integer-state* if for each enabled transition t at M, $J(t)$ is an integer.

Time Petri nets have generally an infinite number of states. However, only a finite number of integer-states belong to each marking of a TPN of finite delay.

Definition 2.20 The graph $R_\gamma((M_0, J_0))$ is called the *reachability graph* of a TPN if its nodes are integer-states reachable from (M_0, J_0) and its arcs are defined as triples $((M, J), r, (M', J'))$, respectively $((M, J), t, (M', J'))$, where $(M, J)[r\rangle_\gamma(M', J')$ and $(M, J)[t\rangle_\gamma(M', J')$.

Definition 2.21 A TPN γ is *bounded* if there exists a natural number n such that $M(s) \leq n$, for every state (M, J) reachable from (M_0, J_0) and every place $s \in S$.

The following results are proved in [36].

Theorem 2.5 Let γ be a TPN of finite delay. γ is bounded if and only if $R_\gamma((M_0, J_0))$ is finite.

Definition 2.22 L_γ is called the *language* of a TPN γ if

$$L_\gamma = \{w \mid w \in T^* \wedge (M_0, J_0)[w.\rangle_\gamma\}.$$

Definition 2.23 Let γ be a TPN and L_γ its language.

1. A transition t is called *live* at (M, J) if

$$\forall(M', J')((M', J') \in [(M, J)\rangle_\gamma \Rightarrow$$
$$\exists(M'', J'')((M'', J'') \in [(M', J')\rangle_\gamma \wedge (M'', J'')[t\rangle_\gamma).$$

2. γ is *live* if all the transitions t are live in γ.

3. L_γ is called *live* if

$$\forall w \forall t(w \in L_\gamma \wedge t \in T \Rightarrow \exists u(u \in T^* \wedge wut \in L_\gamma))$$

Theorem 2.6 Let γ be a TPN. γ is live if and only if L_γ is live.

Let $\gamma = (\Sigma, I)$ be a TPN of finite delay and (M, J) a state of it. We shall transform γ into a time equivalent TPN $\gamma^* = (\Sigma, I^*)$ as follows:

Let

$$J(t) = \begin{cases} \#, & \text{if } \exists\, s \in S \text{ such that } W(s,t) > M(s); \\ \frac{p_t}{q_t}, & \text{if } W(s,t) \le M(s), \text{ for all } s \in S. \end{cases}$$

where $p_t, q_t \in \mathbf{N}$.

We consider I^* such that

$$I_1^*(t) = I_1(t) \cdot r,$$

$$I_2^*(t) = I_2(t) \cdot r.$$

where

$$r = \text{ the least common multiple of } \left\{ \frac{p_t}{q_t} \mid W(s,t) \le M(s), \text{ for all } s \in S \right\}.$$

It holds $I^*(t) \in \mathbf{N} \times \mathbf{N}$, for each $t \in T$.

Let

$$J^*(t) = \begin{cases} \#, & \text{if } J(t) = \#,; \\ J(t) \cdot r, & \text{otherwise.} \end{cases}$$

for each $t \in T$.

It is clear that $J^*(t) \in \mathbf{N} \cup \{\#\}$.

Theorem 2.7 Let γ be a TPN of finite delay, (M, J) a state of it, γ^*, (M^*, J^*) be defined as above. Then, (M, J) is reachable in γ if and only if (M^*, J^*) is reachable in γ^*.

In [37] two classes of time Petri nets which have the same liveness and boundedness behavior are presented. The first class contains all time Petri nets whose transitions can (but do not have to) fire immediately. The second one is defined by the constraint: the $slft$ of all transitions are infinite. Thus, in these nets each transition has to wait for a certain time in order to fire when it becomes enabled but it can not be forced to fire at any time.

The following properties are structural ones and therefore easy to be proved.

Proposition 2.1 Let $\gamma = (\Sigma, I)$ be a TPN such that $I_1(t) = 0$, for all $t \in T$. The following properties holds:

1. Σ is unbounded if and only if γ is unbounded, and
2. Σ is live if and only if γ is live.

Proof (1) Let Σ be unbounded. We shall show that γ is unbounded too. We have to prove that

$$\forall n(n \in \mathbf{N} \Rightarrow \exists(M, J)\exists s((M, J) \in [(M_0, J_0)\rangle_\gamma \wedge s \in S \wedge M(s) > n)).$$

Because Σ is unbounded, for each natural number n there exists a reachable marking $M^n \in [M_0\rangle_\Sigma$ and a place $s \in S$ such that $M^n(s) > n$.

Since the marking M^n is reachable in Σ there exists a sequence

$$M_0[t_1\rangle M_1[t_2\rangle \ldots [t_k\rangle M_k = M^n$$

We consider the states (M_i, J_i) with

$$J_i(t) = \begin{cases} 0, & \text{if } W(s,t) \leq M(s), \text{ for all } s \in S \\ \#, & \text{if } \exists\ s \in S \text{ such that } W(s,t) > M(s). \end{cases}$$

It is easy to see that the sequence

$$(M_0, J_0)[t_1\rangle(M_1, J_1)[t_2\rangle \ldots [t_k\rangle(M_k, J_k) = (M^n, J^n)$$

is enable in γ too. Therefore, the state $(M_k, J_k) = (M^n, J^n)$ is reachable in γ and (M^n, J^n) satisfies that $\exists s \in S : M^n(s) > n$.

The converse is straightforward.

(2) Let Σ be live. Then

$$\forall t \forall M(t \in T \wedge M \in [M_0\rangle_\Sigma \Rightarrow \exists M'(M' \in [M\rangle_\Sigma \wedge M'[t\rangle_\Sigma)).$$

We shall show that γ is live too.

Let (M, J) be an arbitrary reachable state and t be an arbitrary transition of γ. It is enough to show that

$$\exists(M', J')((M', J') \in [(M, J)\rangle_\gamma \wedge (M', J')[t\rangle_\gamma. \quad (2.11)$$

The inequality $I_1(t) \leq J'(t)$ is always true, because $I_1(t') = 0$, for all $t' \in T$. Thus, we are looking for a state (M', J'), reachable from (M, J) in γ, with the property: $W(s,t) \leq M'(s)$, for all $s \in S$.

Since (M, J) is a reachable state in γ it is easy to see that M is a reachable marking in Σ. Because Σ is live there exists a marking M' reachable from M and $W(s,t) \le M'(s)$, for all $s \in S$. Thus, there exists a sequence $t'_1, \ldots, t'_r$ with

$$M = M'_0[t'_1\rangle M'_1[t'_2\rangle \ldots [t'_r\rangle M'_r = M'$$

in Σ and $W(s,t) \le M'(s)$, for all $s \in S$.

Now, we consider the state sequence $(M'_0, J'_0), \ldots, (M'_r, J'_r)$ with

$$J'_0(t) = J(t)$$

$$J'_{i+1}(t) = \begin{cases} J'_i(t), & \text{if } W(s,t) \le M_{i+1}(s) \wedge W(s,t) \le M_i(s), \text{ for all } s \in S \\ & \wedge\ {}^\bullet t \cap {}^\bullet t'_i = \emptyset; \\ \#, & \text{if } \exists\ s \in S \text{ such that } W(s,t) > M_{i+1}(s); \\ 0, & \text{otherwise} \end{cases}$$

for each $i = 0, \ldots, r-1$.

Obviously, this last sequence can fire in γ (from (M, J)) and therefore the state $(M', J') = (M'_r, J'_r)$ is a state that satisfies 2.11.

Conversely, let γ be live. Then, for each transition $t \in T$ and each state (M, J) reachable from (M_0, J_0) in γ there exists a state (M', J') reachable from (M, J) and t is ready to fire at (M', J'). We shall show that Σ is live too.

Let M be an arbitrary marking reachable from M_0 and t an arbitrary transition of Σ. We have to show that

$$\exists M'(M' \in [M\rangle_\Sigma \wedge M'[t\rangle_\Sigma).$$

Because $M \in [M_0\rangle_\Sigma$ there exist the sequences $M_0, \ldots, M_n = M$ and $t_1, \ldots, t_n$ such that in Σ we have:

$$M_0[t_1\rangle M_1[t_2\rangle \ldots [t_n\rangle M_n = M.$$

Now, considering the state sequence $(M_0, J_0), (M_1, J_1), \ldots, (M_n, J_n)$ with

$$J_i(t) = \begin{cases} 0, & \text{if } W(s,t) \le M(s), \text{ for all } s \in S \\ \#, & \text{if } \exists\ s \in S \text{ such that } W(s,t) > M(s). \end{cases}$$

it is easy to see that this sequence can fire in γ. Therefore, $(M, J) = (M_n, J_n)$ is a reachable state from (M_0, J_0) in γ.

Because γ is live, we can find a state (M', J') with the property:

$$(M', J') \in [(M, J)\rangle_\gamma \wedge W(s, t) \le M'(s), \text{ for all } s \in S, \wedge I_1(t) \le J'(t).$$

Let $(M, J) = (M'_0, J'_0)[r_0 t'_1 r_1\rangle(M'_1, J'_1)[t'_2 r_2\rangle \ldots [t'_k r_k\rangle(M_k, J_k) = (M', J')$. We have that

$$M = M'_0[t'_1\rangle M'_1[t'_2\rangle \ldots [t'_2\rangle M'_k = M'$$

and $W(s, t) \le M'(s)$ for all $s \in S$. $\square$

Proposition 2.2 Let $\gamma = (\Sigma, I)$ be a TPN such that $I_2(t) = \infty$, for all $t \in T$. Then, it holds:

1. Σ is unbounded if and only if γ is unbounded, and
2. Σ is live if and only if γ is live.

Proof (1) Let Σ be unbounded. We shall show that γ is unbounded too.

Let n be an arbitrary natural number. Then, because of the unboundedness of Σ there is a place $s \in S$ and a reachable marking M of Σ such that $M(s) > n$. Thus, in Σ there exists a sequence $M_0[t_1\rangle M_1[t_2\rangle \ldots [t_k\rangle M_k = M$.

We consider the state sequence

$$(M_0, J_0)[r_0\rangle(M'_0, J'_0)[t_1\rangle(M_1, J_1)[r_1\rangle(M'_1, J'_1)[t_2\rangle(M_2, J_2)[r_2\rangle \ldots$$
$$[t_k\rangle(M_k, J_k)[r_k\rangle(M'_k, J'_k) \qquad (2.12)$$

where $r_i = max\{I_1(t) | W(s, t) \le M_i(s), \text{ for all } s \in S\}$, for each $i = 0, \ldots, k$.

The sequence 2.12 can fire in γ because the $slft$ of all transitions are infinite. Thus, the state (M, J_k) where $M = M_k$ is reachable in γ and therefore γ is unbounded.

Conversely, the proof is obvious.

(2) It is easy to prove this direction using the same construction as in the proof of the Proposition 2.1 (2).

Conversely, let γ be live, that is, for each transition $t \in T$ and each state (M, J) reachable from (M_0, J_0) there exists a (M', J') reachable from (M, J) such that t is enabled at (M', J'). We shall show that Σ is live too.

Let M be an arbitrary marking reachable from M_0 and t an arbitrary transition of Σ. There exist the sequences $M_0, \ldots, M_n = M$ and $t_1, \ldots, t_n$ such that in Σ we have

$$M_0[t_1\rangle M_1[t_2\rangle \ldots [t_n\rangle M_n = M.$$

Analogously to the construction in the proof of the Proposition 2.1 (2), we obtain a state $(M, J) \in [(M_0, J_0)\rangle_\gamma$. Now we can find a sequence in γ, starting with the state (M, J) and ending with a state in which the transition t is ready to fire (analogously to the proof in Proposition 2.1 (2)). $\square$

2.2 Timed Petri Nets

Timed Petri nets have been introduced by Ramchandani in 1974 [41] as a tool for performance analysis of real-time systems [40, 57, 42, 17]. These kinds of Petri nets are simply obtained by associating firing finite time durations to each transition of ordinary Petri nets. The classical firing rule was modified to account for the time it takes to fire transitions.

In 1980 timed Petri nets were used by Ramamoorthy and Ho for performance evaluation of asynchronous concurrent systems [40]. Further, in 1985 (1987), Holliday and Vernon considered a little modified version of these nets in order to evaluate the performance of computer systems [17]. They added about durations also firing frequencies and resources to the transitions in the net.

Between 1980 and 1985 general references could be done at Zuberek's variant of timed Petri nets from 1980 [57] and Razouk and Phelps's variant [42].

Previous studies of 1989 are concentrated in showing how timed Petri nets can be used for performance analysis of the systems. The paper of Starke from 1989 [49] is probably the first one that initiates a study on the nature and properties of these types of nets. Starke defines timed Petri nets as follows:

Definition 2.24 A *timed Petri net* (T_dPN) is a tuple $\gamma = (\Sigma, \delta)$, where:

- $\Sigma = (S, T, F, W, M_0)$ is a marked Petri net called the *underlying net of* γ;

- $\delta : T \rightarrow \mathbf{R}_0^+$ is the *duration function* of γ that associates to each transition a nonnegative real number.

For each transition t, the number $\delta(t)$ is interpreted as the duration of the firing of t, that is, if t fires at time β, the corresponding numbers of tokens are removed from the pre-places of t at time β and the corresponding numbers of tokens are added on the post-places of t at time $\beta + \delta(t)$.

The duration function δ of a T_dPN is pictorially represented by labeling each transition t by $\delta(t)$.

Definition 2.25 A transition t is *enabled* at a marking M in γ if it is enabled at M in Σ.

Several firing strategies have been defined for timed Petri nets [49, 45, 46, 56, 22], and all of them are based on the concept of a *state*.

The firing strategy used by Starke in [49] is the so-called *earliest firing schedule*. According to it, if a transition t is enabled at the marking M at time β, it is obliged to fire at that time.

The duration $\delta(t) = 0$ raises problems in determining the marking at time points $\beta > 0$ because a firing sequence that yields to the moment β should uniquely determine the marking at that moment. If $\delta(t) = 0$ for a transition t that is enabled at time β, then t can change the marking without consuming time. For this reason, there are considered positive durations $\delta(t) > 0$ for all transitions t. Moreover, there are assumed the values $\delta(t)$ to be rational, that is, $\delta : T \rightarrow \mathbf{Q}^+$.

Since T is finite, the time scale can be stretched in such a way that within the new time scale all the durations are positive natural numbers (by multiplication with the least common multiple of the denominators of all quotients $\delta(t)$) without changing the structure of the reachability set and, therefore, without changing the properties of the net.

Definition 2.26 A *discretely timed Petri net* is a timed Petri net $\gamma = (\Sigma, \delta)$, where $\delta : T \rightarrow \mathbf{N}^+$.

Due to the fact that in [49] there are considered only discretely timed Petri nets we shall refer to them as being T_dPN.

In [49] transitions neither are allowed to fire concurrently with themselves nor are allowed to start a second firing before the first one is finished. Moreover, the *maximum firing rule* is used, that is, at any point of time, a maximal set of concurrently transitions (*maximal step*) is fired.

Definition 2.27 Let $\gamma = (\Sigma, \delta)$ be a T_dPN. A *state of* γ is a pair (M, u), where M is a marking of Σ and u is a function $u : T \to \mathbf{N}$.

$u(t) = 0$ means that t is not currently firing at the state (M, u) and $u(t) > 0$ is the time elapsed since the transition t has started its firing.

The initial state of γ is $(M_0, \underline{0})$, where $\underline{0}(t) = 0$ for all $t \in T$.

Definition 2.28 Let $\gamma = (\Sigma, \delta)$ be a T_dPN and (M, u) a state of it.

1. A subset $U \subseteq T$ is a *step* at (M, u) if
   (a) $u(t) = 0$ for all $t \in U$;
   (b) if $U = \emptyset$ then u is not identically zero;
   (c) $\sum_{t \in U} W(s, t) \leq M(s)$, for all $s \in S$.
2. U is a *maximal step* at (M, u) if U is a step at (M, u) and there is no step U' at (M, u) such that $U \subset U'$.

Therefore, a step at the state (M, u) is a set of transitions which are not currently firing at (M, u), such that there are enough tokens to fire them all concurrently. The empty set is a step at the state (M, u) if u is not identically zero, that is, if there is at least a transition which is firing at (M, u).

Definition 2.29 Let $\gamma = (\Sigma, \delta)$ be a T_dPN, (M, u) a state of it and U a maximal step at (M, u). The firing of U at (M, u) at time β yields the new state (M', u') at time $\beta + 1$, denoted by $(M, u)[U\rangle_\gamma(M', u')$, where:

1. $M'(s) = M(s) - \sum_{t \in U} W(s, t) + \sum_{t \in U \wedge \delta(t)=1} W(t, s) + \sum_{u(t)=\delta(t)-1} W(t, s)$, for all $s \in S$;
2. $u'(t) := \begin{cases} 1, & \text{if } t \in U \wedge \delta(t) > 1, \\ u(t) + 1, & \text{if } t \in T - U \wedge 0 < u(t) < \delta(t) - 1, \\ 0, & \text{otherwise.} \end{cases}$

The firing of a step at a state at time β yields a new state at time $\beta + 1$. The marking at the time $\beta + 1$ automatically considers the effect of firing all the transitions in the step which have the firing duration 1 and of the transitions which are already firing and whose firing durations elapse at time $\beta + 1$.

In [49] it was shown that discretely timed Petri nets working under the maximum firing rule are equivalent with classical Petri nets and there were presented two conditions under which a live Petri net remains live if arbitrary firing durations are imposed to transitions.

The approach in [45] extends the one in [49] by considering steps consisting of multisets of transitions (and not necessarily sets of transitions).

