Efficient Verification of Parallel Real–Time Systems

Tomohiro Yoneda
Tokyo Institute of Technology
(yoneda@cs.titech.ac.jp)

Bernd–Holger Schlingloff
Technische Universität München
(schlingl@informatik.tu-muenchen.de)

Abstract

This paper presents an efficient model checking algorithm for one–safe time Petri nets and a timed temporal logic. The approach is based on the idea of (1) using only differences of timing variables to be able to construct a finite representation of the set of all reachable states and (2) further reducing the size of this representation by exploiting the concurrency in the net. This reduction of the state space is possible, because the considered linear–time temporal logic is stuttering invariant. The firings of transitions are only partially ordered by causality and a given formula; therefore the order of firings of independent transitions is irrelevant, and only one of several equivalent interleavings has to be generated for the evaluation of the given formula. In this paper the theory of timing verification with time Petri nets and temporal logic is presented, a concrete model checking algorithm is developed and proved to be correct, and some experimental results demonstrating the efficiency of the method are given.

1 Introduction

Model checking has proved to be useful for the automatic verification of finite state systems; see, e.g. [CES86] and others. Unfortunately, the verification of large parallel systems suffers from the so called *state explosion problem*: the number of states to be checked is exponential in the size of the system. An approach to confine this problem is to use partial orders and thus to avoid the construction of equivalent states reachable by different interleaving of atomic events. Several methods [Val90, God90] based on this approach have been proposed for reachability analysis and various other properties of Petri nets.

Those *untimed* verification techniques are suitable to check qualitative timing properties. Recently, the demand for correctness proofs of real–time systems increases rapidly. In real–time systems, the system correctness depends not only on the functional results of the system but also on the time at which these results are produced.

Such systems are often represented by finite automata, whose transitions are labeled by time intervals [AH89, and others], or which have a finite number of clocks [ACD90]. However, concurrency can not be modeled directly by such timed state graphs. On the other hand, *time Petri nets* were considered in [MF76]. Time Petri nets are an adequate model of timed concurrent systems, which generalizes other models (e.g., those of [dBak92]) in a natural way. Using time Petri nets, it is very easy to model, for example, logic gates with bounded delays or network protocols (for an example, see section 6).

In order to specify and verify real–time systems, languages for reasoning about quantitative timing properties are necessary. Many timed temporal logics have been proposed to express such properties [AH89, ACD90, and others]. But again, for practical applications, state explosion is a big problem. There are only a few reports on the avoidance of state explosion in the case of real–time systems. Reachability analysis techniques for time Petri nets using partial orders have been reported in [YTK91]. Symbolic model checking for real–time systems is proposed in [HNSY92].

In this paper, we develop an efficient model checking algorithm for the verification of real–time systems based on the partial order approach. The given real–time system is modeled by a time Petri net. For the specification of properties and time constraints of the time Petri nets we use a suitably extended linear temporal logic. The language is designed such that it fits to the partial order analysis. Automatic verification is achieved by generating a reduced state space of the net, which is traversed with the given formula.

The rest of this paper is organized as follows. In the next section, several definitions concerning time Petri nets are given. In Sect. 3, we introduce our logic. Both the basic model checking algorithm and its partial order improvement are developed in the following two sections. In Sect. 6, some experimental results are presented which demonstrate the efficiency of the proposed method. Finally, we summarize our discussion.

2 Time Petri Nets

Time Petri nets were first defined in [MF76], and used for timing verification in [BD91, RB86]. The definitions here are based on [Sta90].

Let $\mathbf{Q}$ be the set of rational numbers, and $\mathbf{Q}^+$ the set of nonnegative rational numbers. A *time Petri net* N is six–tuple, $N = (P, T, F, \mathit{Eft}, \mathit{Lft}, \mu_0)$, where

- $P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of *places*;
- $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of *transitions* ($P \cap T = \emptyset$);
- $F \subseteq (P \times T) \cup (T \times P)$ is the *flow relation*;
- $\mathit{Eft}, \mathit{Lft} : T \rightarrow \mathbf{Q}^+$ are functions for the *earliest* and *latest firing times* of transitions, satisfying $\mathit{Eft}(t) \leq \mathit{Lft}(t)$ for all $t \in T$;
- $\mu_0 \subseteq P$ is the *initial marking* of the net.

For any transition t, $\bullet t = \{p \in P \mid (p,t) \in F\}$ and $t\bullet = \{p \in P \mid (t,p) \in F\}$ denote the *preset* and the *postset* of t, respectively. To simplify the presentation, we require that $\bullet t \cap t\bullet = \emptyset$ and $\bullet t \neq \emptyset$ for every transition t; however, this requirement is not essential for our results.

Since we are dealing with finite-state systems, each place is limited to at most one "token": A *marking* μ of N is any subset of P. A transition is *enabled* in a marking μ if $\bullet t \subseteq \mu$ (all its input places have tokens in μ); otherwise, it is *disabled.* Let $\mathit{enabled}(\mu)$ be the set of transitions enabled in μ.

A *state* σ of a time Petri net is a pair (μ, clock), where μ is a marking and *clock* is a function $T \rightarrow \mathbf{Q}^+$. The *initial state* σ_0 is $(\mu_0, \mathit{clock}_0)$, where $\mathit{clock}_0(t) = 0$ for all $t \in T$.

The states of time Petri nets change, if time passes or if a transition fires. In state $\sigma = (\mu, \mathit{clock})$, time $\tau \in \mathbf{Q}^+$ *can pass*, if for all $t \in \mathit{enabled}(\mu)$, $\mathit{clock}(t) + \tau \leq \mathit{Lft}(t)$. In this case, state $\sigma' = (\mu', \mathit{clock}')$ is *obtained by passing* τ from σ, if

1. $\mu = \mu'$, and
2. for all $t \in T$, $\mathit{clock}'(t) = \mathit{clock}(t) + \tau$.

In state $\sigma = (\mu, \mathit{clock})$, transition $t \in T$ *can fire*, if $t \in \mathit{enabled}(\mu)$, and $\mathit{clock}(t) \geq \mathit{Eft}(t)$. In this case, state $\sigma' = (\mu', \mathit{clock}')$ is *obtained by firing* t from σ, if

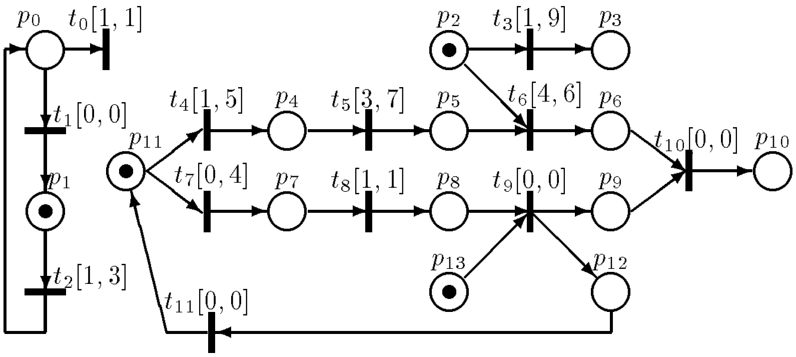

Figure 1: An example of a time Petri net : N_x

1. $\mu' = (\mu - \bullet t) \, \cup \, t\bullet$, and

2. for all $\hat{t} \in T$, $clock'(\hat{t}) = \begin{cases} 0 & \text{if } \hat{t} \in enabled(\mu') - enabled(\mu) \\ clock(\hat{t}) & \text{else} \, . \end{cases}$

Intuitively, this can be interpreted as follows : Firing a transition t consumes no time, but updates μ and *clock* such that the clocks associated with newly enabled transitions (i.e., transitions which are enabled in μ' but not in μ) are reset to 0. Clock values of other transitions (i.e., transitions not affected by t) are left unchanged.

A *run* $\rho = (\sigma_0, \sigma_1, \sigma_2, \ldots)$ of N is a finite or infinite sequence of states such that σ_0 is the initial state, and σ_i is obtained from σ_{i-1} by passing some time τ_i (possibly 0) and then firing some transition t_i. We write $\sigma_i(\rho)$ for the i–th state of ρ, and similarly $\mu_i(\rho)$ and $clock_i(\rho)$, and omit the argument (ρ) whenever appropriate. A run is *maximal*, if it is infinite or in its last state there is no enabled transition. The *behavior* $B(N)$ of N is the set of all maximal runs of N.

Thus $Eft(t)$ and $Lft(t)$ can be seen as a "firing interval" for each transition t which constrains the timing associated with runs of the net. If N' differs from N only in the earliest and latest firing times of its transitions, such that the firing intervals associated with each transition in N' are a subinterval of those of N, then $B(N') \subseteq B(N)$.