In this case, a *state* is defined as a pair formed by a marking M and, instead of a function $u : T \to \mathbf{N}$, it is considered a finite multiset of pairs $M' : T \times \mathbf{N}^+ \to \mathbf{N}$ such that, for any $t \in T$ and any $\tau \geq \delta(t)$ we have $M'(t,\tau) = 0$. $M'(t,\tau) > 0$ means that there are $M'(t,\tau)$ instances of the transition t currently firing and each of them has $\tau - 1$ units of time until their completion.

The firing rule is exactly the one in [49] adjusted to multisets of transitions (the weights $W(t,s)$ and $W(s,t)$ are multiplied by the multiplicity of t in the multiset that fires).

A step semantics is associated to a timed Petri net as follows:

Definition 2.30 Let $\gamma = (\Sigma, \delta)$ be a T_dPN and M_0 a marking of Σ. We say that $\sigma = M_0[R_0\rangle \ldots M_{n-1}[R_{n-1}\rangle M_n$ is a *finite timed step sequence* of γ, denoted by $M_0[\sigma\rangle M$ if:

- $\forall i \in \{0, \cdots, n-1\}$: R_i is a multiset of transitions over T (which can be empty);
- $\forall i \in \{1, \cdots, n\}$: $M_{i-1}[R_{i-1}\rangle M_i$, where M_{i-1} and M_i are markings of Σ.

From σ we can construct a corresponding step sequence

$$\sigma' = M_0[R_1'^{(\beta_1)}\rangle \cdots [R_n'^{(\beta_n)}\rangle M_n,$$

denoted by $M_0[\sigma'^{(\beta)}\rangle M$, where:

- $\beta_1 < \beta_2 < \cdots < \beta_n$ are moments of time;
- $\beta_n = \beta$;
- R_i' is a nonempty multiset of transitions in T.

Definition 2.31 Let $\gamma = (\Sigma, \delta, M_0)$ be a marked T_dPN, $s \in S$, and $k, \beta \in \mathbf{N}$.

1. A marking M is reachable (strictly reachable at the instant β) in γ if there exists a finite timed step sequence σ such that $M_0[\sigma\rangle M$ ($M_0[\sigma^{(\beta)}\rangle M$).

2. γ is *s,k-linearly unbounded* if there exists some $\tau \geq k$ and some markings M reachable from M_0 such that $M(s) \geq \tau$.

3. γ is *uniformly s-linearly unbounded* if for all markings M reachable from M_0 there exists some $\tau \in \mathbf{N}$ and some markings M' reachable from M such that $M'(s) \geq \tau$.

4. A transition t is *β-live* if for any reachable marking M there exists a marking M' strictly reachable from M at the instant β which enables t. γ is *β-live* if all transitions of γ are β-live.

5. A marking M of γ is *dead* if there exists no transition enabled at M. γ *can β-deadlock* if there is some marking M strictly reachable from M_0 at the instant β which is dead.

In [45] it is effectively shown how discretely timed Petri nets can be simulated by classical Petri nets. As a consequence of this simulation all the problems mentioned above are decidable for discretely timed Petri nets.

There are situations in the analysis of a system when it is necessary to consider rational or even real positive firing durations for transitions. Such timed constraints appear more or less explicitly in the papers written until 1990. The firing strategy is different than the one above for the case of rational or real time durations because on the axis of rational or real numbers there is no more an immediate next time of a certain instant [46]. Thus, it has to be considered the effect of the firing of a step at the instant at which it occurs.

For timed Petri nets with rational durations a *marking evolution* is defined. If we know the state at an instant β and we want to apply a step at time $\beta + \tau$, even if in the time interval $[\beta, \beta + \tau]$ no transition fires, we have to update the state by applying all the transitions that are currently firing and complete their firing during this time, and then to apply the step.

Definition 2.32 Let $\gamma = (\Sigma, \delta)$ be a T_dPN with rational positive durations, (M, M') a state of it at an instant $\beta \in \mathbf{Q}$ and $\tau \in \mathbf{Q}$. Then, if there are no transitions that fire in the time interval $[\beta, \beta + \tau]$, the obtained marking at time $\beta + \tau$ is

$$Ev((M, M'), \tau) = (M_1, M_1'),$$

where:

1. $M_1(s) = M(s) + \sum_{t \in C_1} \sum_{\alpha \in End(t,\tau)} M'(t, \alpha) \cdot W(t, s)$, for all $s \in S$

   $$C_1 = \{t \in T \mid \exists \alpha \in \mathbf{Q}^+ : M'(t, \alpha) > 0 \wedge \alpha \leq \tau\},$$

   $$End(t, \tau) = \{\alpha \in \mathbf{Q}^+ \ \mid \ M'(t, \alpha) > 0 \wedge \alpha \leq \tau\}, \text{ for each } t \in C_1;$$

2. $M_1'(t, \beta') = M'(t, \beta' + \tau)$.

Definition 2.33 Let $\gamma = (\Sigma, \delta)$ be a $T_d PN$ with rational positive durations, (M, M') a state of it at an instant β, and $\tau \in \mathbf{Q}$.

1. A multiset of transitions R is *enabled* at the state (M, M') at time $\beta + \tau$ (assuming that nothing else has been fired in the meantime) if for the marking $(M_1, M_1') = Ev((M, M'), \tau)$ we have:

   $$\forall s \in S, \ M_1(s) \geq \sum_{t \in T} R(t) \cdot W(s, t).$$

2. If a multiset of transitions R is enabled at a state (M, M') at an instant $\beta + \tau$, the firing of it at that instant yield at the same time the state $(\widetilde{M}_1, \widetilde{M'}_1)$ given by:

   (a) $\widetilde{M}_1(s) = M_1(s) - \sum_{t \in T} R(t) \cdot W(s, t)$, for all $s \in S$;

   (b) $$\widetilde{M'}_1(t, \beta') = \begin{cases} R(t), & \text{if } \beta' = \delta(t) \\ M_1'(t, \beta'), & \text{otherwise.} \end{cases}$$

The firing rule is the same for the case of real durations and in the paper it was shown that the strict reachability problem is decidable for timed Petri nets with rational and real durations.

New ideas of timed constraints appear in 1995 in [56]. Starting from the studies connected with the testing of asynchronous systems modeled by Petri nets in [56] there are considered three firing rules for discretely timed Petri nets. Further, another variant of firing rules [22] have been added to the ones mentioned above. Moreover, in [22] the passage of time is allowed in a continuous manner. These variants are the ones we shall use in our thesis to define timed workflow nets.

The timed Petri nets we shall consider in the sequel are like the ones in [49] with the difference that the underlying net is not a marked one and δ is a function from T into $\mathbf{N}$, in other words, there are allowed zero firing durations.

The behavior of these variants of timed Petri nets is based on the following idea: each transition t is decomposed into two parts (t^+, t^-), where t^+ is the part which "starts the activity of t" by applying $W(s,t)$ for all $s \in S$, and t^- is the part which "ends the activity of t" by applying $W(t,s)$ for all $s \in S$. First t^+ must fire; once it fired, it lasts for a time duration according to the type of behavior and then t^- is allowed to fire.

Because in the rest of this section we shall use only such kind of nets, we shall refer to them as T_dPN instead of discretely timed Petri nets.

The behavior of such T_dPN is based on *timed states* which can be defined as in [49]. However, we shall prefer the following variant.

Definition 2.34 A *timed states* of a T_dPN $\gamma = (\Sigma, \delta)$ is a tuples (M, C, ρ), where:

- M is a marking of Σ;
- $C \subseteq T$ is a subset of transitions called *current transitions*;
- $\rho : C \to \mathbf{N}$ is the *residual time function* of the current transitions ($\rho(t)$ is the *residual time* of transition t).

Definition 2.35 A *marked timed Petri Net* (mT_dPN) is a 3-tuple

$$\gamma = (\Sigma, \delta, (M_0, \emptyset, \emptyset)),$$

where (Σ, δ) is a T_dPN and $(M_0, \emptyset, \emptyset)$ is a state of (Σ, δ) called the *initial state* of γ.

In the sequel, we shall present, with slight modifications, the four types of firing rules (L, E, A, and S) that have been introduced in [56] and [22].

Definition 2.36 Let γ be a T_dPN and $\epsilon \in T^{\pm} \cup \{(r)|r \in \mathbf{N} - \{0\}\}$. We write $(M, C, \rho)[\epsilon\rangle_\gamma(M', C', \rho')$ if one of the following cases applies:

1. if $\epsilon = t^+$ then:
   (a) $t \notin C$ and $M[t\rangle_\Sigma$;
   (b) $M'(s) = M(s) - W(s,t)$, for all $s \in S$, $C' = C \cup \{t\}$, and $\rho'(t) = \rho \cup \{(t, \delta(t))\}$;
2. if $\epsilon = t^-$ then:
   (a) $t \in C$ and $\rho(t) = 0$;
   (b) $M'(s) = M(s) + W(t,s)$, for all $s \in S$, $C' = C - \{t\}$, and $\rho' = \rho|_{C'}$;
3. if $\epsilon = (r)$ for some $r \in \mathbf{N} - \{0\}$, then:
   (a) $r \leq min\{\rho(t) \mid t \in C\}$; (type E behavior)
   (b) $(T - C)(M) = \emptyset$; (type A behavior)
   (c) $(T - C)(M) = \emptyset$ and $r \leq min\{\rho(t) \mid t \in C\}$; (type S behavior)
   (d) $M' = M$, $C' = C$, and $\rho' = \rho \dot{-} r$.

In the definition above, items 1, 2, and 3(d) define the *type L* (or *liberal*) *behavior*, items 1, 2, and 3(a)(d) define the *type E* (or *mixed*) *behavior*, items 1, 2, and 3(b)(d) define the *type A behavior*, and items 1, 2, and 3(c)(d) define the *type S* (or *strict*) *behavior* [56, 22]. 1(a) (2(a), 3(a), 3(b), 3(c)) is usually called the *enabling rule*, while 1(b) (2(b), 3(d)) is called the *computation rule*. When the enabling rule is satisfied for $\epsilon \in T^{\pm} \cup \{(r)|r \in \mathbf{N} - \{0\}\}$, we will say that ϵ is *enabled* at (M, C, ρ) (or (M, C, ρ) *enables* ϵ) and write $(M, C, \rho)[\epsilon\rangle_\gamma$ As we can see, there is no enabling rule for $\epsilon = (r)$ in the case of type L behavior. ϵ is called a *transition* if $\epsilon \in T^{\pm}$, and a *time transition* if $\epsilon \in \{(r)|r \in \mathbf{N} - \{0\}\}$.

In the case of type L behavior, passage of time is allowed in any timed state no matter how large is the time duration (which may exceed transition durations); in the mixed case, time is allowed to pass only if no transition duration will be exceeded. In the case of type A behavior, time may pass if no not-activated transition (i.e., transition in $T - C$) can start at the current state. The strict case simply combines the cases E and A.

Remark 2.1 Unlike Starke's definition [49], the firing of a transition does not change the current time. In order to allow the changing of time, a time transition is introduced in the above definition. Because by firing a transition the time is not changed, sequences of transitions may be applied one by one simulating this way the firing of a step (not necessarily a maximal one). Moreover, in the case A time may pass if no not-activated transitions can start allowing the simulation of the maximal steps.

A T_dPN whose behavior is of type X, where $X \in \{L, E, A, S\}$, is called an *X-timed Petri net* (XT_dPN).

The concepts of "firing sequence" and "reachability" for XT_dPN are introduced as for classical Petri nets:

Definition 2.37 Let γ be an XT_dPN. We say that $\sigma \in (T^{\pm} \cup \{(r)|r \in \mathbf{N} - \{0\})^*$ is a *firing sequence* (from (M, C, ρ)) in γ if there exist a timed state (M', C', ρ') such that $(M, C, \rho)[\sigma\rangle_\gamma(M', C', \rho')$. We call the state (M', C', ρ') *reachable* (from (M, C, ρ)) in γ.

$[(M, C, \rho)\rangle_\gamma$ stands for the set of all reachable timed states from (M, C, ρ) in γ.

Definition 2.38

1. An XT_dPN γ is *bounded with respect to a timed state* (M, C, ρ) if there exists a natural number n such that $M'(s) \leq n$, for every timed state (M', C', ρ') reachable from (M, C, ρ) and every place $s \in S$.

2. γ is called *live with respect to a timed state* (M, C, ρ) if for every timed state (M', C', ρ') reachable from (M, C, ρ) and every transition t, there exists a timed state (M'', C'', ρ'') reachable from (M', C', ρ') which enables t^+.

3. A transition t of γ is called *quasi-live with respect to a timed state* (M, C, ρ) if there exists a timed state (M', C', ρ') reachable from (M, C, ρ) which enables t^+.

4. γ is called *quasi-live with respect to a timed state* (M, C, ρ) if each transition of it is quasi-live with respect to (M, C, ρ).

5. A timed state (M', C', ρ') is called *coverable with respect to a timed state* (M, C, ρ) if there exists a timed state (M'', C'', ρ'') reachable from (M, C, ρ) such that $M' \leq M''$.

Definition 2.39 Let $\gamma = (\Sigma, \delta)$ be an XT_dPN, where $X \in \{L, E, A, S\}$, (M, C, ρ) a state and t a transition of γ.

1. If t is in C, then t is called *active* at (M, C, ρ); otherwise, t is not active at this state.

2. A sequence $\sigma \in (T^{\pm} \cup \{(r)|r \in \mathbf{N} - \{0\}\})^*$ is called an *activating sequence for t at* (M, C, ρ) if t is not active at this state, $t^+ \notin \sigma$, and
$$(M, C, \rho)[\sigma\rangle(M', C', \rho')[t^+\rangle,$$
for some state (M', C', ρ').

3. σ is called a *deactivating sequence for t at* (M, C, ρ) if t is active at this state, $t^- \notin \sigma$, and
$$(M, C, \rho)[\sigma\rangle(M', C', \rho')[t^-\rangle,$$
for some state (M', C', ρ').

Remark 2.2 It is important to note that for L-, E-, or A-timed Petri nets, once a transition got activated there exists a deactivation sequence for the transition. Let (M, C, ρ) be a timed state. In the case of type L behavior any transition $t \in C$ can be deactivated by the sequence $(\rho(t))t^-$. For type E behavior, we repeatedly deactivate (as above) the transition with the minimum residual time in the set of current transitions until t is reached.

As a conclusion, the property $(*)$ below holds true for type L and type E behaviors:
$$(*) \quad (M, C, \rho)[\sigma\rangle(M', \emptyset, \emptyset),$$
for some $M' \geq M$ and some σ consisting of only transitions in C and time transitions.

For the case of type A behavior, things are a little bit more evolved. Given a transition $t \in C$, in order to deactivate it we first activate all transitions in $T - C$ that can be activated (in an arbitrary order), then we apply a time transition to urge the transition t (i.e., to make 0 its time duration in the set of current transitions), and then apply t^-. We emphasize that $(*)$ is not necessarily guaranteed in this case.

In the case of S-timed Petri nets there may be active transitions that cannot be deactivated. For example, if we consider the S-timed Petri net in Figure 2.6, we can easily see that the transition t_1 cannot be deactivated once it was activated.

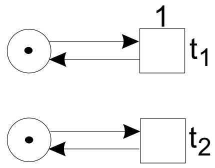


Figure 2.6: Under the type S behavior, t_1 cannot be deactivated once it was activated

Example 2.2 We shall consider the process of releasing a construction authorization. This process can be modeled as in Figure 2.7. The names of the transitions are quite suggestive. First, a person must apply for an authorization. The application is registered. If it is a new application, a tax calculus is performed. After the payment of the taxes, the receipts are registered and the documents are sent for evaluation. A verification of the documents is then performed. If more documents are needed or more taxes has to be payed, a corresponding request is sent. Otherwise, the documentation is processed in order to release the authorization.

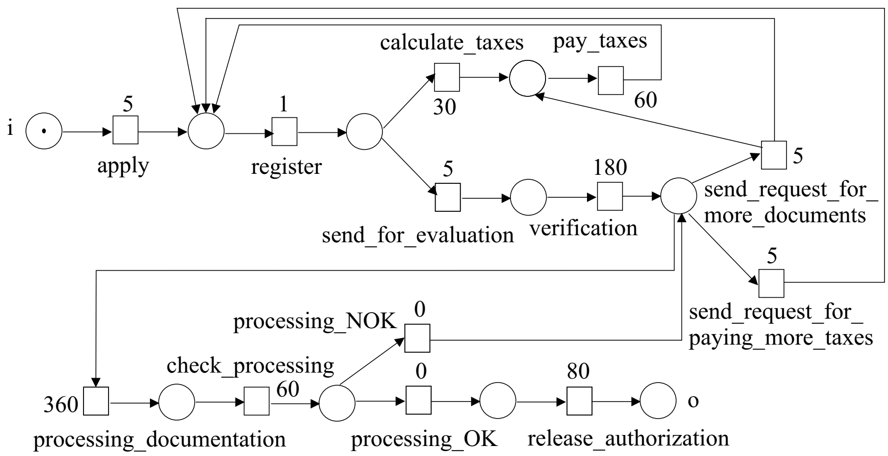


Figure 2.7: Releasing a construction authorisation

Every task in the process above needs an amount of time for completion. For example, the process of processing the documentation takes 360 units of time to complete, but it can be delayed. Therefore, once the preset of the transition `processing_documentation` holds, the processing of the documentation process may start and it will be complete after at least 360 units of time. Thus, we are naturally led to considering LT_dPN in order to model such kinds of processes.

There are situations when the process of releasing an authorization has a deadline and, therefore, the tasks of the process cannot be delayed more than the time needed for their completion. Such cases can be naturally modeled by ET_dPN.

Another constraint usually considered in the process of releasing an authorization is to require for each task to interrupt its current activity while some previous task is sending data to it. Therefore, in this case, time cannot pass as long as there are tasks enabled in the system. AT_dPN are the most suitable to model such processes.

A natural combination of the last two cases leads to ST_dPN.