Given any run ρ and $i \geq 0$, we define $time_i(\rho)$ to be the sum of all times τ passed between $\sigma_0(\rho)$ and $\sigma_i(\rho)$; that is, $time_0(\rho) = 0$ and $time_{i+1}(\rho) = time_i(\rho) + clock_{i+1}(t) - clock_i(t)$ for some t which is not newly enabled in μ_{i+1}.

A state σ is *reachable* if there exists a finite run whose last state is σ. A time Petri net is *one–safe*, if for every state $\sigma = (\mu, clock)$ obtained by passing time from any reachable state σ', and for every transition t which can fire in σ, $t \bullet \cap \mu = \emptyset$. The restriction to one–safe nets simplifies both the analysis of time Petri nets and the reduced state space generation.

Further, for the proof of the finiteness of the graphs introduced in Sect. 4, we need the following *progress condition* [AH89]: The sum of earliest firing times of transitions forming any loop in N is positive. More precisely, for every set $\{t_1, t_2, \ldots, t_n\}$ of transitions such that $t_1 \bullet \cap \bullet t_2 \neq \emptyset, t_2 \bullet \cap \bullet t_3 \neq \emptyset, \ldots, t_n \bullet \cap \bullet t_1 \neq \emptyset$ it holds that $Eft(t_1) + Eft(t_2) + \cdots + Eft(t_n) > 0$. This guarantees that in any infinite run time is increasing beyond any bound.

In the sequel, a *net* will always be a one–safe time Petri net satisfying the progress condition.

Fig. 1 shows an example net N_x. Pairs of numbers after transition names represent earliest and latest firing times, respectively. Since, for example, t_2 can fire at any time between 1 and 3 after being enabled, the behavior $B(N_x)$ contains an infinite number of runs. Furthermore, since $\mathit{Eft}(t_0) > \mathit{Lft}(t_1)$, t_0 can never fire, and thus every run of N_x is infinite.

In Fig. 2 an example run of the net is shown, where the "p" and "t" columns describe the marking μ_i and clock values $clock_i$ of enabled transitions, respectively. Times τ_i and transitions t_i are chosen arbitrarily, such that the conditions that τ_i can pass and t_i can fire are not violated.

3 TNL, a Timed Temporal Logic for Nets

In this section, we propose a temporal logic for the specification of net properties. On one hand, every such logic should be expressive enough to be capable of formalizing "interesting" properties including quantitative time requirements, and on the other hand there should exist an efficient model checking algorithm for the logic avoiding the state explosion problem. In this paper, we focus on linear time temporal logic.

Given a net N and formula φ, we want to find whether there exists a run ρ of N satisfying φ (written $\rho \models \varphi$). In general there are infinitely many runs of N, therefore we group these into a finite number of equivalence classes $[\rho_1]$, $[\rho_2]$, ..., $[\rho_c]$, such that any run ρ satisfies φ iff every element of the equivalence class $[\rho]$ satisfies φ. Thus we only have to check a finite number of equivalence classes, and a coarser partition yields a better algorithm.

Consider a set of atomic propositions $\{p_1, \ldots, p_k\}$ of a logic, such that the notion of *validity* ($(\rho, i) \models p_j$) of an atomic proposition p_j in a state σ_i of a run ρ is defined. Two runs ρ and ρ' are *strongly equivalent* with respect to $\{p_1, \ldots, p_k\}$, if $(\rho, i) \models p_j$ iff $(\rho', i) \models p_j$ for all $i \geq 0$ and all atomic propositions $p_j \in \{p_1, \ldots, p_k\}$.

A state σ_{i+1} in a run ρ is *stuttering* with respect to $\{p_1, \ldots, p_k\}$, if $(\rho, i) \models p_j$ iff $(\rho, i+1) \models p_j$ for all $p_j \in \{p_1, \ldots, p_k\}$. Two runs ρ and ρ' are *stuttering equivalent* w.r.t. $\{p_1, \ldots, p_k\}$, if the two sequences obtained by eliminating all stuttering states from ρ and ρ' are strongly equivalent w.r.t. $\{p_1, \ldots, p_k\}$. Define a formula φ to be *stuttering invariant*, if for any two runs ρ and ρ' which are stuttering equivalent with respect to the atomic propositions in φ it holds that $\rho \models \varphi$ iff $\rho' \models \varphi$.

Stuttering invariance allows to group all stuttering equivalent runs into the same equivalence class, thereby reducing the average complexity of the model checking. In particular, all runs which differ only in the interleaving of independent transitions are stuttering equivalent with respect to places not connected to these transitions.

i	(τ_i, t_i)	p_0	p_1	p_2	p_3	p_{11}	p_4	p_5	p_{13}	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	$time_i$
0			●	●		●			●			0	0	0			0	0
1	$(2, t_3)$		●		●	●			●			2		2			2	2
2	$(1, t_4)$		●		●		●		●			3			0			3
3	$(0, t_2)$	●			●		●		●	0	0				0			3
4	$(0, t_1)$		●		●		●		●			0			0			3
5	$(3, t_5)$		●		●			●	●			0						6
6	$(0, t_2)$	●			●			●	●	0	0							6
7	$(0, t_1)$		●		●			●	●			0						6
8	...					...								...				

Figure 2: An example run of net N_x

Unfortunately, most formulas of existing real–time logics are not stuttering invariant. Firstly, uncautious use of a "next–state" operator inhibits stuttering invariance. Moreover, if the logic allows to directly refer to the time associated with a state in a run, then a similar effect as with a "next–state" operator can result. In other words, classical real–time logics are inappropriate for our purpose. Therefore, our logic only refers to *differences* of firing times of transitions.

Our logic, which we call **TNL**, is formally defined as follows. Given any net $N = (P, T, F, \mathit{Eft}, \mathit{Lft}, \mu_0)$, let $\mathcal{P} = \{p^\bullet \mid p \in P\} \cup \{p^\circ \mid p \in P\}$ be the set of *time variables.* The set of *propositional variables* is P. The *formulas* of **TNL** are defined inductively:

- If $x, y \in \mathcal{P}$ and $c \in \mathbf{Q}$, then $x - y \leq c$ is a formula.
- Every propositional variable is a formula.
- *false* is a formula.
- If φ_1 and φ_2 are formulas, then $(\varphi_1 \rightarrow \varphi_2)$ and $(\varphi_1 \,\mathcal{U}\, \varphi_2)$ are formulas.

false, propositional variables, and $x - y \leq c$ for $x, y \in \mathcal{P}$ and $c \in \mathbf{Q}$ are called *atomic propositions.* Formulas $x - y \leq c$ are called *atomic inequalities*, and an *atomic formula* is either an atomic proposition or an atomic inequality. Additional boolean connectives *true*, $\neg$, $\wedge$, $\vee$, $\leftrightarrow$, and temporal connectives $\Box$, $\Diamond$ can be defined as usual. Also formulas $x - y \sim c$, where $\sim$ is any relation from $\{<, =, \geq, >\}$, can be defined in an obvious way.

In order to define the semantics of **TNL**, the value of time variables in a state of a run has to be defined. Intuitively, $p^\bullet$ and $p^\circ \in V$ represent the time when the place p got or lost the latest token, respectively.

Let ρ be a run of N, $i \geq 0$, and let $x \in \mathcal{P}$.

$$\mathit{eval}_i(x) = \begin{cases} 0 & \text{if } i = 0 \\ \mathit{time}_i(\rho) & \text{if } x = p^\bullet,\ p \in \mu_i - \mu_{i-1} \\ \mathit{time}_i(\rho) & \text{if } x = p^\circ,\ p \in \mu_{i-1} - \mu_i \\ \mathit{eval}_{i-1}(x) & \text{otherwise} \end{cases}$$

Validity of a **TNL** formula φ in a run ρ at point $i \geq 0$, denoted by $(\rho, i) \models \varphi$, is now defined by induction on φ as usual:

1. $(\rho, i) \models x - y \leq c$ iff $\mathit{eval}_i(x) - \mathit{eval}_i(y) \leq c$
2. $(\rho, i) \models p$ iff $p \in \mu_i$ for $p \in P$
3. $(\rho, i) \not\models \mathit{false}$
4. $(\rho, i) \models (\varphi_1 \rightarrow \varphi_2)$ iff $(\rho, i) \models \varphi_1$ implies $(\rho, i) \models \varphi_2$
5. $(\rho, i) \models (\varphi_1 \,\mathcal{U}\, \varphi_2)$ iff there exists $j \geq i$ such that $(\rho, j) \models \varphi_2$, and for all k such that $i \leq k < j$, $(\rho, k) \models \varphi_1$

ρ *satisfies* φ, denoted by $\rho \models \varphi$, if $(\rho, 0) \models \varphi$. Thus we adopt the so-called *initial* semantics which is more natural for our purposes. φ is *satisfiable* in N if there exists a (maximal) run $\rho \in B(N)$ such that $\rho \models \varphi$.

Consider our example net from Fig. 1. Then the formula $\Diamond p_{10}$ is satisfiable if the place p_{10} is reachable, which is the case, and $\Diamond(p_{10} \wedge p_{10}^\bullet - p_{10}^\circ \leq 8)$ is satisfiable if it can be reached within 8 time units, which is not the case. (Note that $\mathit{eval}_i(p_{10}^\circ) = 0$ for all i; in general we can reference the start time of the net by any time variable related to a constant place.) $\Box\Diamond(p_1^\circ - p_1^\bullet > 2)$ means that t_2 may infinitely often need more than 2 time units to fire.

4 Model Checking for Nets and TNL

In general, there exist infinitely many runs of a given net N. In this section, we will construct a finite graph G such that the paths through G represent exactly the runs of N, and that every node in G determines the truth value of all atomic propositions appearing in the given **TNL** formula. Thus, the **TNL** model checking problem is reduced to the LTL model checking problem, for which an algorithm can be found in [LP85].

Basically, we use a set of inequalities to represent a number of different clock functions. By an *inequality* we mean any string of the form "$x - y \sim c$", where x and y are from a designated set of variables, $c \in \mathbf{Q}$ and $\sim$ is a relation symbol from $\{\leq, <, =, >, \geq\}$. We use "$c \leq x - y \leq d$" as abbreviation for the two inequalities"$x - y \leq d$" and "$x - y \geq c$", and "$x \sim y$" for "$x - y \sim 0$". If I is a set of inequalities, then $var(I)$ denotes the set of variables that I contains; we say that I is a set of inequalities *over* $var(I)$.

Let I be a set of inequalities over $\{x_1, x_2, \ldots, x_n\}$. A *feasible vector* or *solution* for I is a tuple $(c_1, c_2, \ldots, c_n)$ of constants $c_i \in \mathbf{Q}$, such that every inequality obtained by replacing every x_i by c_i $(1 \leq i \leq n)$ in any inequality from I holds in the theory of rational numbers. The *solution set* of I is the set of feasible vectors for I. A set of inequalities is *consistent* if its solution set is nonempty. Two sets of inequalities are *isomorphic*, if they have the same solution set.

The *closure* of a **TNL**–formula φ, denoted by $Cl(\varphi)$, is the smallest set of inequalities such that for every inequality "$x - y \leq c$" appearing in φ, both "$x - y \leq c$"$\in Cl(\varphi)$ and "$x - y > c$"$\in Cl(\varphi)$. A *maximal consistent set* of φ is a maximal set $F \subseteq Cl(\varphi)$ of inequalities which is consistent. Given any set I of inequalities, a *complete extension* I' of I and φ is any consistent set $I' = I \cup I''$, such that I'' is a maximal consistent set of φ. $CE(I, \varphi)$ denotes the set of all complete extensions of I and φ. Note that for consistent I, $CE(I, \varphi)$ is nonempty and finite.

In the previous section, time variables representing times when the corresponding places got or lost its latest token were introduced. In order to grasp the future behavior of the net, we introduce another sort of time variables, called *transition variables*, representing the possible next firing time of (enabled) transitions. Since there is no confusion, we use the set T to denote transition variables as well as transitions; all inequalities in this section will therefore use variables from $V = \mathcal{P} \cup T$. $\mathcal{P}_\varphi$ denotes the set of time variables appearing in φ.

An *atom* is a pair $\alpha = (\mu, I)$, where μ is a marking and I is a set of inequalities. The *initial* atom is $\alpha_0 = (\mu_0, I_0')$, where μ_0 is the initial marking of the net, and I_0' is the unique complete extension of the following set I_0 of inequalities:

$$\begin{array}{rl} I_0 = & \{\text{“}x = y\text{”} \mid x, y \in \mathcal{P}\} \cup \\ & \{\text{“}Eft(t) \leq t - x \leq Lft(t)\text{”} \mid t \in enabled(\mu_0), x \in \mathcal{P}\} \end{array}$$

The first line defines the initial values of all time variables to be equal, and the second line gives the timing constraints on the next firing of transitions enabled in the initial marking.

In our example net N_x, the initial atom contains the inequalities: {"$p_0^\bullet = p_0^\circ = \cdots = p_{13}^\bullet = p_{13}^\circ$", "$1 \leq t_2 - p_0^\bullet \leq 3$", "$1 \leq t_3 - p_0^\bullet \leq 9$", "$1 \leq t_4 - p_0^\bullet \leq 5$", "$0 \leq t_7 - p_0^\bullet \leq 4$"}

We are now going to describe how the set of successor atoms α' of an atom α can be computed. To this end we need the notion of *deletion* of a set U of variables from a set I of inequalities. For every such I and U there exists an (up to isomorphism) unique set $I' = delete(I, U)$ of inequalities over $var(I) - U$, such that the solution set of I' is equal to the solution set of I, projected on $var(I) - U$. For example, if $I = \{$"$1 \leq t_2 - p_0^\bullet \leq 3$", "$1 \leq t_3 - p_0^\bullet \leq 9$"$\}$, then $delete(I, \{p_0^\bullet\}) =$

$\{$"$-8 \leq t_2 - t_3 \leq 2$"$\}$. As shown in [JM87], I' can be computed by a graph–based algorithm in time $O(|var(I)|^3)$.

If we delete in this way all time variables from the above set, we obtain $\{$ "$-8 \leq t_2 - t_3 \leq 2$", "$-4 \leq t_2 - t_4 \leq 2$", "$-3 \leq t_2 - t_7 \leq 3$", "$-4 \leq t_3 - t_4 \leq 8$", "$-3 \leq t_3 - t_7 \leq 9$", "$-3 \leq t_4 - t_7 \leq 5$" $\}$. In fact, before generating the successor atoms of the initial atom we delete from it all time variables which are not necessary for the evaluation of the given formula by applying $delete(I_0, \mathcal{P} - \mathcal{P}_\varphi)$.

Let $\alpha = (\mu, I)$ be an atom, and t_f be a transition enabled in μ. Transition t_f is called *firable in* α, if $I \cup \{$"$t_f \leq t$"$\mid t \in enabled(\mu)\}$ is consistent. That is, t_f is firable in α if it can fire earlier than all other (enabled) transitions in the given marking and timing. *firable*(α) denotes the set of transitions firable in α. Let t_f be a transition in *firable*(α), $\mu' = (\mu - \bullet t_f) \cup t_f \bullet$, and $U_f = \{p^\circ \mid p \in \bullet t_f\} \cup \{p^\bullet \mid p \in t_f \bullet\}$. If $t \in T$ and $p \in \bullet t$, then $parent(t,p) = \{$"$Eft(t) \leq t - p^\bullet \leq Lft(t)$"$\} \cup \{$"$q^\bullet \leq p^\bullet$" $\mid q \in \bullet t\}$ is a set of inequalities describing that the next firing of t is determined by the value of $p^\bullet$. We define the following sets of inequalities:

- $J_1 = I \cup \{$"$t_f \leq t$" $\mid t \in enabled(\mu)\}$
- $J_2 = delete(J_1, U_f)$
- $J_3 = J_2 \cup \{$"$x = t_f$" $\mid x \in U_f\}$
- $J_4 = delete(J_3, \{t \mid t \notin enabled(\mu')\})$
- $J_5 = J_4 \cup \bigcup\{parent(t,p) \mid t \in enabled(\mu') - enabled(\mu),\ p \in t_f \bullet \cap \bullet t\}$
- $J_6 = delete(J_5, \mathcal{P} - \mathcal{P}_\varphi)$

Intuitively, this can be read as follows: J_1 describes that t_f fires first, i.e., earlier than other enabled transitions. J_2 is obtained from J_1 by eliminating all time variables U_f which have to be updated. This updating is then done in J_3 by fixing the value of these variables to be equal to the firing time of t_f. In J_4 the transition variables of disabled transitions are deleted. J_5 relates the transition variables of newly enabled transitions to the updated time variables. Finally, all irrelevant time variables are removed. Note that our definition of the J_i's contains some redundancies; e.g., J_6 can be computed by using the operation $delete(I, U)$ only once. For any α and t_f, J_6 is uniquely determined (up to isomorphism); we say J_6 is *obtained by firing* t_f *from* α. $\alpha' = (\mu', I')$ is a t_f*-successor atom* of α, if $I' \in CE(J_6, \varphi)$ for J_6 obtained by firing some firable transition t_f from α.