Lemma 2.4 If an XT_dPN γ is unbounded with respect to a timed state (M_0, C_0, ρ_0), where $X \in \{L, E, A, S\}$, then there are two timed states (M, C, ρ) and (M', C', ρ') reachable from (M_0, C_0, ρ_0) such that $M < M'$ and $C = C'$.

Proof Let $\gamma = (\Sigma, \delta)$ be an unbounded XT_dPN with respect to (M_0, C_0, ρ_0), where $X \in \{L, E, A, S\}$. Then, for any $n \in \mathbf{N}$ there exists a timed state (M, C, ρ) reachable from (M_0, C_0, ρ_0) such that $M(s) > n$ for some place s. Therefore, there exists an infinite sequence of states reachable from (M_0, C_0, ρ_0),

$$(M_1, C_1, \rho_1), (M_2, C_2, \rho_2), \ldots$$

such that

$$M_1 < M_2 < \cdots .$$

As each C_i is a subset of a finite set, it follows that there are i and j such that $i < j$, $M_i < M_j$, and $C_i = C_j$. □

2.2.1 Liveness and Boundedness for LT_dPN

Given an LT_dPN $\gamma = (\Sigma, \delta)$ we associate to it a Petri net Σ', called the *untimed Petri net associated to* γ. This Petri net is obtained by replacing each transition t in Σ, as it is given in Figure 2.8(a), by the structure given in Figure 2.8(b) (we emphasize that the newly added arcs have the weight 1).

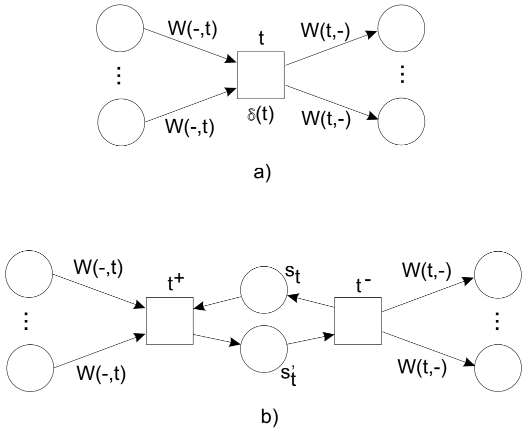


Figure 2.8: Constructing the untimed Petri net for an LT_dPN

Definition 2.40 A *good marking* of Σ' is any marking M of Σ' satisfying $M(s_t) + M(s_t') = 1$, for any transition $t \in T$.

As we can easily see, any reachable marking from a good marking in Σ' is a good marking too. Figure 2.9(a) shows an LT_dPN γ together with a marking M of it, and Figure 2.9(b) shows its associated untimed Petri net Σ' together with a good marking of it.

Let γ be an LT_dPN and Σ' its associated untimed Petri net.

To any timed state (M, C, ρ) of γ we associate a good marking M_C of Σ' by:

$$M_C(s) = \begin{cases} M(s), & \text{if } s \in S \\ 1, & \text{if } s = s_t \text{ for some } t \in T - C \\ 0, & \text{if } s = s_t' \text{ for some } t \in T - C, \\ 0, & \text{if } s = s_t \text{ for some } t \in C \\ 1, & \text{if } s = s_t' \text{ for some } t \in C, \end{cases}$$

for any $s \in S'$.

Conversely, any good marking M' of Σ' can be considered as being associated to some timed state (M, C, ρ) of γ, where $M = M'|_S$, $C = \{t \in T | M'(s_t) = 0\}$, and ρ is an arbitrary function from C into $\mathbf{N}$.

For example, the marking in Figure 2.9(b) is associated to the timed state (M, C, ρ), where M is the marking in Figure 2.9(a), $C = \{t_2\}$, and ρ is a corresponding residual time.

Now, we can prove the following result.

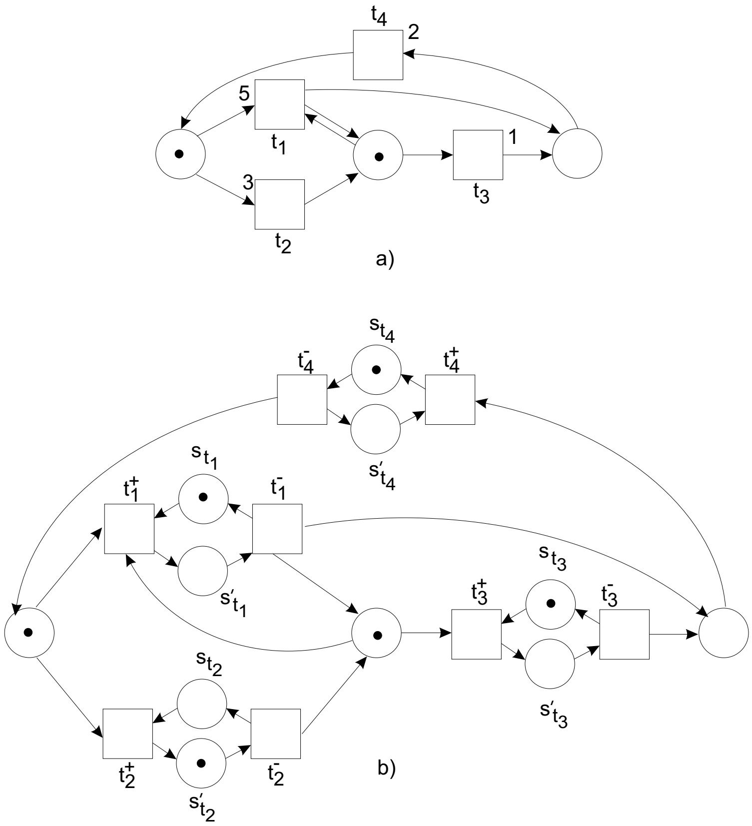


Figure 2.9: Example of untimed Petri net

Theorem 2.8 Let $\gamma = (\Sigma, \delta)$ be an LT_dPN and Σ' be its associated untimed Petri net. Then, the following properties hold:

(1) for any two timed states (M, C, ρ) and (M', C', ρ') of γ and any $\epsilon \in T^{\pm}$, if

$$(M, C, \rho)[\epsilon\rangle_\gamma(M', C', \rho')$$

then $M_C[\epsilon\rangle_{\Sigma'} M'_{C'}$;

(2) for any two good markings M_C and $M'_{C'}$ of γ', any residual times ρ and ρ', and any $\epsilon \in T^{\pm}$, if $M_C[\epsilon\rangle_{\Sigma'} M'_{C'}$ then either $(M, C, \rho)[\epsilon\rangle_\gamma(M', C', \rho')$ or

$$(M, C, \rho)[r\rangle_\gamma(M, C, \rho \dot{-} r)[\epsilon\rangle_\gamma(M', C', \rho')$$

for some $r > 0$.

Proof (1) trivially holds from the definition of Σ'. For (2), we remark first that the property trivially holds if $\epsilon = t^+$ for some transition t. As for $\epsilon = t^-$ we have to take into consideration two cases. If $\rho(t) = 0$, then $(M, C, \rho)[t^-\rangle_\gamma(M', C', \rho')$; otherwise, consider $r = \rho(t)$ and

$$(M, C, \rho)[r\rangle_\gamma(M, C, \rho \dot{-} r)[t^-\rangle_\gamma(M', C', \rho')$$

holds. □

Corollary 2.2 The reachability, boundedness, and liveness properties are all decidable for LT_dPN.

Proof Follows easily from Theorem 2.8 and from the fact that these properties are decidable for Petri nets. □

The type L behavior rule we have adopted does not allow more instances of a same transition to run in the same time because a new instance of a transition t may fire only if no other instance of t is currently running. Formally, t^+ can be applied to a timed state (M, C, ρ) only if $t \notin C$.

2.2.2 Auto-concurrency for LT_dPN

"Auto-concurrency" in a timed Petri net means that a new instance of some transition may be started while other instances of the same transition are still running. For example, the verification of a document in the timed workflow net in Example 2.2 may be started while another document is still under the verification process.

To allow "auto-concurrency", just removing "$t \notin C$" is not enough because C is a set and firing multiple instances of t will add to C only one instance of t. Therefore, in such a case, only one instance of t will be terminated.

In order to allow auto-concurrency, timed states should be defined as triples (M, C, ρ), where:

- M is a marking;
- C is a multiset over T;
- ρ is a function from T into the set of all finite multisets over $\mathbf{N}$ such that $|\rho(t)| = C(t)$ for any $t \in T$.

For example, if exactly two instances of a transition t are currently running, then $C(t)$ should be 2. Each instance has a residual time and, therefore, $\rho(t)$ should be a finite multiset over $\mathbf{N}$ of cardinality 2.

Now, the type L behavior rule can be easily redefined for this case.

Definition 2.41 We write $(M, C, \rho)[\epsilon\rangle_\gamma(M', C', \rho')$ if one of the following cases applies:

1. if $\epsilon = t^+$ then:
   (a) $M[t\rangle_\Sigma$;
   (b) $M'(s) = M(s) - W(s,t)$ for all s, $C'(t) = C(t) + 1$, $C'(t') = C(t')$ for any $t' \neq t$, $\rho'(t) = \rho(t) \cup \{\delta(t)\}$, and $\rho'(t') = \rho(t')$ for any $t' \neq t$;
2. if $\epsilon = t^-$ then:
   (a) $C(t) > 0$ and $0 \in \rho(t)$;
   (b) $M'(s) = M(s) + W(t,s)$ for all s, $C'(t) = C(t) - 1$, $C'(t') = C(t')$ for any $t' \neq t$, $\rho'(t) = \rho(t) - \{0\}$, and $\rho'(t') = \rho(t')$ for any $t' \neq t$;
3. if $\epsilon = (r)$ for some $r \in \mathbf{N} - \{0\}$, then:
   (a) $M' = M$, $C' = C$, and $\rho' = \rho \dot{-} r$.

A T_dPN under the type L behavior rule defined above is called an LT_dPN *with auto-concurrency*.

All the results developed for LT_dPN can be easily rewritten for LT_dPN with auto-concurrency.

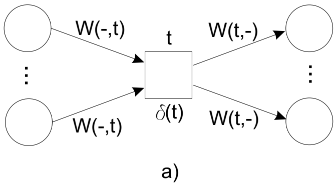
The construction of the untimed Petri nets for a LT_dPN is changed as in Figure 2.10.

Let γ be an LT_dPN with auto-concurrency and Σ' its associated untimed Petri net. To any timed state (M, C, ρ) of γ we associate a marking M_C of Σ' by:

$$M_C(s) = \begin{cases} M(s), & \text{if } s \in S \\ C(t), & \text{if } s = s_t \text{ for some } t \in T, \end{cases}$$

for any $s \in S'$.

Conversely, any marking M' of Σ' can be considered as being associated to some timed state (M, C, ρ) of γ, where $M = M'|_S$, $C(t) = M'(s_t)$, and



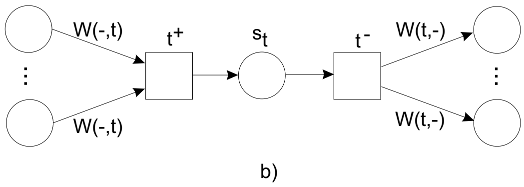


Figure 2.10: Constructing the untimed Petri net for an LT_dPN with auto-concurrency

ρ is an arbitrary function from T into the set of all finite multisets over $\mathbf{N}$ such that $|\rho(t)| = C(t)$, for any transition t.

Now, Theorem 2.8 can be rewritten identically for LT_dPN with auto-concurrency, except for the fact that "good markings" should be replaced by "markings" in the item 2 of the theorem.

Theorem 2.9 Let $\gamma = (\Sigma, \delta)$ be an LT_dPN with auto-concurrency and Σ' be its associated untimed Petri net. Then, the following properties hold:

(1) for any two timed states (M, C, ρ) and (M', C', ρ') of γ and any $\epsilon \in T^{\pm}$, if

$$(M, C, \rho)[\epsilon\rangle_\gamma(M', C', \rho')$$

then $M_C[\epsilon\rangle_{\Sigma'}M'_{C'}$;

(2) for any two markings M_C and $M'_{C'}$ of γ', any residual times ρ and ρ', and any $\epsilon \in T^{\pm}$, if $M_C[\epsilon\rangle_{\Sigma'}M'_{C'}$ then either $(M, C, \rho)[\epsilon\rangle_\gamma(M', C', \rho')$ or

$$(M, C, \rho)[r\rangle_\gamma(M, C, \rho \dot{-} r)[\epsilon\rangle_\gamma(M', C', \rho')$$

for some $r > 0$.

Proof (1) trivially holds from the definition of Σ'. For (2), the property trivially holds if $\epsilon = t^+$ for some transition t. If $\epsilon = t^-$ and $0 \in \rho(t)$, then $(M, C, \rho)[t^-\rangle_\gamma(M', C', \rho')$; otherwise, consider $r \in \rho(t)$ and

$$(M, C, \rho)[r\rangle_\gamma(M, C, \rho \dot{-} r)[t^-\rangle_\gamma(M', C', \rho')$$

holds. □

Corollary 2.3 The reachability, boundedness, and liveness properties are all decidable for LT_dPN with auto-concurrency.

Proof Follows easily from Theorem 2.9 and from the fact that these properties are decidable for Petri nets. □

2.2.3 Liveness and Boundedness for ET_dPN

In this subsection we shall prove that the reachability, liveness, coverability, and boundedness properties are decidable for mET_dPN. In order to do that, we consider *Petri nets with zero-tests on bounded places*, we reduce those properties for mET_dPN to the same properties for Petri nets with zero-tests on bounded places, and we show that the reachability, liveness, coverability, and boundedness problems are decidable for the last nets.

Definition 2.42 A *Petri Nets with zero-tests on bounded places* (PN_{0tb}) is an mPN_{0t} $\gamma = (\Sigma, J, M_0)$ such that

$$(\exists k \geq 0)(\forall M \in [M_0\rangle_\gamma)(\forall s \in S)(s \in pr_1(J) \Rightarrow M(s) \leq k).$$

Proposition 2.3 Reachability and liveness for PN_{0tb} are decidable.

Proof We shall reduce reachability and liveness for PN_{0tb} to the same properties for $\mathcal{L}_3$-conditional Petri nets.

Let $\gamma = (\Sigma, J, M_0)$ be a PN_{b0t}. Without loss of generality we may assume that for any $t \in T$ there exists at most one place $s \in S$ such that $(s, t) \in J$. Moreover, we assume that $|J| = 1$, and let $J = \{(s, t)\}$.

Let k be such that

$$(\forall M \in [M_0\rangle_\gamma)(\forall s \in S)(s \in pr_1(J) \Rightarrow M(s) \leq k).$$

Define the conditional Petri net $\gamma' = (\Sigma, \varphi, M_0)$, where

$$\varphi(t') = \begin{cases} T^*, & \text{if } t' \neq t \\ \{u \in T^* | \Delta u(s) = 0\}, & \text{otherwise} \end{cases}$$

for any transition t'. Because $M(s) \leq k$ for any reachable marking M and $s \in pr_1(J)$, we can easily prove that $\varphi(t)$ is a regular language. Therefore, γ' is an $\mathcal{L}_3$-conditional Petri net. Moreover, it is straightforward to see that the reachability and liveness for γ are equivalent to the reachability and liveness, respectively, for γ'. As these two properties are decidable for $\mathcal{L}_3$-conditional Petri nets, we conclude that they are decidable for PN_{0tb} as well. □

Proposition 2.4 Coverability and boundedness are decidable for PN_{0tb}.

Proof For PN_{0tb}, the coverability tree is finite and can be effectively constructed as for ordinary Petri nets but with the difference that ω will be never used for places in $pr_1(J)$.

Both coverability and boundedness follow by a simple inspection of the coverability tree. □

Now we are in a position to attack the boundedness and liveness properties for E-timed Petri nets.

Given an mET_dPN $\gamma = (\Sigma, \delta, (M_0, \emptyset, \emptyset))$ we associate to it an inhibitor Petri net $\gamma' = (\Sigma', J, M_0')$, called the *untimed Petri net associated to* γ, as follows:

- a transition t_C^+ is associated to each transition t of γ and each subset $C \subseteq T$ which does not contain t (intuitively, t_C^+ says that t is the transition to be activated when the set of current transitions is C);

- a transition t_C^- is associated to each transition t of γ and each subset $C \subseteq T$ which does contain t (intuitively, t_C^- says that t is the transition to be deactivated when the set of current transitions is C);

- a place s_C is associated to each subset $C \subseteq T$ (intuitively, a token in s_C says that C is the set of current transitions);

- a place $s_{\delta(t)}$ is associated to each transition t (intuitively, $s_{\delta(t)}$ will hold $\delta(t)$ tokens when t is activated);

- two places s_{t^+} and s_{t^-} are associated to each transition t (these two places will control the beginning and the end, respectively, of the transition t);

- a transition $time_C$ is associated to each nonempty subset $C \subseteq T$. This transition will simulate the time passage when the set of current transitions is C by removing exactly one token at a firing from each place $s_{\delta(t)}$ with $t \in C$;

- $J = \{(s_{\delta(t)}, t_C^-) | C \subseteq T \ \wedge \ t \in C\}$ is the set of test arcs.

The flow and weight functions are illustrated in Figure 2.11.

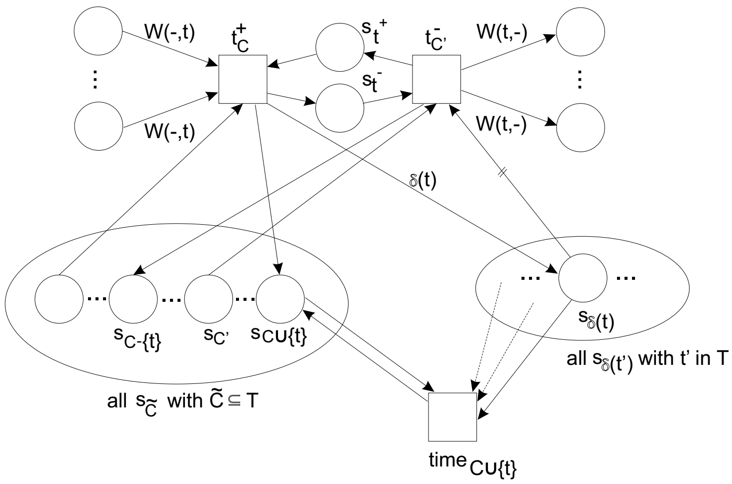


Figure 2.11: Constructing the untimed Petri net for an mET_dPN

The initial marking of γ' is M_0' given by:

$$M_0'(s) = \begin{cases} M_0(s), & \text{if } s \in S \\ 1, & \text{if } s = s_{t^+} \text{ for all } t \in T \\ 0, & \text{if } s = s_{t^-} \text{ for all } t \in T \\ 1, & \text{if } s = s_\emptyset \\ 0, & \text{if } s = s_{\delta(t)} \text{ for all } t \in T \\ 0, & \text{if } s = s_C \text{ for all } C \subseteq T \text{ such that } C \neq \emptyset, \end{cases}$$

for any $s \in S'$.

It is easy to see that the places $s_{\delta(t)}$ can contain at most $\delta(t)$ tokens in any reachable marking (from M'_0); therefore, the associated untimed Petri net γ' is a Petri net with zero tests on bounded places.

To any timed state (M,C,ρ) of γ we associate a marking $M_{(C,\rho)}$ of γ' by:

$$M_{(C,\rho)}(s) = \begin{cases} M(s), & \text{if } s \in S \\ 1, & \text{if } s = s_{t^+} \text{ for some } t \in T - C \\ 0, & \text{if } s = s_{t^-} \text{ for some } t \in T - C \\ 0, & \text{if } s = s_{t^+} \text{ for some } t \in C \\ 1, & \text{if } s = s_{t^-} \text{ for some } t \in C \\ 1, & \text{if } s = s_C \\ 0, & \text{if } s = s_{C'} \text{ for some } C' \neq C \\ \rho(t), & \text{if } s = s_{\delta(t)} \text{ for some } t \in C \\ 0, & \text{if } s = s_{\delta(t)} \text{ for some } t \notin C, \end{cases}$$

for any $s \in S'$.

Conversely, any reachable marking M' of γ' can be considered as being associated to some timed state (M,C,ρ) of γ, where $M = M'|_S$, $C = \{t \in T | M'(s_{t^+}) = 0\}$, and $\rho(t) = M'(s_{\delta(t)})$ for all $t \in C$.

Now, we can prove the following result.

Theorem 2.10 Let $\gamma = (\Sigma, \delta, (M_0, \emptyset, \emptyset))$ be an mET_dPN and γ' be its associated untimed Petri net. Then, the following properties hold:

1. for any two timed states (M,C,ρ) and (M',C',ρ') of γ and any $\epsilon \in T^{\pm} \cup \{(r)|r \in \mathbf{N}\}$, if $(M,C,\rho)[\epsilon\rangle_\gamma(M',C',\rho')$ then
   (a) $M_{(C,\rho)}[\epsilon_C\rangle_{\gamma',i}M'_{(C',\rho')}$, if $\epsilon \in T^{\pm}$, or
   (b) $M_{(C,\rho)}[time^r_C\rangle_{\gamma',i}M'_{(C',\rho')}$, if $\epsilon = (r)$ for some $r \in \mathbf{N}$;
2. for any two markings $M_{(C,\rho)}$ and $M'_{(C',\rho')}$ of γ' and any transition $\epsilon \in T'$, if $M_{(C,\rho)}[\epsilon\rangle_{\gamma',i}M'_{(C',\rho')}$ then
   (a) $(M,C,\rho)[\epsilon\rangle_\gamma(M',C',\rho')$, if $\epsilon \in \{t^+_C, t^-_C\}$, or
   (b) $(M,C,\rho)[(1)\rangle_\gamma(M,C,\rho')$, if $\epsilon = time_C$.

Proof (1a) If $\epsilon = t^+$, then $t \notin C$ and $M[t\rangle_\gamma$. From the definition of $M_{(C,\rho)}$ we have $M_{(C,\rho)}(s_{t^+}) = 1$ and $M_{(C,\rho)}(s_C) = 1$. Therefore, it is easily seen that $M_{(C,\rho)}[t^+_C\rangle_{\gamma',i}M'_{(C',\rho')}$.

If $\epsilon = t^-$, then $t \in C$ and $\rho(t) = 0$. Hence, $M_{(C,\rho)}(s_{\delta(t)}) = 0$ and $M_{(C,\rho)}(s_C) = M_{(C,\rho)}(s_{t^-}) = 1$. Therefore, t_C^- is enabled at the marking $M_{(C,\rho)}$ and its firing yields the new marking $M'_{(C',\rho')}$.

(1b) If $\epsilon = (r)$, then $r \leq min\{\rho(t) \mid t \in C\}$ and $M_{(C,\rho)}(s_{\delta(t)}) = \rho(t)$, for all $t \in C$. Therefore, $M_{(C,\rho)}[time_C^r\rangle_{\gamma',i}M'_{(C',\rho')}$.

(2a) If $\epsilon = t_C^+$, then $M_{(C,\rho)}(s_{t^+}) = 1$. Hence we have $M[t\rangle$, $t \notin C$ and it is easy to see that $(M, C, \rho)[t^+\rangle_\gamma(M', C', \rho')$.

If $\epsilon = t_C^-$, then $M_{(C,\rho)}(s_{t^-}) = 1$ and $M_{(C,\rho)}(s_{\delta(t)}) = 0$. Hence $t \in C$ and $\rho(t) = 0$. Therefore $(M, C, \rho)[t^-\rangle_\gamma(M', C', \rho')$.

(2b) trivially holds from the definition of $M_{(C,\rho)}$. □

Corollary 2.4 The reachability, boundedness, and liveness properties are all decidable for mET_dPN.

Proof Directly from Theorem 2.10, Proposition 2.3, and Proposition 2.4. □

2.2.4 Liveness and Boundedness for ST_dPN

In what follows we shall show that reachability, coverability, boundedness, and quasi-liveness are all undecidable for ST_dPN. The proof technique is by reducing the halting problem for deterministic counter machines to each of these decision problems.

The halting problem of a DCM remains undecidable even if the following constraints are imposed:

- $q_0 \neq q_f$;
- the instruction that is executed in any state q does not return the machine to the initial state q_0, that is:
  - if $I(q, c, q') \in I$, then $q' \neq q_0$;
  - if $I(q, c, q', q'') \in I$, then $q' \neq q_0$ and $q'' \neq q_0$;
- there is a path from q_0 to any state $q \in Q - \{q_0\}$ (a path from a state q to a state $q' \neq q$ is a sequence of states $q_1 = q, q_2, \ldots, q_l = q'$ such that, for any $1 \leq i < l$, $I(q_i, c, q_{i+1}) \in I$ or $I(q_i, c, q_{i+1}, q'') \in I$ or $I(q_i, c, q'', q_{i+1}) \in I$, for some c and q'');

- in the initial configuration all counters hold the number 0.

We shall show how deterministic counter machines as those described above can be simulated by ST_dPN.

Let $A = (Q, q_0, q_f, C, x_0, I)$ be a DCM. We associate to it an S-timed Petri net γ' as follows:

- to each increment instruction $I(q, c, q')$ we associate a structure as in Figure 2.12, where t is a transition which has the time duration 0;

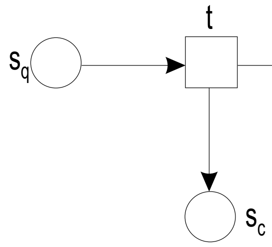


Figure 2.12: The structure associated to $I(q, c, q')$

- to each test instruction $I(q, c, q', q'')$ we associate a structure as in Figure 2.13, where the time durations of all transitions are 0 except the one of t'_2 which is 1.

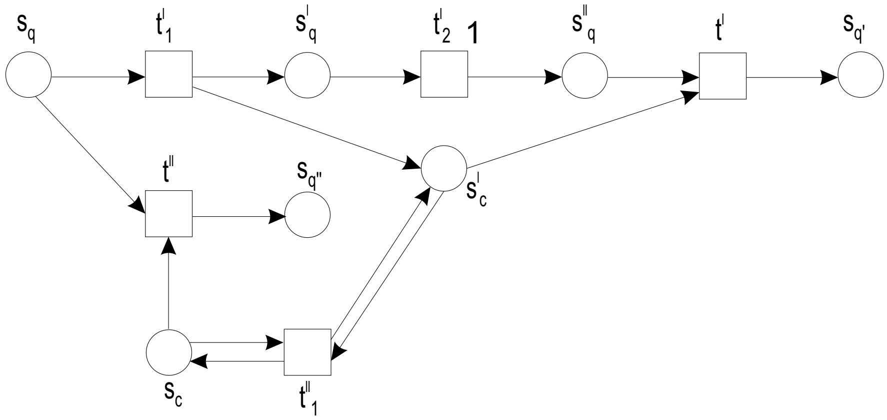


Figure 2.13: The structure associated to $I(q, c, q', q'')$

We shall consider this timed net under type S behavior.

A configuration $\sigma = (q, x)$ of A is simulated in γ by the timed state $(M_\sigma, \emptyset, \emptyset)$ where:

$$M_\sigma(s) = \begin{cases} 1, & \text{if } s = s_q, \\ x(c) & \text{if } s = s_c \text{ for all } c \in C, \\ 0, & \text{otherwise} \end{cases}$$

$(M_{\sigma_0}, \emptyset, \emptyset)$ simulates the initial configuration σ_0 of A.

We call $\gamma = (\Sigma, \delta)$ the ST_dPN *associated to* A.

Lemma 2.5 Let $A = (Q, q_0, q_f, C, x_0, I)$ be a DCM, $\gamma = (\Sigma, \delta)$ be its associated S-timed Petri net, and $\sigma = (q, x)$ and $\sigma' = (q', x')$ be two configurations of A. Then, the following properties hold:

1. if $\sigma \vdash_A \sigma'$ then there exists a firing sequence w in γ such that

$$(M_\sigma, \emptyset, \emptyset)[w\rangle_\gamma (M_{\sigma'}, \emptyset, \emptyset);$$

2. if $(M_\sigma, \emptyset, \emptyset)[w\rangle_\gamma (M_{\sigma'}, \emptyset, \emptyset)$ for some firing sequence w and there is no other intermediate state $(M_{\sigma''}, \emptyset, \emptyset)$ in the firing sequence w, then

$$\sigma \vdash_A \sigma'.$$

Proof (1) If $(q, x) \vdash_A (q', x')$ by an increment instruction $I(q, c, q')$, then $x'(c) = x(c) + 1$ and $x'(c') = x(c')$, for all $c' \neq c$. Then, according to the construction in Figure 2.12, the only transition that can begin its firing in γ is t. Because $\delta(t) = 0$, the transition t must complete its firing at the same time removing the token from s_q and adding a token in both $s_{q'}$ and s_c. The new yielded marking is exactly $M_{\sigma'}$. Thus, $(M_\sigma, \emptyset, \emptyset)[t^+ t^-\rangle (M_{\sigma'}, \emptyset, \emptyset)$.

If $(q, x) \vdash_A (q', x')$ by a test instruction $I(q, c, q_1, q_2)$ and $x(c) = 0$, then $q' = q_1$ and $x'(c) = x(c)$, for all counters c. As in the marking M_σ of γ there is only one token in s_q and no token in s_c, according to the structure in Figure 2.13 the only transition which can begin its firing is t'_1. Thus, $(M_\sigma, \emptyset, \emptyset)[{t'}_1^+ {t'}_1^- {t'}_2^+ (1) {t'}_2^- t'^+ t'^-\rangle (M_{\sigma'}, \emptyset, \emptyset)$.

If $(q, x) \vdash_A (q', x')$ by a test instruction $I(q, c, q_1, q_2)$ and $x(c) > 0$, then $q' = q_2$, $x'(c) = x(c) - 1$, and $x'(c') = x(c')$, for all counters $c' \neq c$. In the marking M_σ of γ there is one token in s_q and $x(c)$ tokens in s_c. Thus, the only transitions enabled at the state $(M_\sigma, \emptyset, \emptyset)$ are t'_1 and t''. We can fire t''^{+} and, because its duration is 0, it must complete its firing by applying t''^{-} which yields a state whose marking is $M_{\sigma'}$. Thus, in γ we have $(M_\sigma, \emptyset, \emptyset)[t''^{+}t''^{-}\rangle(M_{\sigma'}, \emptyset, \emptyset)$.

(2) In the state $(M_\sigma, \emptyset, \emptyset)$ there is one token in the place s_q and $x(c)$ tokens in each counter c. If s_q is connected to a transition t as in Figure 2.12, then the instruction associated to the state q is an increment one $I(q, c, q')$ and the next state reachable from $(M_\sigma, \emptyset, \emptyset)$ and whose set of current transitions is empty is obtained by firing the sequence $w = t^{+}t^{-}$. The marking in this state is $M_{\sigma'}$ and it corresponds to the configuration σ' obtained by executing the instruction $I(q, c, q')$ in A.

If s_q has two output transitions t'_1 and t'', then the instruction associated to the state q is a test instruction $I(q, c, q_1, q_2)$. In order to reach the next timed state whose set of current transitions is empty we can apply two firing sequences $w = t''^{+}t''^{-}$ or $w = t'^{+}_1 t'^{-}_1 t'^{+}_2(1)t'^{-}_2 t'^{+}t'^{-}$. The first one corresponds to the situation in which the counter c is 0 and the second one to the situation in which $x(c) > 0$. In both cases, the marking $M_{\sigma'}$ reached by applying w at $(M_\sigma, \emptyset, \emptyset)$ is associated to σ' reached by executing the instruction $I(q, c, q_1, q_2)$ in A.

The two firing sequences above are the only ones possible in the marking $(M_\sigma, \emptyset, \emptyset)$ because, if there is at least a token in s_c and t'^{+}_1 fires, then t'^{-}_1 must apply as there are no other enabled transitions. The firing of t'^{-}_1 adds a token in s'_c enabling two new transitions t'^{+}_2 and t''^{+}_1. If t'^{+}_2 fires, as the duration of t'_2 is 1 and γ works under type S behavior, t''^{+}_1 must be infinitely fired. Thus, the transition t'_2 could not complete its firing. □

We define now three new S-timed Petri nets obtained from the S-timed Petri net γ associated to a DCM A:

- $\gamma_1 = (\Sigma_1, \delta_1, (M^1_{\sigma_0}, \emptyset, \emptyset))$ is obtained from γ as described in Figure 2.14, where $t_1, \ldots, t_r$ are new transitions with time duration 0. The initial marking $M^1_{\sigma_0}$ marks all the places as M_{σ_0} does.

- $\gamma_2 = (\Sigma_2, \delta_2, (M^2_{\sigma_0}, \emptyset, \emptyset))$ is obtained from γ as described in Figure 2.15, where $\tilde{s}$ is a new place. The initial marking $M^2_{\sigma_0}$ marks $\tilde{s}$ by zero tokens, and all the other places as M_{σ_0} does.

- $\gamma_3 = (\Sigma_3, \delta_3, (M^3_{\sigma_0}, \emptyset, \emptyset))$ is obtained from γ as described in Figure 2.16, where $\tilde{t}$ is a new transition with time duration 0. The initial marking $M^3_{\sigma_0}$ marks all the places as M_{σ_0} does.

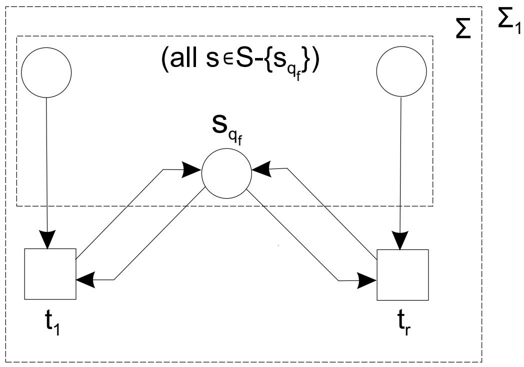


Figure 2.14: Undecidability of coverability for ST_dPN

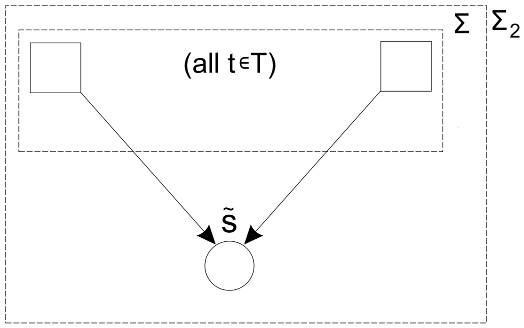


Figure 2.15: Undecidability of boundedness for ST_dPN

Now, we obtain the following results:

Theorem 2.11 Let γ be the ST_dPN associated to a DCM A and γ_i, $i = 1, 2, 3$, be the ST_dPNs constructed as above. The following properties hold:

1. a timed state $(M_\sigma, \emptyset, \emptyset)$ such that $M_\sigma(s_{q_f}) = 1$ is reachable in γ iff A halts;

2. the timed state $(M_\sigma, \emptyset, \emptyset)$, where

$$M_\sigma(s) = \begin{cases} 1, & s = s_{q_f} \\ 0, & \text{otherwise}, \end{cases}$$

   is coverable in γ_1 iff A halts;

3. γ_2 is bounded iff A halts;

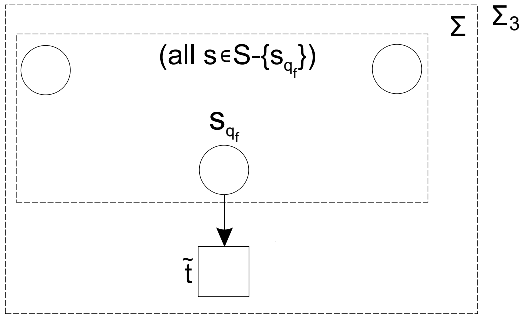


Figure 2.16: Undecidability of quasi-liveness for ST_dPN

4. $\tilde{t}$ in γ_3 is quasi-live iff A halts.