An *atom sequence* ϱ is a finite or infinite sequence $\varrho = \alpha_0 \xrightarrow{t_1} \alpha_1 \xrightarrow{t_2} \cdots$, such that α_0 is the initial atom and α_{i+1} is a t_{i+1}-successor atom of α_i for any $i \geq 0$. The *atom graph* $G_\alpha(N, \varphi)$ consists of all atoms reachable by a finite atom sequence, and its edges represent the successor relation of atoms.

To illustrate this construction, in Fig. 3 the first few atoms of N_x (with $\mathcal{P}_\varphi = \emptyset$) are given as calculated in a depth-first search.

Given any atom sequence ϱ, satisfaction of φ in ϱ ($\varrho \Vdash \varphi$) is defined in an obvious way, the relevant clause for atomic inequalities being $(\varrho, i) \Vdash x - y \leq c$ iff "$x - y \leq c$"$\in I_i$. Now, the question of whether there exists a run of N satisfying φ can be reduced to the question of whether there exists a satisfying atom sequence:

Theorem 1

- *For any atom sequence* ϱ *there exists a run* ρ *such that* $\rho \models \varphi$ *iff* $\varrho \Vdash \varphi$ *(correctness), and*
- *for any run* ρ *there exists an atom sequence* ϱ *such that* $\rho \models \varphi$ *iff* $\varrho \Vdash \varphi$ *(completeness).*

```
atom 0 :      u={ p1 p2 p11 p13 }   I={ (-8 ≤ t4 - t3 ≤ 4) (-9 ≤ t7 - t3 ≤ 3) (-8 ≤ t2 - t3 ≤ 2)
                                        (-5 ≤ t7 - t4 ≤ 3) (-4 ≤ t2 - t4 ≤ 2) (-3 ≤ t2 - t7 ≤ 3) }
firable : t3 t4 t7 t2

atom 1: t3    u={ p1 p3 p11 p13 }   I={ (-4 ≤ t7 - t4 ≤ 3) (-4 ≤ t2 - t4 ≤ 2) (-3 ≤ t2 - t7 ≤ 2) }
firable : t4 t7 t2

atom 2: t4    u={ p1 p3 p4 p13 }    I={ (-7 ≤ t2 - t5 ≤ -1) }
firable : t2

atom 3: t2    u={ p0 p3 p4 p13 }    I={ (-6 ≤ t0 - t5 ≤ 0) (-7 ≤ t1 - t5 ≤ -1) (-1 ≤ t1 - t0 ≤ -1) }
firable : t1

atom 4: t1    u={ p1 p3 p4 p13 }    I={ (-6 ≤ t2 - t5 ≤ 2) }
firable : t5 t2

atom 5: t5    u={ p1 p3 p5 p13 }    I={ }
firable : t2

atom 6: t2    u={ p0 p3 p5 p13 }    I={ (-1 ≤ t1 - t0 ≤ -1) }
firable : t1

old atom 5: t1
back to depth 6
back to depth 5

atom 7: t2    u={ p0 p3 p4 p13 }    I={ (-5 ≤ t0 - t5 ≤ 1) (-6 ≤ t1 - t5 ≤ 0) (-1 ≤ t1 - t0 ≤ -1) }
firable : t5 t1

old atom 6: t5

atom 8: t1    u={ p1 p3 p4 p13 }    I={ (-5 ≤ t2 - t5 ≤ 3) }
firable : t5 t2

...
```

Figure 3: Part of the state space of N_x

Proof: To show the correctness of our analysis method we have to show that any atom sequence ϱ corresponds to a possible run ρ of the net. But, this is almost immediate, since any sequence of feasible vectors for the transition variables of an atom sequence determines a run of the net. Note that the set of inequalities in any reachable atom is consistent, because *firable* selects only appropriate transitions, and J_3 and J_5 add only inequalities for previously unconstrained variables. Thus, assume that an atom sequence $\varrho = \alpha_0 \xrightarrow{t_1} \alpha_1 \xrightarrow{t_2} \cdots$ with solutions is given. Define $clock_i(t)$ such that $clock_0(t) = 0$ for all $t \in T$, and $clock_{i+1}(t) = 0$, if t is newly enabled in μ_{i+1}, else $clock_{i+1}(t) = clock_i(t) + c_{i+1}$, where c_{i+1} is the value associated to t_{i+1} in the solution of I_i. Then, it is routine to show that the sequence $((\mu_i, clock_i))$ is a valid run ρ of the net. Clearly, the evaluations of φ in ϱ and ρ are the same.

Let us now prove that the atom graph is also complete, i.e., that any run is represented as a sequence of atoms. Given any run $\rho = (\sigma_0 \xrightarrow{t_1} \sigma_1 \xrightarrow{t_2} \cdots)$ and transition variable t, define $eval_i(t) = \max(time_i(\rho) + Eft(t) - clock_i(t), time_j(\rho))$, where j is the smallest index greater or equal to i such that t is disabled at σ_j. For an enabled transition, $eval_i(t)$ is the time when t fires next, or (if it is disabled before firing) when it could have fired. We construct a sequence $\varrho = (\alpha_0 \xrightarrow{t_1} \alpha_1 \xrightarrow{t_2} \cdots)$ of atoms such that the markings of α_i and σ_i are equal, and $eval_i$ for both time variables and transition variables determines a feasible vector for the inequalities I_i. Then, (ϱ, i) satisfies the same atomic propositions as (ρ, i), and hence both evaluations of φ are the same. Let α_0 be the initial atom. All inequalities of I_0 are valid: Since $eval_0(x) = 0$ for all $x \in \mathcal{P}$, we have $eval_0(x) = eval_0(y)$ for all $x, y \in \mathcal{P}$. Furthermore, for all $t \in enabled(\mu_0)$, it holds that $eval_i(t) = \max(0 + Eft(t) - 0, time_j(\rho)) \geq Eft(t)$, and $eval_i(t) \leq Lft(t)$ since both $Eft(t) \leq Lft(t)$ and $time_j(\rho) \leq Lft(t)$ (note that in any run any enabled transition will be disabled before its latest firing time is elapsed).

Assume inductively that the markings of α_i and σ_i are equal, and that all inequalities in I_i are satisfied by $eval_i$. Let σ_{i+1} be obtained from σ_i by firing t_f. We have to show that $t_f \in \mathit{firable}(\alpha_i)$. Since $t_f \in \mathit{enabled}(\mu_i)$ and $t_f \notin \mathit{enabled}(\mu_{i+1})$ (firing any transition disables it), $eval_i(t_f) = time_{i+1}(\rho)$. Moreover, for any other transition $t \in \mathit{enabled}(\mu_i)$ it is the case that $eval_i(t) \geq time_{i+1}(\rho)$. Therefore $eval_i$ satisfies $I_i \cup \{t_f \leq t \mid t \in \mathit{enabled}(\mu_i)\}$, i.e., $t_f \in \mathit{firable}(\alpha_i)$.

Let I'_{i+1} be the set of inequalities obtained by firing t_f from α_i. We show that $eval_{i+1}$ is a solution for I'_{i+1}. Note that $eval_{i+1}(x) = eval_i(x)$ for any $x \in \mathcal{P} - U_f$, and that $eval_{i+1}(x) = time_{i+1}(\rho)$ for any $x \in U_f$. Similarly, $eval_{i+1}(t) = eval_i(t)$ for any $t \in \mathit{enabled}(\mu_{i+1}) \cap \mathit{enabled}(\mu_i)$, and $eval_{i+1}(t) = \max(time_{i+1}(\rho) + \mathit{Eft}(t), time_j(\rho))$ for $t \in \mathit{enabled}(\mu_{i+1}) - \mathit{enabled}(\mu_i)$.

As we already have seen, $eval_i$ is a solution for $I_i \cup \{t_f \leq t \mid t \in \mathit{enabled}(\mu_i)\}$ (which is J_1 in the above construction). Therefore, the vector assigning $eval_{i+1}(x) = time_{i+1}(\rho) = eval_i(t_f)$ to any $x \in U_f$ and $eval_i$ to all other variables is a solution for J_3, which differs from J_1 only in inequalities over U_f. Since J_5 differs from J_3 only with respect to the timing relations of newly enabled transitions, and J_5 does not contain any transition variables for disabled transitions, $eval_{i+1}$ gives a solution for J_5 (note that $eval_{i+1}(t) \geq time_{i+1}(\rho) + \mathit{Eft}(t)$ and $eval_{i+1}(t) \leq time_{i+1}(\rho) + \mathit{Lft}(t)$ from $time_j(\rho) \leq time_{i+1}(\rho) + \mathit{Lft}(t)$ and $time_{i+1}(\rho) + \mathit{Eft}(t) \leq time_{i+1}(\rho) + \mathit{Lft}(t)$, thus $\mathit{Eft}(t) \leq eval_{i+1}(t) - time_{i+1}(\rho) \leq \mathit{Lft}(t)$). Of course, any solution for a set of inequalities is a solution for the set of inequalities obtained by deleting arbitrarily many variables.

To conclude, let I_{i+1} be the complete extension of I'_{i+1} which satisfies the same atomic inequalities from φ as $(\rho, i+1)$. Note that I_{i+1} is uniquely determined since the value of any atomic inequality can be calculated using $eval_{i+1}$. Since all possible complete extensions of I'_{i+1} are represented in the atom graph, $\alpha_{i+1} = (\mu_{i+1}, I_{i+1})$ is the required successor atom of α_i. **(end of proof)**

If φ contains no time variables, then $G_\alpha(N, \varphi)$ is finite as shown in [BD91]. In this case, all inequalities of all atoms are over the set of transition variables T. Furthermore, any inequality $t_1 - t_2 \simeq c$ satisfies $|c| \leq \mathit{Lft}_{\max}$, where $\mathit{Lft}_{\max}$ is the maximal latest firing time in the net. For, if both t and t' are newly enabled in α, then $\mathit{Eft}(t) \leq t - x \leq \mathit{Lft}(t)$ and $\mathit{Eft}(t') \leq t' - x \leq \mathit{Lft}(t')$ are in J_5. Hence $\mathit{Eft}(t) - \mathit{Lft}(t') \leq t - t' \leq \mathit{Lft}(t) - \mathit{Eft}(t')$ is in J_6. Since $0 \leq \mathit{Eft}(t) \leq \mathit{Lft}(t) \leq \mathit{Lft}_{\max}$, we know that I implies that $-\mathit{Lft}_{\max} \leq t - t' \leq \mathit{Lft}_{\max}$. Similarly, if α' is obtained from α by firing t_f, and $t \in \mathit{enabled}(\mu') \cap \mathit{enabled}(\mu)$, and $t' \in \mathit{enabled}(\mu') - \mathit{enabled}(\mu)$, then we can assume as inductive hypothesis that $-\mathit{Lft}_{\max} \leq t_f - t \leq \mathit{Lft}_{\max}$. J_1 sets $t_f \leq t$, J_3 sets $x = t_f$, and J_5 sets $\mathit{Eft}(t') \leq t' - x \leq \mathit{Lft}(t')$. Hence, we have $-\mathit{Lft}_{\max} \leq -\mathit{Lft}_{\max} + \mathit{Eft}(t') \leq t' - t \leq \mathit{Lft}(t') \leq \mathit{Lft}_{\max}$. Finally, if both t and t' are enabled in α' and α, any inequality $t - t' \simeq c$ is in I iff it is in I'. Therefore, we have shown that the differences between transition variables are bounded.

Moreover, every appearing constant is a linear combination of other rational constants: $c = \sum_{t \in T} \big(n_t \cdot \mathit{Eft}(t) + m_t \cdot \mathit{Lft}(t)\big)$, where n_t and m_t are integers. Since there are only finitely many linear combinations of rationals in a bounded interval, the set of all inequalities in the atom graph is finite. Of course, since we are dealing with one-safe nets, there are also only finitely many possible markings, and hence, finitely many different reachable atoms.

Otherwise, however, if φ refers to the time when certain places got or lost tokens, an infinite number of different atoms may be reachable from the initial atom, because the difference $x - y$ between some time variables may become arbitrarily large. E.g., in our example, for $\varrho = \alpha_0 \xrightarrow{t_3} \alpha_1 \xrightarrow{t_2} \alpha_2 \xrightarrow{t_1} \alpha_3 \xrightarrow{t_2} \alpha_4 \xrightarrow{t_1} \alpha_5 \cdots$, each of $I_1 \cup \{$"$p1^\bullet - p3^\bullet = 0$"$\}$, $I_3 \cup \{$"$p1^\bullet - p3^\bullet = 3$"$\}$, and $I_5 \cup \{$"$p1^\bullet - p3^\bullet = 6$"$\}$ is consistent. In this case, however, every atomic proposition $p1^\bullet - p3^\bullet \leq c$ and $p3^\bullet - p1^\bullet \leq c$ will eventually become constantly false and true, respectively, and thus all α_i in which the difference surpasses a certain threshold value can be considered to be equivalent.

Let max_const be the absolute value of the maximal constant appearing in any subformula of φ, and let I be a set of inequalities. A time variable $x \in \mathcal{P}_\varphi$ is called *saturated in* I, if there is no transition variable $t \in var(I)$ such that the set $I \cup \{$"$t - x \leq max\_const$"$\}$ is consistent; that is, I implies that $max\_const < t - x$ for all enabled transitions. For any two reachable atoms $\alpha_1 = (\mu, I_1)$ and $\alpha_2 = (\mu, I_2)$, let $D = \{x \mid x \text{ is saturated in } I_1 \text{ and } I_2\}$. α_1 and α_2 are *equivalent*, denoted by $\alpha_1 \simeq \alpha_2$, if $I_1 \cap Cl(\varphi) = I_2 \cap Cl(\varphi)$ and $delete(I_1, D) = delete(I_2, D)$, that is, if the same maximal consistent set of φ is a subset of both I_1 and I_2 and the timing relations of I_1 and I_2 with respect to unsaturated variables are isomorphic.

From these definitions we can prove, using similar techniques as in [ACD90]:

Theorem 2

1. *$\simeq$ is a bisimulation; that is, $\simeq$ is an equivalence relation, and for any α_1 and α_2 such that $\alpha_1 \simeq \alpha_2$, and for any α'_1 which is a successor of α_1 there exists a successor α'_2 of α_2 such that $\alpha'_1 \simeq \alpha'_2$.*

2. *$\simeq$ is an equivalence relation of finite index, that is, containing only finitely many equivalence classes.*

Proof: In the proof of $\simeq$ being an equivalence relation, reflexivity and symmetry is immediate. For transitivity, note that if $\alpha_1 \simeq \alpha_2$, then x is saturated in I_1 iff x is saturated in I_2. For, if x were saturated in I_1 but not in I_2, then $x \notin D$, hence "$t - x > c$" $\in I_1$ iff "$t - x > c$" $\in I_2$. But, this contradicts the assumption that $I_2 \cup \{$"$t - x \leq max\_const$"$\}$ is consistent for some t, whereas I_1 implies $t - x > max\_const$. Therefore, $\alpha_1 \simeq \alpha_2$ and $\alpha_2 \simeq \alpha_3$ imply $delete(I_1, D) = delete(I_2, D)$, $delete(I_2, D') = delete(I_3, D')$, and $D = D'$, where $D = \{x \mid x \text{ is saturated in } I_1 \text{ and } I_2\}$ and $D' = \{x \mid x \text{ is saturated in } I_2 \text{ and } I_3\}$. Hence, $delete(I_1, D) = delete(I_3, D)$ as well as $I_1 \cap Cl(\varphi) = I_2 \cap Cl(\varphi) = I_3 \cap Cl(\varphi)$, which means $\alpha_1 \simeq \alpha_3$.

Now, we show that $\simeq$ is a bisimulation. If $\alpha_1 \simeq \alpha_2$, then $\mathit{firable}(\alpha_1) = \mathit{firable}(\alpha_2)$, because the timing with respect to transition variables is isomorphic. Let I'_1 and I'_2 be the sets of inequalities obtained by firing a firable transition t_f from α_1 and α_2, respectively, let D be the set of saturated variables from I_1 (or I_2), and let $D' = D - U_f$. Then, D' is the set of saturated time variables in I'_1 and I'_2, because the modified transition variables are only those for transitions newly enabled I'_1 (I'_2), and for such transition variables t, I'_1 (I'_2) implies $t_f \leq t$, and hence $max\_const < t_f - x \leq t - x$. Furthermore, assume any complete extension I_1^+ of I'_1, and show that a corresponding complete extension I_2^+ of I'_2 exists. For any inequality $x - y \simeq c$ in $I_1^+ \cap Cl(\varphi)$, x and y are either from U_f, from D', or from $\mathcal{P} - (D' \cup U_f)$. If both x and y are in U_f, then I'_1 and I'_2 imply $x - y = 0$. If neither x nor y are from U_f, then $x - y \simeq c$ is in I_i iff it is in I'_i for $i = 1, 2$. If $x \in U_f$ and $y \in D'$, then both I_1 and I_2 imply that $t_f - y > max\_const$, hence both I'_1 and I'_2 imply that $x - y > max\_const$. Therefore, for these cases, the value of "$x - y \simeq c$" is fixed to the same value both in I_1^+ and I_2^+. If $x \in U_f$ and $y \in \mathcal{P} - (D' \cup U_f)$, then since the timing of I_1 and I_2 with respect to unsaturated variables is isomorphic ($delete(I_1, D) = delete(I_2, D)$), we have "$e \leq t_f - y \leq l$" in I_1 iff "$e \leq t_f - y \leq l$" in I_2. From this, "$e \leq x - y \leq l$" in I'_1 iff "$e \leq x - y \leq l$" in I'_2. Therefore, in this case, any consistent extension fixing the value of $x - y \simeq c$ in I_1^+ extends I'_2 consistently to I_2^+. Hence, for any complete extension I_1^+ and the above corresponding I_2^+, we have $I_1^+ \cap Cl(\varphi) = I_2^+ \cap Cl(\varphi)$. Similarly, it is easy to show $delete(I_1^+, D') = delete(I_2^+, D')$, and therefore $\alpha'_1 \simeq \alpha'_2$ is valid.