Proof (1) If A halts then $(q_0, x_0) \vdash_A (q_f, x)$. According to Lemma 2.5, the timed state $(M_\sigma, \emptyset, \emptyset)$ such that $M_\sigma(s_{q_f}) = 1$ is reached in γ.

If $(M_\sigma, \emptyset, \emptyset)$ is reachable in γ then, there exists a firing sequence w such that $(M_{\sigma_0}, \emptyset, \emptyset)[w\rangle_\gamma(M_\sigma, \emptyset, \emptyset)$. According to Lemma 2.5, in A we have $(q_0, x_0) \vdash_A (q_f, x)$ meaning that A halts.

(2) If A halts, according to Lemma 2.5, the state $(M_{\sigma'}, \emptyset, \emptyset)$ such that $M_{\sigma'}(s_{q_f}) = 1$ and $M_{\sigma'}(s) \geq 0$, for some $s \in S$, is reachable in γ. $(M_{\sigma'}, \emptyset, \emptyset)$ is reachable in γ_1 as well and we have $M_\sigma \leq M_{\sigma'}$.

If $(M_\sigma, \emptyset, \emptyset)$ is coverable in γ_1 then there exists a reachable timed state (M, C, ρ) such that $M_\sigma \leq M$. Thus, $M(s_{q_f}) = 1$. This timed state is obtained from a reachable timed state $(M_{\sigma'}, \emptyset, \emptyset)$ of γ by applying some t_i, $i = 1, \cdots, r$. Therefore, $M_{\sigma'}(s_{q_f}) = 1$ and A halts.

(3) If A halts then the final state q_f is reached after applying a finite numbers of instructions. Thus, in every reachable marking of γ_2 each place will hold a finite numbers of tokens. Therefore, γ_2 is bounded.

If γ_2 is bounded and we assume that A does not halt, then we can apply an infinite sequence of instructions form the initial state. According to Lemma 2.5, in γ_2 an infinite sequence of transitions is firing, each transition adding a token in $\tilde{s}$ contradicting the fact that γ_2 is bounded.

(4) If A halts then in γ_3 a state whose marking has a token in s_{q_f} is reachable from the initial state. $\tilde{t}^+$ can be fired at this state.

If $\tilde{t}$ is quasi-live in γ_3 then there exists a reachable timed state which enables $\tilde{t}^+$. The marking of it must have a token in s_{q_f} and the only timed state which satisfy that condition is $(M_\sigma, \emptyset, \emptyset)$, where $\sigma = (q_f, x)$. This timed state is reachable in γ as well and according to Lemma 2.5 we have $(q_0, x_0) \vdash_A (q_f, x)$ in A. Therefore, A halts. □

Corollary 2.5 The reachability, coverability, boundedness, and quasi-liveness problems are all undecidable for S-timed Petri nets.

2.2.5 Liveness and Boundedness for AT_dPN

In what follows we shall show that reachability, coverability, boundedness, and quasi-liveness are all undecidable for AT_dPN. The proof technique is similar to the one for S-timed Petri nets, except for the fact that the simulation of counter machines is a little bit more subtle.

In this case, an increment instruction $I(q, c, q')$ is simulated by a structure like the one in Figure 2.17, where t has time duration 0, $x = 1$ if $q' = q_f$ or the instruction in q' is an increment one, and $x = 2$ if the instruction in q' is a test one.

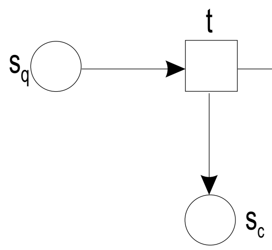


Figure 2.17: The structure associated to $I(q, c, q')$

A test instruction $I(q, c, q', q'')$ is simulated by a structure like the one in Figure 2.18, where the durations of all transitions are 0 except the one of t'_1 which is 1, and the x's on those two arcs have a similar meaning as the one in Figure 2.17.

We shall consider this timed Petri net under the A behavior.

A configuration $\sigma = (q, x)$ of A is simulated by the timed state $(M_\sigma, \emptyset, \emptyset)$, where:

$$M_\sigma(s) = \begin{cases} 1, & \text{if } s = s_q \text{ and } q = q_f \text{ or there is an increment instruction in } q \\ 2, & \text{if } s = s_q \text{ and there is a test instruction in } q \\ x(c) & \text{if } s = s_c \text{ for all } c \in C, \\ 0, & \text{otherwise} \end{cases}$$

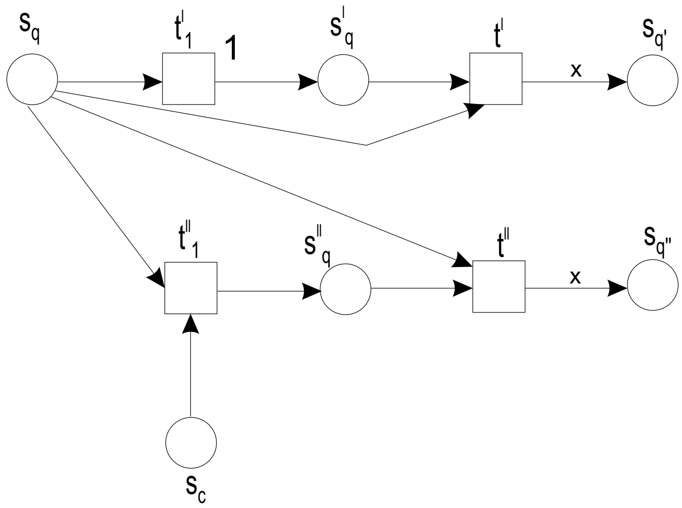


Figure 2.18: The structure associated to $I(q, c, q', q'')$. The place s_q will be marked by two tokens

$(M_{\sigma_0}, \emptyset, \emptyset)$ simulates the initial configuration σ_0 of A.
$\gamma = (\Sigma, \delta)$ is called the AT_dPN *associated to* A.
Now, we can prove a similar lemma as for S-timed Petri nets.

Lemma 2.6 Let $A = (Q, q_0, q_f, C, x_0, I)$ be a DCM, $\gamma = (\Sigma, \delta)$ be its associated A-timed Petri net, and $\sigma = (q, x)$ and $\sigma' = (q', x')$ be two configurations of A. Then, the following properties hold:

1. if $\sigma \vdash_A \sigma'$ then there exists a firing sequence w in γ such that

$$(M_\sigma, \emptyset, \emptyset)[w\rangle_\gamma (M_{\sigma'}, \emptyset, \emptyset);$$

2. if $(M_\sigma, \emptyset, \emptyset)[w\rangle_\gamma (M_{\sigma'}, \emptyset, \emptyset)$ for some firing sequence w and there is no other intermediate state $(M_{\sigma''}, \emptyset, \emptyset)$ in the firing sequence w, then

$$\sigma \vdash_A \sigma'.$$

Proof (1) If $(q, x) \vdash_A (q', x')$ by an increment instruction $I(q, c, q')$, then in γ M_σ has only one token in the place s_q and $x(c)$ tokens in the places s_c, for all counters c. According to the construction in Figure 2.17, the only transition that can begin its firing in γ is t. Because $\delta(t) = 0$, the transition

t must complete its firing at the same time removing the token form s_q and adding a token in both $s_{q'}$ and s_c. Thus, $(M_\sigma, \emptyset, \emptyset)[t^+t^-\rangle_\gamma(M'_\sigma, \emptyset, \emptyset)$.

If $(q, x) \vdash_A (q', x')$ by a test instruction $I(q, c, q_1, q_2)$ and $x(c) = 0$ then $q' = q_1$ and $x'(c) = x(c)$, for all counters c. In γ the only transition enabled at M_σ is t'_1. Thus, we have $(M_\sigma, \emptyset, \emptyset)[{t'}_1^+(1){t'}_1^-t'^+t'^-\rangle_\gamma(M_{\sigma'}, \emptyset, \emptyset)$.

If $(q, x) \vdash_A (q', x')$ by a test instruction $I(q, c, q_1, q_2)$ and $x(c) > 0$ then in γ, at $(M_\sigma, \emptyset, \emptyset)$, we can apply the sequence ${t''}_1^+{t''}_1^-t''^+t''^-$ which yields the new timed state $(M_{\sigma'}, \emptyset, \emptyset)$.

(2) If in the state $(M_\sigma, \emptyset, \emptyset)$ there is one token in the place s_q then, the instruction applied in A is an increment one $I(q, c, q')$ and in γ we have $(M_\sigma, \emptyset, \emptyset)[t^+t^-\rangle_\gamma(M_{\sigma'}, \emptyset, \emptyset)$. The timed state $(M_{\sigma'}, \emptyset, \emptyset)$ corresponds to the configuration σ' obtained by executing the instruction $I(q, c, q')$ in A.

If there are two tokens in the place s_q, then the instruction applied in A is a test instruction $I(q, c, q_1, q_2)$. In order to reach the next timed state whose set of current transitions is empty, we can apply two firing sequences $w = {t''}_1^+{t''}_1^-t''^+t''^-$ or $w = {t'}_1^+(\epsilon){t'}_1^-t'^+t'^-$, where $\epsilon \in \mathbf{N}^+$. The first one corresponds to the situation $x(c) = 0$ and the second one to the situation $x(c) > 0$. In both cases, the marking $M_{\sigma'}$ reached by applying w at $(M_\sigma, \emptyset, \emptyset)$ is associated to σ' reached by executing the instruction $I(q, c, q_1, q_2)$ in A.

The two firing sequences above are the only ones possible in the marking $(M_\sigma, \emptyset, \emptyset)$ because, if ${t'}_1^+$ fires first or ${t''}_1^+$ fires first but it is immediately followed by ${t'}_1^+$, then no matter how the other transitions are applied, the places s_{q_1} and s_{q_2} will be never marked. Thus, no next state can be reached.

□

By a similar construction as the one in the previous subsection one can easily obtain that the reachability, coverability, boundedness, and quasi-liveness problems are all undecidable for A-timed Petri nets.

Chapter 3

Workflow Net Theory

In this chapter we shall present the basic concepts on classical workflow nets, the two approaches considered in characterizing the soundness property of a workflow net, and the complexity of the algorithms used to decide this property.

3.1 Workflow Management Systems

In the last years the *Workflow Management Coalition* made a concerned effort for standardization, specification, and analysis of workflow management systems [58]. In spite of this effort the field still lacks precise definitions for some of its concepts. A *workflow management system* is defined by the Workflow Management Coalition as follows [59]:

> A system that defines, creates and manages the execution of workflows through the use of software running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.

Workflow management systems give concrete form to the essential concepts, techniques and methods for workflow management. *Workflow management* supports business processes in organizations and involves managing the flow of work through an organization. A *workflow* is a representation of a given process that consist of well-defined set of activities refered as *tasks*. Workflows are *case-based*, that is, every piece of work is executed for a specific *case*.

Each of the tasks in the process represented by a workflow serves a given function, has some information input requirements and may generate information as part of its output. The tasks in a workflow are executed in a specific order. Because of this order, the *conditions* which correspond to causal dependencies between tasks must be identified. A condition hold or does not hold. Each task has *preconditions* which should hold before the task is executed and *postconditions* which should hold after the task is executed. Some tasks has to be executed for many cases. A task which need to be executed for a specific case is called a *work item*. Most work items are executed by a *resource*. A resource can be a human, a device, or a program. A *resource class* is a group of resources with similar characteristics. A resource class which is based on functional requirements is called a *role*.

A workflow has three dimensions [2]:

1. the case dimension,

2. the process dimension, and

3. the resource dimension.

The case dimension means that all cases are handled individually. In the process dimension the tasks and the routing along this tasks are specified. In the resource dimension, the resources are grouped into roles and organizational units.

It is very important to relate workflows with well established models of computation. Two formalisms for modeling workflows have been suggested.

One formalism for workflow modeling is based on workflow graphs [47, 48]. They provide a more direct way of modeling workflows and are based on directed acyclic graphs, with two types of nodes, tasks and conditions. The graph has one node with no incoming flows, called the initial node, and one node with no outgoing flows, called the final node. Workflow graphs can be used to identify structural conflicts in process models, such as deadlock and lack of synchronization.

Another approach for workflow modeling is based on Petri nets (workflow nets). Van der Aalst identifies three main reasons for using Petri nets for workflow modeling and specification [2]:

- Petri nets possess a formal semantics and an intuitive graphical representation;

- Petri nets can explicitly model states and a clear distinction can be made between the enabling and execution of a task;
- the abundance of available and theoretically proved analysis techniques.

3.2 Workflow Nets

Modeling a workflow process definition by Petri nets is straightforward: *tasks* are modeled by transitions, *conditions* are modeled by places and *cases* are modeled by tokens. Two requirements should be satisfied by a Petri net that model a workflow system. The first one require that a workflow net should have two distinguished places: an *input* place i and an *output* place o. A token in i corresponds to a case which needs to be handled and a token in o corresponds to a case which has been handled. The second requirements specifies that every transition and place should contribute to the processing of cases which means that every transition and place should be on a path from place i to place o.

Definition 3.1 [1] A *workflow net* (*WN*) is a Petri net $\Sigma = (S, T, F)$ with the following properties:

1. Σ has two special places i and o. The place i satisfies $^\bullet i = \emptyset$ and it is called the *input place* of Σ, and the place o satisfies $o^\bullet = \emptyset$ and it is called the *output place* of Σ;
2. every node $x \in P \cup T$ in the graph of Σ is on a path from i to o.

In the rest of this chapter we will consider Petri nets Σ such that, for all $(x, y) \in F$ we have $W(x, y) \leq 1$.

The routing of cases is a main issue for the first two dimensions of a workflow net [3]. In [59] four types of routing are identified: *sequential*, *parallel*, *conditional* and *iteration*.

Sequential routing is used to deal with causal relationships between tasks. In Figure 3.1 tasks A and B are executed sequentially: task B is executed after the completion of task A. Place s_2 models the causal relationship between tasks A and B.

In a *parallel routing* two ore more tasks are executed in parallel. Parallel routing normally commences with an AND-split and conclude with an AND-join. In Figure 3.2 the execution of AND-split A enables both tasks B and

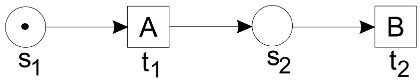


Figure 3.1: Sequential routing

C and AND-join D is enabled after the execution of both tasks B and C, that is, D is used to synchronize two subflows.

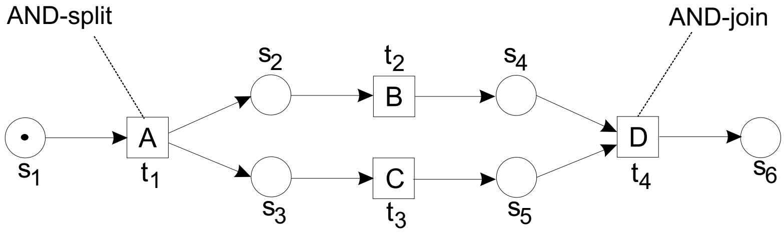


Figure 3.2: Parallel routing

Conditional routing is used in situations where a routing of a case may depend on the workflow attributes, the behavior of the environment, or the workload of the organization. Two building blocks are used to model a choice between two or more alternatives: OR-split and OR-join. In Figure 3.3 a choice is made between B and C, and the execution of one of these two tasks is followed by the execution of D.

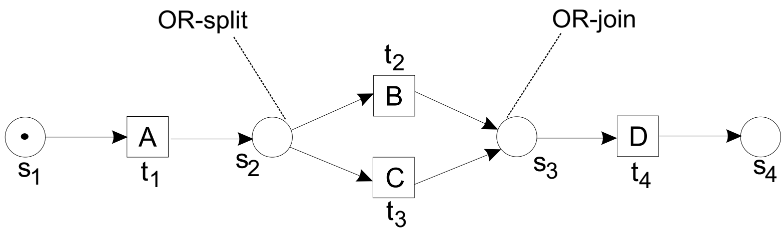


Figure 3.3: Conditional routing

A *iteration* is a cycle involving the repetitive execution of one or more tasks. In Figure 3.4 tasks B and C may execute several times. Iteration is often considered to be an undesirable form of routing because it corresponds to the repetitive execution of the same task without making any real progress. There are situations in real systems where iteration cannot be avoided. For example, in Example 2.2, the documentation supplied by a person is incomplete or more taxes has to be payed.

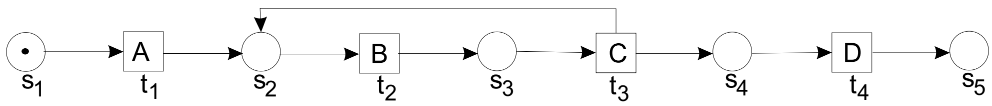


Figure 3.4: Conditional routing

The correctness, effectiveness, and efficiency of the business process supported by the workflow management system are vital to the organization. In the sequel, we shall focus on the verification of workflow nets' correctness.

3.3 Soundness Property

One of the most important correctness properties of a workflow is *proper termination* which means that starting form the initial marking (with only one token in the initial place) it is always possible to reach the marking with only one token in the final place. This concept have been met since 70's in [14] and [38]. In [40] a Petri net is said to be properly terminating if it always terminates in a well-defined manner such that no tokens are left in the net. A system is guaranteed to function in a well behaved manner without any side effects on the next initiation if the Petri net that model the system satisfies is properly terminating.

There is one more requirement which says that there should be *no dead transitions*. These two properties defines the correctness criterion called *soundness*. This concept was proposed in [1] by Wil van der Aalst in the context of modeling workflow systems by Petri nets and it gained much attention both from the workflow management community and the Petri net community. In [1] the definition of soundness requires another property to be satisfied: the state containing only one token in the final place should be unique. As we shall see, this third property can be derived from the other two.

Definition 3.2 A *WN* Σ is called *sound* if it satisfies the following properties:

1. $M_o \in [M\rangle$, for any M reachable from M_i;

2. $(\forall t \in T)(\exists M \in [M_i\rangle)(M[t\rangle)$.

Remark 3.1 The third property in the definition of soundness in [1] is:

$$(\forall M \in [M_i\rangle)(M(o) \geq 1 \Rightarrow M = M_o).$$

As it was shown in [16], this property can be derived from Definition 3.2 (1) and (2). Indeed, let M be a reachable marking such that $M(o) \geq 1$. If we assume that $M_o = 1$ and $M(s) \geq 1$ for some places $s \neq o$, or $M(o) > 1$, then the marking M_o would be never reachable from M because each transition in Σ has at least an output arc and $o^\bullet = \emptyset$.

The concept of soundness is closely related to the one of "home marking".

Definition 3.3 Let $\gamma = (\Sigma, M_0)$ be a marked Petri net. A marking M of γ is called a *home marking* if $M \in [M'\rangle$ for all $M' \in [M_0\rangle$.

It is clear that the first property from Definition 3.2 can be replaced by:

"M_o is a home marking of γ"

Along the years, two major approaches for deciding soundness have been proposed. The first one, due to van der Aalst, reduce the soundness problem to the liveness and boundedness of a Petri net associated to the workflow net for which the soundness property is studied. The second approach reduce the soundness problem to the "home marking" problem. In what follows we shall discuss these two approaches.

3.3.1 The closure of a Workflow Net

Definition 3.4 [1] A Petri net $\overline{\Sigma}$ is called a *closure* of an *WN* Σ, if $\overline{\Sigma}$ is obtained from Σ just by adding a new transition t^* and two arcs (o, t^*) and (t^*, i).

As two closures of the same *WN* Σ differs only by the "name" of the new added transition, we may say that the closure is unique and always denote the new transition by t^*.

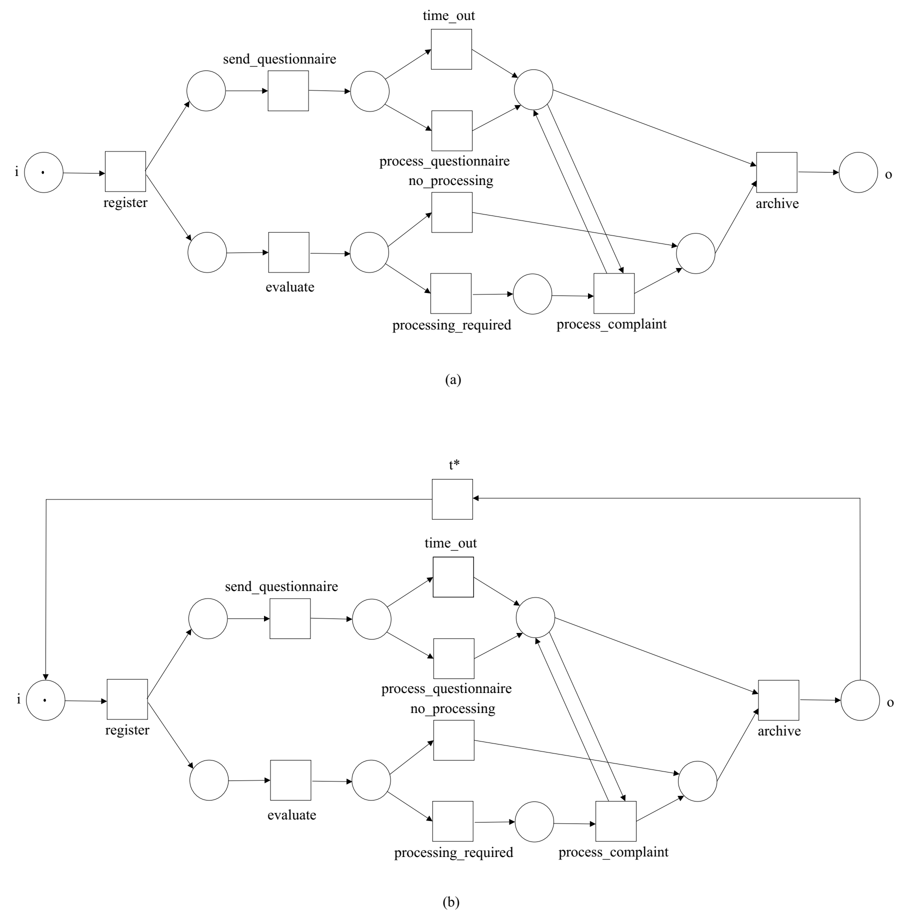


Figure 3.5: The workflow net for processing of complaints (a) and its closure (b)

Example 3.1 Let us consider the example for processing of complaints. First the complaint is registered, then in parallel a questionnaire is sent to the complainant and the complaint is evaluated. If the complainant returns the questionnaire within a week, the task process questionnaire is executed. If the questionnaire is not returned within a week, the result of the questionnaire is discarded (task time-out). Based on the result of the evaluation, the complaint is processed or not. The actual processing of the complaint is delayed until the questionnaire is processed or a time-out has occurred. Finally, task archive is executed. Figure 3.5 (a) shows the workflow net which models the processing of complaints and Figure 3.5 (b) shows the closure of this workflow net.

Now we can characterize the soundness property of a workflow net by its closure as follows:

Theorem 3.1 [1] A workflow net Σ is sound if and only if its closure $\overline{\Sigma}$ is live and bounded.

Proof Assume that $\overline{\Sigma}$ is live and bounded with respect to M_i. We consider a marking M reachable from M_i in Σ. Clearly, M is also reachable in $\overline{\Sigma}$. Since $\overline{\Sigma}$ is live there exists a marking M' reachable form M such that $M'[t^*\rangle_{\overline{\Sigma}}$.

Obviously, $M' \geq M_o$. If we assume that $M' > M_o$ then, by firing t^* a marking M'' which satisfies $M'' > M_i$ is obtained contradicting the fact that $\overline{\Sigma}$ is bounded. Therefore, Definition 3.2(1) holds true. The fact that Definition 3.2(2) holds true follows directly from the liveness property of $\overline{\Sigma}$.

Conversely, assume that Σ is sound and. We can easily see that

$$[M_i\rangle_\Sigma = [M_i\rangle_{\overline{\Sigma}}$$

because firing of t^* in $\overline{\Sigma}$ returns the net to the marking M_i. Hence, to show that $\overline{\Sigma}$ is bounded with respect to M_i it suffices to show that Σ is bounded with respect to M_i.

Now, let us assume that γ is not bounded with respect to M_i. Then, there are two markings M and M' reachable from M_i such that $M' > M$ [35]. Since Σ is sound there exists a firing sequence σ such that $M[\sigma\rangle_\Sigma M_o$. Since $M' > M$ we have $M'[\sigma\rangle_\Sigma M''$ and $M'' > M_o$ contradicting the fact that Σ is sound.

Let us prove now that $\overline{\Sigma}$ is live with respect to M_i. Let M be a marking reachable from M_i in $\overline{\Sigma}$ and t be an arbitrary transition. M is reachable from M_i in Σ as well, and since Σ is sound we have that M_o is reachable from M in Σ. This shows that M_o is reachable from M in $\overline{\Sigma}$ too.

By $M_o[t^*\rangle_{\overline{\Sigma}} M_i$ and by the property in Definition 3.2(2), there exists M' reachable from M_i (both in Σ and in $\overline{\Sigma}$) which enables t. Therefore, $\overline{\Sigma}$ is live. □

The verification of the soundness property is reduced to checking whether its closure is live and bounded. Therefore, standard Petri nets based analysis tools can be used to decide soundness.

Most of the workflow management systems abstract from states between tasks (states are not represented). Because of that, every choice is made inside an OR-split building block. In a Petri net such a block corresponds to a number of transitions sharing the same set of input places. This means that for these workflow management systems, a workflow procedure corresponds to a free-choice Petri net. Therefore, it make sense to consider free-choice Petri nets.

Definition 3.5 [1] A Petri net Σ is called *free-choice* if for every two transitions t_1 and t_2 ${}^\bullet t_1 \cap^\bullet t_2 \neq \emptyset$ implies ${}^\bullet t_1 =^\bullet t_2$.

There are efficient algorithms to decide boundedness and liveness for free-choice Petri nets, hence to decide soundness of free-choice workflow nets. Moreover, a sound free-choice workflow net is guaranteed to be safe.

Definition 3.6 Let Σ be a Petri net. Σ is called *well-formed* if there exists a marking M such that (Σ, M) is live and bounded.

Lemma 3.1 [1] A sound free-choice workflow net is safe.

Proof Let Σ be a sound free-choice workflow net and $\overline{\Sigma}$ its closure. $\overline{\Sigma}$ is free-choice and well-formed. Hence, $\overline{\Sigma}$ is covered by S-components [8]. Since $(\overline{\Sigma}, M_i)$ is free-choice, live, and bounded, as shown in [8], the bound of each place $s \in S$ is

$$min\{\sum_{s' \in S'} M_i(s') \mid (S', T', F') \text{ is an } S\text{-component of } \overline{\Sigma} \text{ containing } s\} = 1.$$

Therefore, $\overline{\Sigma}$ is safe and so is Σ. $\square$

Safeness is a desirable property because it makes no sense to have multiple tokens in a place representing a condition. A condition is either true (1 token) or false (no tokens).

Although most workflow management systems only allow for free-choice workflows, free-choice workflow nets are not a completely satisfactory structural characterization of "good" workflows. If a workflow can be modeled as a free-choice workflow net it should be done so because for free-choice workflow nets there are efficient analysis techniques.

Another approach to obtain a structural characterization of "good" workflows, is to balance AND/OR-splits and AND/OR-joins. Two parallel flows initiated by an AND-split should not be joined by an OR-join and two alternative created by an OR-split should not be synchronized by an AND-join. We shall formalize the concept in the following definition.

Definition 3.7 1. A Petri net Σ is *well-handled* if for any pair of nodes x and y such that one of the nodes is a place and the other a transition and for any pair of elementary paths c_1 and c_2 leading from x to y, $\alpha(c_1) \cap \alpha(c_2) = \{x, y\} \Rightarrow c_1 = c_2$.

2. A Petri net Σ is *well-structured* if its closure $\overline{\Sigma}$ is well-handled.

Well-handled Petri nets and well-structured workflow nets have a number of nice properties.

Definition 3.8 Let $\Sigma = (S, T, F)$ be a Petri net and Σ' a partial subnet of it. An elementary path $c = x_1, \ldots, x_n$ is a *handle* of Σ' (or Σ' *has a handle* c) if $c \cap (S' \cup T') = \{x_1, x_n\}$.

We classify the handles according to the the nature of their first and last nodes as SS-, ST-, TS-, and TT-handles.

Definition 3.9 Let Σ be a Petri net.

1. Σ is *structurally bounded* if (Σ, M) is bounded for all markings M of Σ.
2. Σ is *structurally live* if there exists a marking M of Σ such that (Σ, M) is live.

Lemma 3.2 [1] A strongly connected well-handled Petri net is well-formed.

Proof Let Σ be a strongly connected well-handled Petri net. Clearly, there are no circuits that have ST-handles nor TS-handles. Therefore, the net is structurally bounded and structurally live [12]. Hence, it is well-formed. $\square$

Moreover, there are efficient algorithms to verify soundness for well-structured workflow nets.

Definition 3.10 A Petri net Σ is *elementary extended non self-controlling* if for every pair of transitions t_1 and t_2 such that ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$, there does not exist an elementary path c leading from t_1 to t_2 such that ${}^\bullet t_1 \cap \alpha(c) = \emptyset$.

Lemma 3.3 [1] Let Σ be a workflow net. If Σ is well-structured, then $\overline{\Sigma}$ is elementary extended non-self controlling.

Proof Assume that $\overline{\Sigma}$ is not elementary extended non self-controlling. This means that there is a pair of transitions t_1 and t_k such that $^\bullet t_1 \cap {^\bullet t_k} \neq \emptyset$ and there exist an elementary path $c = t_1, p_2, t_2, \ldots, p_k, t_k$ leading from t_1 to t_k and $^\bullet t_1 \cap \alpha(c) = \emptyset$. Let $p_1 \in {^\bullet t_1} \cap {^\bullet t_k}$. $c_1 = p_1, t_k$ and $c_2 = p_1, t_1, p_2, t_2, \ldots, p_k, t_k$ are paths leading from p_1 to t_k. Clearly, c_1 is elementary. Since c is elementary and $p_1 \notin \alpha(c)$ because $p_1 \in {^\bullet t_1}$, c_2 is also elementary. As c_1 and c_2 are both elementary paths, $c_1 \neq c_2$, and $\alpha(c_1) \cap \alpha(c_2) = \{p_1, t_k\}$ we conclude that $\overline{\Sigma}$ is not well-handled. $\square$

Lemma 3.4 [1] A sound well-structured workflow net is safe.

Proof Let Σ be sound and well-structured and $\overline{\Sigma}$ be its closure. $\overline{\Sigma}$ is non-self controlling and it is covered by S-components [5]. i is a node of any S-component (S', T', F', M_i). Since each S-component is strongly connected and M_i marks the place i with a token, each S-component is live [8]. Therefore, each S-component (S', T', F', M_i) is b-bounded, where $\sum_{s \in S'} M_i(s) \leq b$ [8]. As $\sum_{s \in S'} M_i(s) = 1$, each S-component is safe. Hence, $\overline{\Sigma}$ is safe and so is Σ.

$\square$

Lemma 3.5 [1] The soundness for well-structured workflow nets can be solved in polynomial time.

Proof Let Σ be a well-structured workflow net. Its closure is elementary extended non self-controlling and structurally bounded. As for bounded elementary extended non self-controlling Petri nets the problem of deciding whether a given marking is live can be solved in polynomial time [5], the problem of deciding whether $(\overline{\Sigma}, M_i)$ is live and bounded can be solved in polynomial time. Therefore, the soundness for well-structured workflow nets can be solved in polynomial time. $\square$

3.3.2 Soundness and Home Markings

We recall that, using the definition of the home marking, a workflow net is sound if M_o is a home marking of it and any transition is quasi-live. Therefore, deciding the soundness property can be reduce to the home marking problem (which consist in deciding whether a given marking is a home marking) and to the quasi-liveness problem of each transition of the workflow net.

The home marking problem is decidable as it was shown in [10].

Theorem 3.2 [10] The home marking problem is decidable for marked Petri nets.

To decide if a transition is quasi-live we can use the coverability tree of the workflow net. More precisely, for each transition t, we define the minimal marking M_t at which t is enabled as being $M_t(s) = W(s,t)$, for all $s \in S$. Then, the transition t is quasi-live if M_t is coverable in the workflow net. Since the coverability problem is decidable, we obtain that the quasi-liveness problem is decidable as well and we can use the coverability tree to decide it.

3.4 Complexity

Since the soundness of a workflow net is equivalent with the boundedness and liveness of its closure, then to decide the soundness of workflow nets is as hard as to decide the boundedness and liveness of Petri nets. This two properties are decidable for Petri nets and we shall show that they have very large complexities [11].

Boundedness was proved decidable by Karp and Miller in [25] but the algorithm they gave turned out to be inefficient. In 1978, Rackoff gave an algorithm to decide boundedness [39] that works in $2^{cn\, log\, n}$ space, for some constant c. Rackoff's result is refined in [44] by Rosier and Yen. Their algorithm have complexity $2^{ck\, log\, k}(l + log\ n)$ space, where k is the number of places, l is the maximum number of inputs or outputs of a transition, and n is the number of transitions. They also have shown that, if k is kept constant, then the problem is $PSPACE$-complete for $k \geq 4$.

The problem of deciding boundedness is $PSPACE$-complete even for classes of simple Petri nets.

Definition 3.11 Let $\gamma = (\Sigma, M_0)$ be a marked Petri net.

1. A sequence $M_0[t_1\rangle M_1[t_2\rangle \cdots [t_n\rangle M_n$, where $t_j \in T$, $j = 1, \ldots, n$, and M_j are markings of Σ, is called a *finite path*. A path is *complete* if it cannot be extended.

2. γ is a *single-path Petri net* if it has only a complete path (only a transition is enabled at every reachable marking).

Theorem 3.3 [18] Boundedness problem is $PSPACE$-complete for single-path Petri nets.

To decide if a Petri net is single-path is as hard as to decide boundedness for them.

Theorem 3.4 [18] The problem of recognizing a single-path Petri nets is $PSPACE$-complete.

There are several classes of Petri nets whose boundedness can be decided more efficient.

Definition 3.12 Let $\gamma = (\Sigma, M_0)$ be a marked Petri net. We shall assume that each circuit $c = x_1, x_2, \cdots, x_k$ begin with a place and we denote by $pl(c)$ the set of places in c.

1. c is called *minimal* if $pl(c)$ does not properly include a set of places in any other circuit.

2. c has a *sink* if for some marking M reachable from M_0 and some $w \in T^*$ and M' such that $M[w\rangle M'$, we have $M(s) > 0$, for some $s \in pl(c)$ but $M'(s) = 0$, for all $s \in pl(c)$. c is called *sinkless* if it does not have a sink.

3. γ is a *sinkless Petri net* if each minimal circuit of it is sinkless.

4. γ is a *normal Petri net* if every minimal circuit c and each transition t_j satisfy $\sum_{s_i \in pl(c)} I_\Sigma(i,j) \geq 0$ (the firing of a transition at any marking can not decrease the number of tokens of a minimal circuit).