To show that there are only finitely many inequivalent atoms, we have to show that in any atom the differences between unsaturated variables are bounded by constants. We construct a constant c such that for any atom sequence $\varrho = (\alpha_0 \xrightarrow{t_1} \alpha_1 \xrightarrow{t_2} ...)$, atom $\alpha_i = (\mu_i, I_i)$, unsaturated time variable x and transition variable t, if t is enabled in μ_i, then I_i implies that $0 \leq t - x \leq c$ holds (and hence $-c \leq x - y \leq c$ for any two time variables x, y). Finiteness then follows from the finiteness of $Cl(\varphi)$ and the fact that every constant in every inequality is a linear combination of *Eft*'s and *Lft*'s as above.

Recall that $Lft_{\max}$ is the maximal value of $Lft(t)$ of all $t \in T$. Given i and x, let $j \leq i$ be the maximal index such that x is updated in α_j. By induction on $i - j$ we show that I_i implies that for all t enabled in μ_i, $t - x \leq (i - j + 1) \cdot Lft_{\max}$. As shown above, for all enabled transitions t and t', I_i implies $t - t' \leq Lft_{\max}$. If x is updated in α_i, then $x = t_i$ was set by J_3. Thus, if $i = j$, we have $t - x \leq Lft_{\max}$. If $i > j$, then according to the induction hypothesis, I_{i-1} implies that $t_i - x \leq (i - j) \cdot Lft_{\max}$, since $t_i \in enabled(\mu_{i-1})$. J_3 sets $y = t_i$ for some updated y, and for any newly enabled t we have $t - y \leq Lft(t)$ by J_5. Hence for any such t it holds that $t - x \leq (i - j) \cdot Lft_{\max} + Lft(t) \leq (i - j + 1) \cdot Lft_{\max}$. For t which remain enabled from α_{i-1} to α_i, if I_{i-1} implies that $t - x \leq (i - j) \cdot Lft_{\max}$, then I_i implies that $t - x \leq (i - j) \cdot Lft_{\max}$, hence I_i implies that $t - x \leq (i - j + 1) \cdot Lft_{\max}$.

As a lower bound, note that in any atom α_i for all enabled t and all $x \in \mathcal{P}_\varphi$, from I_i it follows that $t - x \geq 0$. For, if x is updated in α_i, then for all newly enabled transitions t we have $Eft(t) \leq t - x$ and hence $t - x \geq 0$, and for all transitions enabled in μ_{i-1} and not disabled in μ_i, we have $x = t_f \leq t$ by J_3 and J_1, hence $t - x \geq 0$. Furthermore, if x is not updated in α_i, then the minimal value of $t - x$ is nondecreasing: If $c \leq t - x$ for all enabled t in α_{i-1}, then in particular, $c \leq t_f - x$, and hence for all t continuously enabled in α_i also $c \leq t - x$, and for all newly enabled transitions t also $c + Eft(t) \leq t - x$ holds.

We can show that for those x which are not updated, the minimal value of $t - x$ eventually increases: Let $Eft_{\min}$ denote the minimal value of all $Eft(t)$ which is not zero. The progress condition guarantees that there is a constant $M \leq 2^P$ such that in any atom sequence of length M at least one transition t' (newly enabled in the atom sequence) with $Eft(t') \geq Eft_{\min}$ is fired. Consider any atom sequence $(\alpha_\nu, ..., \alpha_{\nu+M})$ in which x is not updated, and suppose that $c \leq t - x$ for all enabled t in α_ν. Furthermore, let t' be newly enabled in $\alpha_{\nu'}$, and fired in $\alpha_{\nu''}$. Then, in $\alpha_{\nu'}$ it holds that $c + Eft(t') \leq t' - x$, and in $\alpha_{\nu''}$ it holds that $c + Eft(t') \leq t - x$ for all enabled t. Since the minimal value of $t - x$ is nondecreasing, in $\alpha_{\nu+M}$ it holds that $c + Eft_{\min} \leq t - x$ for all enabled t.

Summing up, we have shown that with $r = \big((i - j)\ DIV\ M\big)$ in α_i for all enabled t the following inequality holds:

$$r \cdot Eft_{\min} \leq t - x \leq (r + 1) \cdot M \cdot Lft_{\max}$$

Since x was assumed to be unsaturated in I_i, there exists a transition t such that $I_i \cup \{t - x \leq max\_const\}$ is consistent. Hence, $I_i \cup \{r \cdot Eft_{\min} \leq max\_const\}$ is consistent. Since r is a constant value depending only on i and x, this is equivalent to stating that $r \leq max\_const/Eft_{\min}$, which means that I_i implies

$$0 \leq t - x \leq (max\_const/Eft_{\min} + 1) \cdot M \cdot Lft_{\max}.$$

(end of proof)

This theorem shows that there exists a finite set G of representative atoms such that for any atom sequence $\varrho_1 = (\alpha_0, \alpha_1, \alpha_2, \ldots)$ there is a strongly equivalent sequence $\varrho_2 = (\alpha'_0, \alpha'_1, \alpha'_2, \ldots)$ in G such that $\alpha_i \simeq \alpha'_i$ $(i \geq 0)$ and thus $\varrho_1 \mid\!\vdash \varphi$ iff $\varrho_2 \mid\!\vdash \varphi$. The atom graph G can be constructed by depth–first–search from the initial atom, where the equivalence of atoms can be checked efficiently using hash–tables. Note, however, that the size of G can be more than exponential in the size of the net and depends on the number of different constants and their values.

Model checking of **TNL** is performed by building the product of G with the set of all sets γ of subformulas of φ, eliminating from this product all pairs (α, γ) inconsistent with φ, and decomposing the resulting graph into maximal strongly connected components. φ is satisfiable by N iff there is a self–fulfilling strong component, i.e., one which contains with any pair (α_1, γ_1) and any formula $(\varphi_1 \mathcal{U} \varphi_2) \in \gamma_1$ also an pair (α_2, γ_2) such that $\varphi_2 \in \gamma_2$. In our implementation the product and strongly connected components are calculated "on the fly", during the depth-first enumeration of the state space. Thus, if a self–fulfilling strong component is found in an initial part of the state space, we can report a satisfying sequence even if the whole state space is too large to fit into the available memory.

5 Efficiency Improvement by Partial Orders

In this section we show how to reduce the size of the atom graph of a given net and formula without affecting the correctness of the model checking procedure. The reduced state space is obtained by considering a coarser equivalence on atom sequences than the one defined in the previous section. It satisfies the requirement that for any run of the net there exists a stuttering equivalent (w.r.t. atomic propositions in φ) atom sequence in the reduced state space, and vice versa.

Given any atom α_0, firable transition t' and set W of firable transitions, we say that W is *independent from* t' with respect to α and φ, if for any atom sequence $\varrho = (\alpha_0, \alpha_1', \alpha_2', \ldots)$ such that α_1' is obtained by firing t' from α_0 there exists a stuttering equivalent (w.r.t. atomic propositions in φ) atom sequence $\varrho = (\alpha_0, \alpha_1, \alpha_2, \ldots)$ such that α_1 is obtained by firing some $t \in W$ from α_0. Otherwise, we say that W *depends* on t'.

If W is independent from t', we do not have to consider the firing of t' when generating the successors of α in the depth–first–search; there will be a stuttering equivalent sequence constructed by the firing of some $t \in W$.

However, the above definition is not effective; there is no efficient way to compute the smallest set of transitions independent from all other transitions in a given α. Therefore, subsequently we give an algorithm to compute an approximation, that is, for a given firable transition t, we construct the set *dependency*(t, α, φ) (or *dependency*(t), in short) of transitions containing t such that any t' on which *dependency*(t) might depend is included in *dependency*(t).

This idea is similar to the *stubborn set theory* of [Val90] and the *interleaving set temporal logic* of [KP90]; a similar concept was developed independently in [YNT89].