Theorem 3.5 [21] Boundedness problem is *co-NP*-complete for sinkless and normal Petri nets.

Theorem 3.6 [21]

1. The sink detection problem is *NP*-complete for Petri nets.

2. The problem of determining whether a Petri net is normal is *co-NP*-complete.

Definition 3.13 Let $\gamma = (\Sigma, M_0)$ be a marked Petri net. γ is a *conflict-free* if for every place s with more then one output transition, every output transition of s is also its input transition.

Theorem 3.7 [20] The boundedness problem for conflict-free Petri nets is solvable in O^{n^2} time, where n is the number elements of the incidence matrix of the Petri net.

In [15], Hack has shown that the liveness problem is recursively equivalent to the reachability problem. The computational complexity of the liveness problem is open, there exists partial solution for different classes.

Theorem 3.8 The liveness problem is:

1. $PSPACE$-complete for safe Petri nets [7];
2. *co-NP*-complete for free-choice Petri nets [23];
3. polynomial for bounded free-choice Petri nets [13];
4. polynomial for conflict-free Petri nets [19].

In 1992 Kemper and Bause [26] give a polynomial time algorithm which decide if a free-choice Petri net is structurally live and structurally bounded and has a live initial marking. The complexity of the algorithm is estimated in the worst case to $O(n^4)$ with $n = max(|S|, |T|)$.

In [8] Desel and Espartza describe an efficient algorithm which checks whether a free-choice Petri net satisfies both properties, liveness and boundedness.

The algorithm is based on the rank theorem given below.

Definition 3.14 Let Σ be a Petri net, and x a node of it. The *cluster* of x, denoted by $[x]$, is the minimal set of nodes such that

- $x \in [x]$;
- if a place s belongs to $[x]$ then $s^\bullet$ is included in $[x]$;

- if a transition t belongs to $[x]$ then ${}^{\bullet}t$ is included in $[x]$.

We denote by C_Σ the set of clusters of Σ.

Theorem 3.9 [8] Let Σ be a free-choice net, I_Σ be its incidence matrix, and C_Σ the set of clusters of Σ. Σ is well-formed if and only if:

1. it is connected and has at least one place and one transition;

2. it has a positive S-invariant;

3. it has a positive T-invariant;

4. $Rank(I_\Sigma) = |C_\Sigma| - 1$.

Corollary 3.1 [8] Given a free-choice marked Petri net, we can decide in polynomial time if it is live and bounded.

Corollary 3.2 [1] The soundness of free-choice workflow nets can be solved in polynomial time.

Chapter 4

Time Constraints in Workflow Net Theory

The workflow management is a complex process including definition, verification, monitoring, control, optimization of processes that are often subject to timing constraints. Traditional workflow models support representation of external events and simultaneous actions and are able to deal with the combination of sequential relationship and concurrency.

Time plays an important role in the management of business processes. Business processes try to reduce turnaround times and improve process execution duration estimates in order to improve competitiveness. Many of them have time-related restrictions, including bounded execution durations for activities and subprocesses and absolute deadlines associated with activities and sub-processes. Consequently, time management should be part of the core management functionality provided by workflow systems to control the life cycle of business processes.

Time planning relies on estimates based on experience. Time management during the execution of a process becomes even more important when time monitoring is essential for adjusting plans to avoid deadline misses.

In the remainder of this thesis we shall present some extensions of workflow nets with timing constraints, each one corresponding to a type of Petri net introduced in Chapter 2.

4.1 Time Workflow Nets

Definition 4.1 A TPN $\gamma = (\Sigma, I)$ is called a *time workflow net* (TWN) if its underlying net is a workflow net.

The definition of the soundness property of time workflow nets is similar to the one of classical workflow nets.

Definition 4.2 A TWN $\gamma = (\Sigma, I)$ is called *sound* if it satisfies the following properties:

1. $(M_o, \emptyset) \in [(M, I)\rangle$, for any (M, I) reachable from (M_i, I_0);
2. $(\forall t \in T)(\exists \theta \in \mathbf{Q}_0^+ \exists (M, I) \in [(M_i, I_0)\rangle)((M, I)[(t, \theta)\rangle)$.

Like classical workflow nets, the soundness property of a time workflow net can be characterized by its closure.

Definition 4.3 A TPN $\overline{\gamma} = (\overline{\Sigma}, \overline{I})$ is called a *closure* of a TWN $\gamma = (\Sigma, I)$ if $\overline{\Sigma}$ is obtained from Σ just by adding a new transition t^* and two arcs (o, t^*) and (t^*, i), and setting the static interval of t^* to $(0, 0)$.

Theorem 4.1 ([30]) A TWN γ is sound if and only if its closure $\overline{\gamma}$ is live and M_i is a home marking of $\overline{\gamma}$.

Proof If a TWN γ is sound then the liveness of the transitions in $\overline{\gamma}$ is equivalent with the fact that γ has no dead transition. $\square$

Now, we can give the characterization theorem for soundness property of a time workflow net:

Theorem 4.2 A TWN γ is sound if and only if its closure $\overline{\gamma}$ is bounded and live with respect to (M_i, I_0).

Proof Let γ be a TWN and $\overline{\gamma}$ its closure.

Assume that $\overline{\gamma}$ is bounded and live with respect to (M_i, I_0). In order to prove that Definition 4.2(1) holds, we consider a reachable state (M, I) from (M_i, I_0) in γ. Clearly, (M, I) is also reachable in $\overline{\gamma}$ from (M_i, I_0), and the liveness property shows that there exists (M', I') reachable from (M, I) in $\overline{\gamma}$ such that $(M', I')[(t^*, \theta)\rangle_{\overline{\gamma}}$, where θ is the time at which the marking (M', I') is reached.

Clearly, $M' \geq M_o$. If we assume that $M' > M_o$ then, by firing t^* a state (M'', I'') which satisfies $M'' > M_i$ is obtained. However, this contradicts the fact that $\overline{\gamma}$ is bounded. Therefore, Definition 4.2(1) holds true. The fact that Definition 4.2(2) holds true follows easily from the liveness property of $\overline{\gamma}$.

Conversely, assume that γ is sound. Let us assume that $\overline{\gamma}$ is not bounded with respect to (M_i, I_0). Then, for all $n \in \mathbf{N}$ there exists a state (M, I) reachable from (M_i, I_0) and some $s \in S$ such that $M(s) > n$. Thus, there exists $t \in s^\bullet$ such that t may fire at (M, I) twice successively. Therefore, for all $n \in \mathbf{N}$ there exists $s' \in t^\bullet$ and two times θ_1 and θ_2 such that $(M, I)[(t, \theta_1)(t, \theta_2)\rangle_{\overline{\gamma}}(M', I')$ and $M'(s') > n$. If we repeat the previous process, then for all (M', I') reachable from (M, I) there is $s \in S$ such that $M'(s) > n$. Then, there exists a feasible firing schedule at (M, I) such that $(M, I)[\alpha\rangle_{\overline{\gamma}}(M', I')$ and $M'(o) > n$, which contradicts the fact that $(M_o, \emptyset)$ is reachable from (M, I).

Let us prove now that $\overline{\gamma}$ is live with respect to (M_i, I_0). Let (M, I) be a state reachable from (M_i, I_0) in $\overline{\gamma}$ and t be an arbitrary transition. (M, I) is reachable from (M_i, I_0) in γ as well, and since γ is sound we have that $(M_o, \emptyset)$ is reachable from (M, I) in γ. This shows that $(M_o, \{(0,0)\})$ is reachable from (M, I) in $\overline{\gamma}$ too.

By $(M_o, \{(0,0)\})[(t^*, \theta)\rangle_{\overline{\gamma}}(M_i, I_0)$, where θ is the moment $(M_o, \{(0,0)\})$ is reached, and by the property in Definition 4.2(2), there exists a time θ' and a marking (M', I') reachable from (M_i, I_0) (both in γ and in $\overline{\gamma}$) such that t is firable at (M', I') at time θ'. Therefore, $\overline{\gamma}$ is live. $\square$

We can give a similar proof for Theorem 4.2 by using the firing rule introduced in [36]. Therefore, for the following two classes of time workflow nets the soundness property is equivalent with the soundness property of their underlying nets.

Theorem 4.3 Let $\gamma = (\Sigma, I)$ be a *TWN* such that $I_1(t) = 0$, for all $t \in T$. γ is sound if and only if its underlying net Σ is sound.

Proof Follows directly from Theorem 4.2 and Proposition 2.1. $\square$

Theorem 4.4 Let $\gamma = (\Sigma, I)$ be a *TWN* such that $I_2(t) = \infty$, for all $t \in T$. γ is sound if and only if its underlying net Σ is sound.

Proof Follows directly from Theorem 4.2 and Proposition 2.2. $\square$

In [9] an extension of time workflow nets, called *logical time workflow nets*, is presented. Their underlying nets $\Sigma = (S, T, F)$ are inhibitor nets. Besides ordinary transitions, a logical time workflow net have two sets of logical transitions (one of logical input transitions and one of logical output transitions) and two types of places (control places and interface places). A logical expression is associated to each logical transition. For each place of interface p we have $|({}^\bullet s \cup s^\bullet) \cap T| = 1$ and for all $t \in T$, ${}^\bullet t$ and $t^\bullet$ include one control place at least, respectively. The input and output places are control places.

The firing rule is the same as the one for time Petri nets with test arcs except that:

- a logical input transition is enabled at a state if all its input places satisfy the logical expression associated with the transition;
- if a transition is enabled at a state, then it removes tokens only from those pre-places which are not contained in a conjunctive clause of the logical expression associated to the transition;
- after the firing of a logical output transition all its output places must satisfy the logical expression associated with the transition;

The soundness property for logical time workflow nets is defined as for time workflow nets. Theorem 4.2 holds true for logical time Petri nets too.

4.2 Timed Workflow Nets

In this section we shall introduce X-timed workflow nets, where $X \in \{L, E, A, S\}$. We shall define the soundness property for these types of timed workflow nets and we shall study this property for each type of these nets.

Every type of timed Petri net gives rise, in a very natural way, to a *timed workflow net*.

Definition 4.4 A XT_dPN $\gamma = (\Sigma, \delta)$, where $X \in \{L, E, A, S\}$, is called an *X-timed workflow net* (XT_dWN) if its underlying net is a workflow net.

The soundness property for timed workflow nets can be defined in a similar way as for classical workflow nets and time workflow nets.

Definition 4.5 An XT_dWN $\gamma = (\Sigma, \delta)$, where $X \in \{L, E, A, S\}$, is called *sound* if it satisfies the following properties:

1. $(M_o, \emptyset, \emptyset) \in [(M, C, \rho)\rangle$, for any (M, C, ρ) reachable from $(M_i, \emptyset, \emptyset)$ (i is the input place and o is the output place of the underlying workflow net of γ);

2. $(\forall t \in T)(\exists (M, C, \rho) \in [(M_i, \emptyset, \emptyset)\rangle)((M, C, \rho)[t^+\rangle)$.

Remark 4.1 If $\gamma = (\Sigma, \delta)$ is a sound XT_dWN, where $X \in \{L, E, A, S\}$, then the following property holds true:

$$(\forall (M, C, \rho) \in [(M_i, \emptyset, \emptyset)\rangle)(M(o) \geq 1 \Rightarrow M = M_o).$$

Indeed, let (M, C, ρ) be a reachable timed state such that $M(o) \geq 1$. If we assume that $M_o = 1$ and $M(s) \geq 1$ for some place $s \neq o$, or $M(o) > 1$, then the timed state $(M_o, \emptyset, \emptyset)$ would be never reachable from (M, C, ρ) because each transition in γ has at least an output arc and $o^\bullet = \emptyset$.

Remark 4.2 Definition 4.5(1) also enforces that all transitions of a sound XT_dWN will eventually be deactivated once they become active (see Remark 2.2).

4.2.1 Soundness of LT_dWN and ET_dWN

In this section we shall show that for X-timed workflow nets, where $X \in \{L, E\}$, soundness is decidable and it can be reduced to the boundedness and liveness properties for X-timed Petri nets.

Definition 4.6 An XT_dPN $\overline{\gamma} = (\overline{\Sigma}, \overline{\delta})$, where $X \in \{L, E, A, S\}$, is called a *closure* of an XT_dWN $\gamma = (\Sigma, \delta)$ if $\overline{\Sigma}$ is obtained from Σ just by adding a new transition t^* and two arcs (o, t^*) and (t^*, i) and $\overline{\delta}$ is obtained by extending δ to t^* by $\overline{\delta}(t^*) = 0$.

Theorem 4.5 An XT_dWN γ, where $X \in \{L, E\}$, is sound if and only if its closure is bounded and live with respect to $(M_i, \emptyset, \emptyset)$.

Proof Let γ be an XT_dWN, where $X \in \{L, E\}$, and $\overline{\gamma}$ its closure.

Assume that $\overline{\gamma}$ is bounded and live with respect to $(M_i, \emptyset, \emptyset)$. In order to prove that Definition 4.5(1) holds, we consider a timed state (M, C, ρ) reachable from $(M_i, \emptyset, \emptyset)$ in γ. Clearly, (M, C, ρ) is also reachable in $\overline{\gamma}$ from $(M_i, \emptyset, \emptyset)$, and the liveness property shows that there exists (M', C', ρ') reachable from (M, C, ρ) in $\overline{\gamma}$ such that $(M', C', \rho')[t^{*+}\rangle_{\overline{\gamma}}$. We may assume that $C' = \rho' = \emptyset$ because, otherwise, we can deactivate all transitions in C' by applying only time transitions and transitions in C' and this way we obtain a greater marking than M' which still enables t^*.

Clearly, $M' \geq M_o$. If we assume that $M' > M_o$ then, by firing t^{*+} and t^{*-} (in this order) a timed state $(M'', \emptyset, \emptyset)$ which satisfies $M'' > M_i$ is obtained. However, this contradicts the fact that $\overline{\gamma}$ is bounded. Therefore, Definition 4.5(1) holds true. The fact that Definition 4.5(2) holds true follows easily from the liveness property of $\overline{\gamma}$.

Conversely, assume that γ is sound. It is easily seen that

$$[(M_i, \emptyset, \emptyset)\rangle_\gamma = [(M_i, \emptyset, \emptyset)\rangle_{\overline{\gamma}}$$

because firing of t^* in $\overline{\gamma}$ returns the net to the timed state $(M_i, \emptyset, \emptyset)$. Hence, to show that $\overline{\gamma}$ is bounded with respect to $(M_i, \emptyset, \emptyset)$ it suffices to show that γ is bounded with respect to $(M_i, \emptyset, \emptyset)$.

Now, let us assume that γ is not bounded with respect to $(M_i, \emptyset, \emptyset)$. By Lemma 2.4 there are two timed states (M, C, ρ) and (M', C', ρ') reachable from $(M_i, \emptyset, \emptyset)$ such $M' > M$ and $C' = C$. Since γ is sound there exists a firing sequence σ such that $(M, C, \rho)[\sigma\rangle_\gamma(M_o, \emptyset, \emptyset)$.

For LT_dWN it is easily seen that a firing sequence σ' can be constructed such that $(M', C', \rho')[\sigma'\rangle_\gamma(M'', \emptyset, \emptyset)$ and $M'' > M_o$, contradicting the fact that γ is sound.

In case of ET_dWN, we transform σ into a new firing sequence as follows. First, remove from σ the first occurrence of each transition in C'. Then, we remove all time transitions; let σ_1 be the sequence such obtained. There exists a deactivating sequence θ for all transitions in C'. Moreover, θ consists of only time transitions and transitions t^- with $t \in C'$. Let $(M', C', \rho')[\theta\rangle(M_1, \emptyset, \emptyset)$. Transform now σ_1 into a sequence σ_2 by inserting time transitions so that $(M_1, \emptyset, \emptyset)[\sigma_2\rangle(M'', \emptyset, \emptyset)$ (it is easy to see that this can be always done). Now, the sequence $\sigma' = \theta\sigma_2$ leads the workflow net from (M', C', ρ') to $(M'', \emptyset, \emptyset)$. Moreover, according to the way σ' was obtained, we have $M'' > M_o$, which contradicts γ's soundness.

Let us prove now that $\overline{\gamma}$ is live with respect to $(M_i, \emptyset, \emptyset)$. Let (M, C, ρ) be a timed state reachable from $(M_i, \emptyset, \emptyset)$ in $\overline{\gamma}$ and t be an arbitrary transition. (M, C, ρ) is reachable from $(M_i, \emptyset, \emptyset)$ in γ as well, and since γ is sound we have that $(M_o, \emptyset, \emptyset)$ is reachable from (M, C, ρ) in γ. This shows that $(M_o, \emptyset, \emptyset)$ is reachable from (M, C, ρ) in $\overline{\gamma}$ too.

By $(M_o, \emptyset, \emptyset)[t^{*+}t^{*-}\rangle_{\overline{\gamma}}(M_i, \emptyset, \emptyset)$ and by the property in Definition 4.5(2), there exists (M', C', ρ') reachable from $(M_i, \emptyset, \emptyset)$ (both in γ and in $\overline{\gamma}$) which enables t^+. Therefore, $\overline{\gamma}$ is live. $\square$

As we have seen, the soundness property for XT_dWN, where $X \in \{L, E\}$, can be reduced to the boundedness and liveness properties for XT_dPN. This is similar to the soundness property for workflow nets [1].

If we want to study the soundness property of a given XT_dWN $\gamma = (\Sigma, \delta)$, where $X \in \{L, E\}$, then what we have to do is to construct the untimed Petri net $\overline{\Sigma}'$ associated to the closure $\overline{\gamma}$ of γ, and then to study the boundedness and liveness properties of $\overline{\Sigma}'$ with respect to $M_{i_\emptyset}$. Theorem 2.8 shows that $\overline{\gamma}$ is bounded (live) with respect to $(M_i, \emptyset, \emptyset)$ if and only if $\overline{\Sigma}'$ is bounded (live) with respect to $M_{i_\emptyset}$. Moreover, in case of LT_dWN and LT_dWN with auto-concurrency, $\overline{\Sigma}'$ can be constructed in linear time with respect to the size (number of places and transitions) of γ. This shows that checking soundness of γ is as hard as checking soundness of Σ.