Of course, *dependency*(t) should be as small as possible. For example, if the net N consists of two unconnected subnets N_1 and N_2, and φ mentions only places from one of these, then certainly the set of all firable transitions in N_1 should be independent from any transition in N_2 (if it is not empty), and vice versa. E.g, we don't have to consider the different interleavings of t_2 with t_3, t_4 and t_7 in our example net N_x (shown in Fig. 1) for the formula $\Box\Diamond(p_1^\circ - p_1^\bullet > 2)$.

On the other hand, if for some t, t' which are in *conflict* (i.e., $\bullet t \cap \bullet t' \neq \emptyset$), both t and t' are firable in α, then the firing of t' inhibits that of t; thus in general $\{t\}$ is not independent from t', and we add t' to the dependency of t. So, in N_x, for every firing of t_4 also the alternative of firing t_7 should be considered.

Furthermore, disabled conflicting transitions t' may inhibit the firing of t if they can become enabled by the firing of other (firable) transitions. In the example, although t_6 (in conflict with t_3) is disabled, it may inhibit the firing of t_3, since it can become enabled by the firing of t_4 and t_5. Thus, $\{t_3\}$ may depend on t_4, and t_4 is in *dependency*(t_3).

A set Υ of transitions is *necessary* for t, if for some $p \in \bullet t - \mu$ it holds that $\Upsilon = \{t' \mid p \in t'\bullet\}$. Let *necessary*$^*(t, \alpha)$ be any set of transitions containing t

which is transitively closed under necessity, that is, for any $t' \in \mathit{necessary}^*(t, \alpha)$ such that t' is disabled in μ there exists a set Υ of transitions necessary for t' with $\Upsilon \subseteq \mathit{necessary}^*(t, \alpha)$. For example, $\mathit{necessary}^*(t_6, \alpha_0) = \{t_6, t_5, t_4\}$ in Fig. 1.

If t is in conflict with t_f, then all firable transitions in $\mathit{necessary}^*(t, \alpha)$ should be fired as alternatives to the firing of t_f. The only such transition which could inhibit the firing of t_3 in our above example is t_4.

There is still another class of dependent transitions. We want to obtain stuttering equivalence with respect to the atomic propositions of φ. Usually, φ contains only a few propositional and time variables. A transition t is *visible for* φ if $\bullet t \cup t \bullet$ contains any place p such that p or $p^\bullet$ or p° appears in φ. If t is visible, the firing order with other visible transitions is important. For example, both t_2 and t_3 are visible for the formula $(p_1 \, \mathcal{U} \, p_3)$ in the example net, therefore the firing order between t_2 and t_3 is relevant for the evaluation of $(p_1 \, \mathcal{U} \, p_3)$. Thus t_2 should be in the dependency set of t_3, and vice versa. A visible transition can be regarded as being in conflict with all other visible transitions. Let $\mathit{conflict}^+(t)$ be the set $\{t' \mid \bullet t' \cap \bullet t \neq \emptyset\}$, if t is not visible, else $\mathit{conflict}^+(t)$ is $\{t' \mid \bullet t' \cap \bullet t \neq \emptyset\} \cup \{t' \mid t' \text{ is visible }\}$. Then $\mathit{dependency}(t_f)$ is any set of transitions such that for every $t \in \mathit{conflict}^+(t_f)$ there exists a set $\mathit{necessary}^*(t, \alpha)$ such that all enabled transitions in $\mathit{necessary}^*(t, \alpha)$ are contained in $\mathit{dependency}(t_f)$.

Conceptually, the set of transitions which are fired should be transitively closed under dependency; e.g., in our example, since t_4 is in the dependency set of t_3, and t_7 is in the dependency set of t_4, we have to fire t_7 as an alternative whenever we fire t_3 (p_{10} is only reachable by *first* firing t_7 and *then* t_4). Thus, let $\mathit{dependency}^*(t_f)$ be any set of transitions containing t_f, such that for any $t \in \mathit{dependency}^*(t_f)$ we have $\mathit{dependency}(t) \subseteq \mathit{dependency}^*(t_f)$. When firing a transition t_f we have to fire as alternatives at least all firable transitions which are in $\mathit{dependency}^*(t_f)$.

We should mention that $\mathit{dependency}^*(t_f)$ is a stubborn set in the sense of [Val90]. Dependency sets are "insensitive" to the firing of outside transitions: For any atom α' obtained by firing t' from α, if t' is not in $\mathit{dependency}^*(t_f)$ in α, then any transition $t \in \mathit{dependency}^*(t_f)$ in α is enabled in α' iff it is enabled in α. The same marking will be reached by firing t' after t as by firing t' before t. However, the firability of transition can be affected by this permutation: t might be firable in α' but not in α. Hence, it is not sufficient to fire only transitions in $\mathit{dependency}^*(t_f) \cap \mathit{firable}(\alpha)$, because there might be runs in which a transition in $\mathit{dependency}^*(t_f) - \mathit{firable}(\alpha)$ fires after becoming firable by the firing of an outside transition.

Transition t_f is called *firable with respect to a set of transitions* W in an atom $\alpha = (\mu, I)$, if t_f is enabled in μ and $I \cup \{\text{"}t_f \leq t\text{"} \mid t \in W,\ t \in \mathit{enabled}(\mu)\}$ is consistent. $\mathit{firable}(\alpha, W)$ is the set of transitions firable with respect to W. Note that $\mathit{firable}(\alpha) = \mathit{firable}(\alpha, T)$.

The firability of $t \in \mathit{dependency}^*(t_f)$ with respect to $\mathit{dependency}^*(t_f)$ is not affected by the firing of any $t' \notin \mathit{dependency}^*(t_f)$. When firing a transition t_f, we should fire all transitions in $\mathit{dependency}^*(t_f)$ which are firable with respect to $\mathit{dependency}^*(t_f)$. However, $\mathit{dependency}^*(t_f) \cap \mathit{firable}(\alpha, \mathit{dependency}^*(t_f))$ might contains non-firable transitions. In our algorithm, every successor atom must be obtained by firing a firable transition. Therefore, we are looking for a set of transitions (transitively closed under dependency), such that every transition from this set which is firable with respect to the set is also firable.

Formally, a *ready set*, denoted by $\mathit{ready}(\alpha)$, is a nonempty set of firable transitions such that for any $t \in \mathit{ready}(\alpha)$ it holds that $\mathit{dependency}(t) \cap \mathit{firable}(\alpha, \mathit{ready}(\alpha)) \subseteq \mathit{ready}(\alpha)$. That is, for any $t_f \in \mathit{ready}(\alpha)$, if $t \in \mathit{dependency}(t_f)$ is enabled and can fire earlier than all transitions in $\mathit{ready}(\alpha)$, then $t \in \mathit{ready}(\alpha)$.

If the set $\mathit{firable}(\alpha)$ of all firable transitions is not empty, then it is a ready set, because $\mathit{firable}(\alpha, \mathit{firable}(\alpha)) = \mathit{firable}(\alpha)$ and $\mathit{dependency}(t_f) \cap \mathit{firable}(\alpha) \subseteq \mathit{firable}(\alpha)$. Therefore, for any atom α containing firable transitions there exists at least one ready set. In the sequel $\mathit{ready}(\alpha)$ denotes some such set.

The following algorithm can be used to compute a set $\mathit{ready}(\alpha)$:

1) Start with $\mathit{ready} := \{t_f\}$ for a firable t_f.

2) Iterate $\mathit{ready} := \mathit{ready} \cup (\mathit{dependency}(t) \cap \mathit{firable}(\alpha))$ for some $t \in \mathit{ready}$ until a fixpoint is reached.

3) If there exist $t \in \mathit{ready}$ and $t' \in \mathit{dependency}(t)$ such that $t' \in \mathit{firable}(\alpha, \mathit{ready})$ but not $t' \in \mathit{ready}$, then add some firable t'' to ready, for which I implies $t'' < t'$, and goto 2).

The nondeterminism in the definition of $\mathit{dependency}(t)$ can be resolved by calculating all possibilities and using the smallest set $\mathit{dependency}(t)$. During the construction of the set of successor atoms of an atom we can neglect all firable transitions which are not ready. This results in a considerable average case reduction: For example, in Fig. 1, $\mathit{firable}(\alpha_0) = \{t_2, t_3, t_4, t_7\}$, whereas $\mathit{ready}(\alpha_0) = \{t_2\}$.

However, the construction of the successors of an atom in our partial order method differs from the total order method, because different sets of inequalities have to be built.