Corollary 4.1 The soundness property is decidable for XT_dWN, where $X \in \{L, E\}$.

Proof For LT_dWN it follows from Theorem 4.5 and Corollary 2.2 and for ET_dWN it follows from Theorem 4.5 and Corollary 2.4. $\square$

4.2.2 Undecidability of Soundness for ST_dWN and AT_dWN

In what follows, we shall show the decidability status of the soundness problem for type S- and A-timed workflow nets. We shall prove that the type S- and A-timed Petri nets associated to a DCM can be transformed into type S- and A-timed workflow nets, respectively, and the reachability, coverability, boundedness, and quasi-liveness are all undecidable for them as well. As a direct consequence, soundness is undecidable for type S- and A-timed workflow nets.

The S- (A-) timed Petri net associated to a DCM, defined in Section 2.2.4, might not be timed workflow net with input place s_{q_0} and output place s_{q_f} for two reasons:

- there are places $s \neq s_{q_0}$ such that ${}^\bullet s = \emptyset$;
- there are places $s \neq s_{q_f}$ such that $s^\bullet = \emptyset$.

If this is the case with the S- (A-) timed Petri net defined in Section 2.2.4, then we add to it a new place s^* and a new transition t^* with time duration 0, we connect s^* and t^* by a self-loop, we add an arc (t^*, s) for each place $s \neq s_{q_0}$ with ${}^\bullet s = \emptyset$, we add an arc (s, t^*) for each place $s \neq s_{q_f}$ with $s^\bullet = \emptyset$, and we add the arcs (s_{q_0}, t^*) and (t^*, s_{q_f}).

One can easily prove that the S- (A-) timed Petri net defined as above is an S- (A-) timed workflow net, and we shall call it the *S- (A-) timed workflow net associated to a DCM*.

In order to show the undecidability of coverability, boundedness, and quasi-liveness for ST_dWN (AT_dWN) we will define three new S- (A-) timed workflow nets obtained from the S- (A-) timed workflow net γ associated to a DCM A as in Section 2.2.4:

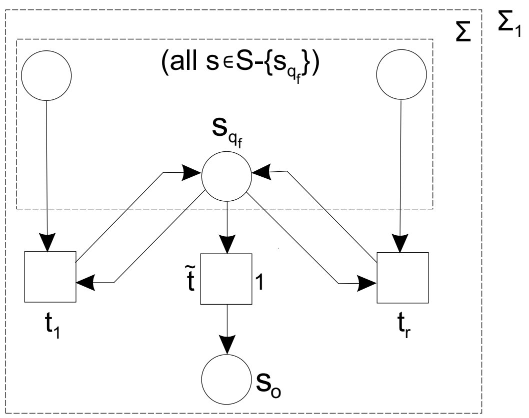


Figure 4.1: Undecidability of coverability for ST_dWN and AT_dWN

- $\gamma_1 = (\Sigma_1, \delta_1, (M^1_{\sigma_0}, \emptyset, \emptyset))$ is obtained from γ as described in Figure 4.1, where $t_1, \ldots, t_r$ are new transitions with time duration 0, $\tilde{t}$ is a new transition with time duration 1, and s_o is a new place. The initial marking $M^1_{\sigma_0}$ marks s_o by zero tokens and all the other places as M_{σ_0} does.

- $\gamma_2 = (\Sigma_2, \delta_2, (M^2_{\sigma_0}, \emptyset, \emptyset))$ is obtained from γ as described in Figure 4.2, where $\tilde{s}$ and s_o are new places and $\tilde{t}$ is a new transition with time duration 0. The initial marking $M^2_{\sigma_0}$ marks s_o and $\tilde{s}$ by zero tokens, and all the other places as M_{σ_0} does.

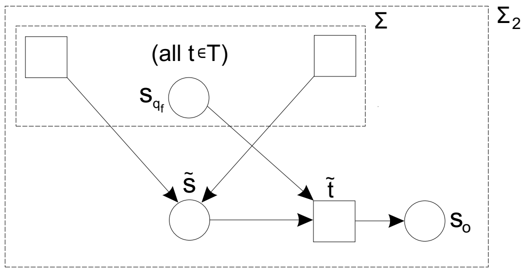


Figure 4.2: Undecidability of boundedness for ST_dWN and AT_dWN

- $\gamma_3 = (\Sigma_3, \delta_3, (M^3_{\sigma_0}, \emptyset, \emptyset))$ is obtained from γ as described in Figure 4.3, where s_o is a new place and $\tilde{t}$ is a new transition with time duration 0. The initial marking $M^3_{\sigma_0}$ marks s_o by zero tokens and all the other places as M_{σ_0} does.

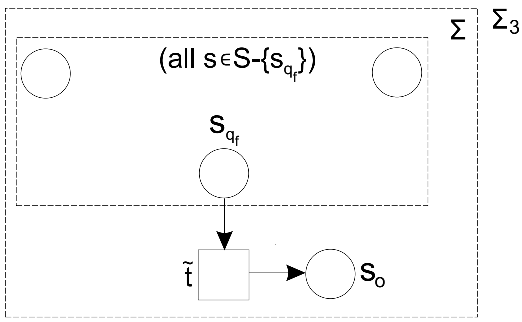


Figure 4.3: Undecidability of quasi-liveness for ST_dWN and AT_dWN

The nets defined above are timed workflow nets of type S (A) with the input place s_{q_0} and the output place s_o. Theorem 2.11 holds true for the S- and A-timed workflow nets associated to a DCM too.

Corollary 4.2 The reachability, coverability, boundedness, and quasi-liveness problems are all undecidable for S-timed workflow nets and A-timed workflow nets.

As the soundness concept requires quasi-liveness (Definition 4.5), we obtain:

Corollary 4.3 The soundness problem for S-timed workflow nets and A-timed workflow nets is undecidable.

Bibliography

[1] W.M.P. van der Aalst, *Structural Characterization of Sound Workflow Nets*, Computing Science Report 96/23, Eindhoven University of Technology, 1996.

[2] W.M.P. van der Aalst, *Three Good Reasons for Using a Petri-net-based Workflow Management System*, Proceeding of the International Working Conference on Information and Process Integration in Enterprises (IPIC96), 179-201, 1996.

[3] W.M.P. van der Aalst, *The application of Petri Nets to workflow Management*, Journal of Circuits, Systems, and Computers, vol. 8, no. 1, 21-66, 1998.

[4] T. Araki, T. Kasami, *Some Decision Problems Related to the Reachability Problem for Petri Nets*, Theoretical Computer Science, vol. 3, no. 1, 85-104, 1977.

[5] K. Barkaoui, J.M. Couvreur, C. Dutheillet, *On Liveness in Extended Non Self-Controlling Nets*, Application and Theory of Petri Nets 1995, Lecture Notes in Computer Science, vol. 935, Springer-Verlag, 25-44, 1995.

[6] B. Berthomieu, M. Diaz, *Modeling and Verification of Time Dependent Systems Using Time Petri Nets*, IEEE Transactions on Software Engineering, vol. 17, no. 3, 259-273, 1991.

[7] A. Cheng, J. Esparza, J. Palsberg, *Complexity Results for 1-Safe Nets*, Theoretical Computer Science, vol. 147, 117-136, 1995.

[8] J. Desel, J. Esparza, *Free-Choice Petri Nets*, Cambridge Tracts in Theoretical Computer Science, nr. 40, Cambridge University Press, 1995.

[9] Y. Du, C. Jiang, *Towards a Workflow Model of Real-time Cooperative Systems*, ICFEM 2003, Lecture Notes in Computer Science, vol. 2885, Springer-Verlag, 452-470, 2003.

[10] D. Frutos Escrig, C. Johnen, *Decidability of Home Space Property*, Report LRI-503, Université de Paris-Sud, Centre d'Orsay, Laboratoire de Recherche en Informatique, 1989.

[11] J. Esparza, M. Nielsen, *Decidability Issues for Petri Nets - a survey*, Bulletin of the EATCS, vol. 52, 244-262, 1994.

[12] J. Esparza, M. Silva, *Circuits, Handles, Bridges and Nets*, Advances in Petri Nets 1990, Lecture Notes in Computer Science, vol. 483, Springer-Verlag, 210-242, 1990.

[13] J. Esparza, M. Silva, *A Polynomial-Time Algorithm to Decide Liveness and Bounded Free-Choice Nets*, Theoretical Computer Science, vol. 102, 185-205, 1992.

[14] K.P. Gostelow, *Flow of Control, Resource Allocation, and the Proper Termination of Programs*, Ph.D. Thesis, School of Engineering and Applied Science, University of California, 1971.

[15] M.H.T. Hack, *Decidability Questions for Petri Nets*, Ph.D. Thesis, Massachusetts Institute of Technology, 1976.

[16] K. van Hee, N. Sidorova, M. Voorhoeve, *Generalised Soundness of Workflow Nets is Decidable*, Proceeding of the 25th International Conference on Application and Theory of Petri Nets, Lecture Notes in Computer Science, vol. 3099, Springer-Verlag, 197-216, 2004.

[17] M.A. Holliday, M.K. Vernon, *A Generalized Timed Petri Net Model for Performance Analysis*, IEEE Transactions on Software Engineering, vol. SE-13, no. 12, 1297-1310, 1987.

[18] R.R. Howell, P. Jančar, L.E. Rosier, *Completeness Results for Single-Path Petri Nets*, Information and Computation, vol. 106, no. 2, 253-265, 1993.

[19] R.R. Howell, L.E. Rosier, *Problems Concerning Fairness and Temporal Logic for Conflict-Free Petri Nets*, Theoretical Computer Science, vol. 64, no. 3, 305-329, 1989.

[20] R.R. Howell, L.E. Rosier, H. Yen, *An $O^{1.5}$ Algorithm to Decide Boundedness for Conflict-Free Vector Replacement Systems*, Information Processing Letters, vol. 25, no. 1, 27-33, 1987.

[21] R.R. Howell, L.E. Rosier, H. Yen, *Normal and Sinkless Petri Nets*, Journal of Computer and System Sciences, vol. 46, no. 1, 1-26, 1993.

[22] L. Jenner, *Further Studies on Timed Testing of Concurrent Systems*, Technical Report 1998-4, Institute of Computer Science, University of Augsburg, 1998.

[23] N.D. Jones, L.H. Landweber, Y.E. Lien, *Complexity of Some Problems in Petri Nets* Theoretical Computer Science, vol. 4, 277-299, 1977.

[24] T. Jucan, F.L. Ţiplea, *Petri Nets. Theory and Practice*, Romanian Academy Press, Bucharest, 1999.

[25] R.M. Karp, R.E. Miller, *Parallel Program Schemata*, Journal of Computer and System Science, vol. 3, 147-195, 1969.

[26] P. Kemper, F. Bause, *An Efficient Polynomial-Time Algorithm to Decide Liveness and Boundedness of Free-Choice Nets*, Proceedings of the 13th International Conference on Application and Theory of Petri Nets, Lecture Notes in Computer Science, vol. 616, Springer-Verlag, 263-278, 1992.

[27] S.R. Kosaraju, *Decidability of Reachability in Vector Addition Systems*, Proceeding of the 14th Annual ACM Symposium on Theory of Computing, 267-281, 1982.

[28] J.Q. Li, Y.S. Fan, M.C. Zhou, *Timing Constraint Workflow Nets for Workflow Analysis*, IEEE Transactions on Systems, Man, and Cybernetics, part A: Systems and Humans, vol. 33, no. 2, 179-193, 2003.

[29] J.Q. Li, Y.S. Fan, M.C. Zhou, *Performance Modeling and Analysis of Workflow*, IEEE Transactions on Systems, Man, and Cybernetics, part A: Systems and Humans, vol. 34, no. 2, 229-242, 2004.

[30] S. Ling, H. Schmidt, *Time Petri Nets for Workflow Modeling and Analysis*, IEEE International Conference on Systems, Man, and Cybernetics (SMC'2000), vol. 4, 3039-3044, 2000.

[31] E. Mayr, *An Algorithm for the General Petri Net Reachability Problem*, Proceeding of the 13th Annual ACM Symposium on Theory of Computing, 238-246, 1981.

[32] P.M. Merlin, *A Study of the Recoverability of Computing Systems*, Ph.D. Thesis, Computer Science Department, University of California, Irvine, 1974.

[33] P.M. Merlin, D.J. Farber, *Recoverability of Communications Protocols - Implications of a Theoretical Study*, IEEE Transactions on Communications, vol. 24, no. 9, 1036-1043, 1976.

[34] M. Minsky, *Recursive Unsolvability of Post's Problem of TAG and Other Topics in Theory of Turing Machines*, Annals of Mathematics, vol. 74, no. 3, 437-455, 1961.

[35] J. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, Englewood Cliffs, 1981.

[36] L. Popova, *On Time Petri Nets*, Journal of Information Processing and Cybernetics, vol. 27, no. 4, 227-244, 1991.

[37] L. Popova-Zeugmann, *On Liveness and Boundedness in Time Petri Nets*, Proceedings of the Workshop on Concurrency, Specification, and Programming, 136-145, 1995.

[38] J.B. Postel, *A Graph Model Analysis of Computer Communications Protocols*, Ph.D. Thesis, Department of Computer Science, University of California, 1974.

[39] C. Rakoff, *The Covering and Boundedness Problem for Vector Addition Systems*, Theoretical Computer Science, vol. 6, 223-231, 1978.

[40] C.V. Ramamoorthy, G.S. Ho, *Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets*, IEEE Transactions on Software Engineering, vol. SE-6, no. 5, 440-449, 1980.

[41] C. Ramchandani, *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*, Project MAC TR-120, Massachusetts Institute of Technology, Cambrige, 1974.

[42] R.R. Razouk, C.V. Phelps, *Performance Analysis Using Timed Petri Nets*, Proceeding of the 1984 International Conference on Parallel Processing, 126-129, 1984.

[43] W. Reisig, *Petri Nets. An Introduction*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1985.

[44] L.E. Rosier, H.C. Yen, *A Multiparameter Analisys of the Boundedness Problem for Vector Addition Systems*, Journal of Computer System Sciences, vol. 32, no. 1, 105-135, 1986.

[45] V. Valero Ruiz, D. de Frutos Escrig, F. Cuartero Gomez, *Simulation of Timed Petri Nets by Ordinary Petri Nets and Applications to Decidability of the Timed Reachability Problem and Other Related Problems*, Proceeding of the 4th International Workshop on Petri Nets and Performance Models, Melbourne, 154-163, 1991.

[46] V. Valero Ruiz, D. de Frutos Escrig, F. Cuartero Gomez, *Decidability of the Strict Reachability Problem for TPN's with Rational and Real Duration*, Proceeding of the 5th International Workshop on Petri Nets and Performance Models, Melbourne, 56-65, 1993.

[47] W. Sadiq, M.E. Orlowska, *Applying Graph Reduction Techniques for Identifying Structural Conflicts in Process Modelss*, Proceedings of the 11th International Conference on Advanced Information Systems Engineering CAiSE99, Lecture Notes in Computer Science, vol. 1626, Springer-Verlag, 195-209, 1999.

[48] W. Sadiq, M.E. Orlowska, *Analyzing Process Models Using Graph Reduction Techniques*, Information Systems, vol. 25, no. 2, 117-134, 2000.

[49] P.H. Starke, *Some Properties of Timed Nets Under the Erliest firing rule*, Advances in Petri Nets 1989, Lecture Notes in Computer Science, vol. 424, Springer-Verlag, 418-432, 1989.

[50] J.J.P Tsai, S.J. Yang, Y.H. Chang *Timing Constraint Petri Nets and Their Application to Schedulability Analysis of Real-Time System Specifications*, IEEE Transactions on Software Engineering, vol. 21, no. 1, 32-49, 1995.

[51] F.L. Ţiplea, C. Bădărău, *A Note for Decidability of Reachability for Conditional Petri Nets*, Acta Cybernetica, vol. 14, no. 3, 455-459, 2000.

[52] F.L. Ţiplea, T. Jucan, C. Masalagiu, *Conditional Petri Net Languages*, Journal of Information Proceeding and Cybernetics, vol. 27, no. 1, 55-66, 1991.

[53] F.L. Ţiplea, G.I. Macovei, *Timed Workflow Nets*, Proceeding of the 7th Interational Symposium on Symbolic and Numeric Algorithms for Scientific Computing - SYNASC 2005, IEEE Computer Society, 361-366, 2005.

[54] F.L. Ţiplea, G.I. Macovei, *E-Timed Workflow Nets*, Proceeding of the 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing - SYNASC 2006, IEEE Computer Society, 423-429, 2006.

[55] F.L. Ţiplea, G.I. Macovei, *Soundness for S- and A-Timed Workflow Nets is Undecidable*, IEEE Transactions on Systems, Man, and Cybernetics, part A: Systems and Humans, vol. 39, No. 4, 924-932, 2009.

[56] W. Vogler, *Timed Testing of Concurrent Systems*, Information and Computation, vol. 121, no. 2, 149-171, 1995.

[57] W.M. Zuberek, *Time Petri Nets and Preliminary Performance Evaluation*, Proceeding of the 7th Annual Symposium on Computer Architecture, 88-96, 1980.

[58] Workflow Management Coalition, http://www.wfmc.com, 1998.

[59] WFMC, *Workflow Management Coalition Terminology and Glosary WFMC-TC-1011*, Technical Report, The Workflow Management Coalition, Brussels, 1999 (http://www.wfmc.com).