Let again $\alpha = (\mu, I)$ be an atom, t_f a transition in $\mathit{ready}(\mu)$, $\mu' = (\mu - \bullet t_f) \cup t_f\bullet$, and $U_f = \{p^\circ \mid p \in \bullet t_f\} \cup \{p^\bullet \mid p \in t_f\bullet\}$. Recall that $\mathit{parent}(t,p)$ is defined to be the set $\{$"$\mathit{Eft}(t) \leq t - p^\bullet \leq \mathit{Lft}(t)$"$\} \cup \{$"$q^\bullet \leq p^\bullet$" $\mid q \in \bullet t\}$. Furthermore, let $\mathit{select}(t)$ be a function selecting some $p \in \bullet t$ for every $t \in T$. We consider the following sets of inequalities:

- $K_1 = I \cup \{$"$t_f \leq t$" $\mid t \in \mathit{ready}(\alpha)\}$
- $K_2 = \mathit{delete}(K_1, U_f)$
- $K_3 = K_2 \cup \{$"$x = t_f$" $\mid x \in U_f\}$
- $K_4 = \mathit{delete}(K_3, \{t \mid t \notin \mathit{enabled}(\mu')\})$
- $K_5 = K_4 \cup \bigcup\{\mathit{parent}(t,p) \mid t \in \mathit{enabled}(\mu') - \mathit{enabled}(\mu),\ p = \mathit{select}(t)\}$
- $K_6 = \mathit{delete}(K_5, \mathcal{P} - \mathcal{P}_\varphi - D)$,
  where $D = \{p^\bullet \mid p \in \mu' \cap \bullet t$ for some transition t disabled in $\mu'\}$

Let us give some comments on this construction. In contrast to the total order method, K_1 relates the firing of t_f only to firings of transitions in the ready set. There might be runs in which transitions that are not ready fire at an earlier time than t_f. K_2, K_3 and K_4 are as in Section 4. K_5 is some set of inequalities obtained by extending K_4 with some parent p for every newly enabled t. Note that in contrast to the total order method, p is not necessarily selected from $t_f\bullet$ as long as p can be a parent (i.e., $p^\bullet$ can be greater than or equal to any other $q^\bullet$ for $q \in \bullet t$). Again, in K_6 irrelevant time variables are deleted, but we keep time variables related to places which could become parents of a transition not (yet) enabled.

$\alpha' = (\mu', I')$ is a *t_f-successor atom* of α, if $I' \in \mathit{CE}(K_6, \varphi)$ for some select-function for which K_5 is a consistent set. The reduced atom graph is constructed in the same way as described in Sect. 4.

Completeness and correctness of our partial order analysis method is granted by the following theorem:

Theorem 3

- *For any run ρ there exists a stuttering equivalent atom sequence ϱ in the reduced atom graph.*

- *For any partial order atom sequence there exists a stuttering equivalent total order atom sequence.*

Proof: We introduce the notion of event in order to distinguish different firings of the same transition and different arriving/leaving of a token in the same place. An *event* e is a pair (x, E), where x is any transition or time variable and E is a set of events. Given any atom sequence or run ϱ, index i, and transition or time variable x, the *history* of x in ϱ and i is the event defined by the following recursive definition:

$$\begin{array}{rcll} \mathit{hist}(t,i,\varrho) & = & (t, \{\mathit{hist}(p_1^\bullet, i, \varrho), ..., \mathit{hist}(p_n^\bullet, i, \varrho)\}) \text{ where } \bullet t = \{p_1, ..., p_n\} \\ \mathit{hist}(p^\bullet,i,\varrho) & = & \left\{ \begin{array}{ll} (p^\bullet, \{\}) & \text{if } i = 0 \\ (p^\bullet, \{\mathit{hist}(t_i, i-1, \varrho)\}) & \text{if } i > 0,\ p \in \mu_i - \mu_{i-1},\ \alpha_{i-1} \xrightarrow{t_i} \alpha_i \\ \mathit{hist}(p^\bullet, i-1, \varrho) & \text{else} \end{array} \right. \\ \mathit{hist}(p^\circ,i,\varrho) & = & \left\{ \begin{array}{ll} (p^\circ, \{\}) & \text{if } i = 0 \\ (p^\circ, \{\mathit{hist}(t_i, i-1, \varrho)\}) & \text{if } i > 0,\ p \in \mu_{i-1} - \mu_i,\ \alpha_{i-1} \xrightarrow{t_i} \alpha_i \\ \mathit{hist}(p^\circ, i-1, \varrho) & \text{else} \end{array} \right. \end{array}$$

An event $e = (x, E)$ *is valid in ϱ at index i*, if $e = \mathit{hist}(x, i, \varrho)$. In case that $e = (t, E)$ is a transition event, we additionally require that t is enabled in μ_i. The event e *is valid in ϱ*, if there exists an i such that e is valid in ϱ at i. As we will see, if an event $e = (x, E)$ is valid in a run or atom sequence, then all events in E are valid in ϱ. A transition event $e = (t, E)$ is *enabled* or *fires* in ϱ at i, if e is valid in ϱ at i, and t is enabled or fires in (ϱ, i). Transition event e is enabled or fires *in* ϱ, if there exists an i such that e is enabled or fires in ϱ at i.

Since the above recursive definition of *hist* is deterministic, for every ϱ and every x, there is no more than one event $e = (x, E)$ valid in ϱ. For any atom sequence $\varrho = ((\mu_0, I_0), (\mu_1, I_1), \cdots)$ and $i \geq 0$, let $\mathcal{I}_i = \bigcup_{k=0}^{i} I'_i$, where I'_i is the set of inequalities such that $\mathit{var}(I'_i)$ is a set of events, that is, I'_i is obtained from I_i by replacing every variable x by the event (x, E) valid in ϱ at i. For every sequence of solutions for $(I_0, I_1, \cdots)$ there is a sequence of solutions for $(\mathcal{I}_0, \mathcal{I}_1, \cdots)$ and vice versa. Similarly, define the value of an event $e = (x, E)$ valid in a run at i to be the value of x in σ_i: $\mathit{EVAL}_i((x, E)) = \mathit{eval}_i(x)$. (Recall that $\mathit{eval}_i(t)$ was defined in the proof of Theorem 1). It is easy to see that if e is valid at i and j, then $\mathit{EVAL}_i(e) = \mathit{EVAL}_j(e)$. Further, if a transition event $e = (t, E)$ is not valid in ρ, but every $d \in E$ is valid in ρ, then we define

$$\mathit{EVAL}(e) = \mathit{EVAL}(d') + \mathit{Lft}(e),$$

where $d' \in E$, for every $d \in E$, $\mathit{EVAL}(d) \leq \mathit{EVAL}(d')$. Therefore, every run ρ determines a unique value $\mathit{EVAL}(e)$ for every event valid in ρ or a transition event like the above.

For the completeness proof, we have to construct for any run ρ a stuttering equivalent atom sequence in the reduced state space. However, in general the two firing sequences will not be strongly equivalent; the atom sequence will correspond to a permutation of the sequence of states of the given run (a *permutation* π of a sequence ρ is any bijection of the integers $\{0, ..., |\rho|\}$ onto itself).

Given any run $\rho = (\sigma_0 \xrightarrow{e'_1} \sigma_1 \xrightarrow{e'_2} \ldots)$, we inductively define a sequence $\varrho = (\alpha_0 \xrightarrow{e_1} \alpha_1 \xrightarrow{e_2} \cdots)$ of atoms in the partial order state space and a permutation π such that the following holds:

1. $\alpha_{i-1} \xrightarrow{e} \alpha_i$ iff $\sigma_{\pi(i)-1} \xrightarrow{e} \sigma_{\pi(i)}$ (the same event fires in α_{i-1} and in $\sigma_{\pi(i)-1}$),

2. If $\mathit{var}(\mathcal{I}_i) = \{e_1, ..., e_n\}$, then $(\mathit{EVAL}(e_1), ..., \mathit{EVAL}(e_n))$ is a solution for $\mathcal{I}_i$,

3. if $\alpha_{i-1} \xrightarrow{e} \alpha_i$ is visible, then the sequences $(\alpha_0, \alpha_1, \cdots, \alpha_i)$ and $(\sigma_0, \sigma_1, \cdots, \sigma_{\pi(i)})$ are stuttering equivalent.

Let α_0 be the initial atom, and $\pi(0) = 0$. Then, the claims 1 – 3 are trivially satisfied for $i = 0$. Assume that $\alpha_0, \ldots, \alpha_i$ has been constructed, and that 1 – 3 hold for all $j \leq i$. Call any state in $\{\sigma_{\pi(0)}, \sigma_{\pi(1)}, \cdots, \sigma_{\pi(i)}\}$ *consumed.*

First, we show that some transition event $e \in \mathit{ready}(\alpha_i)$ fires in ρ. Suppose that $e \in \mathit{ready}(\alpha_i)$. Since e is enabled in α_i and all events up to α_i also fire in ρ, either e is enabled in ρ or a conflicting event with e fires before e being enabled. If e is enabled in some state in ρ, again either e or an event conflicting with e fires. Thus, in any case, e or a conflicting event e' fires in ρ. Since e is in $\mathit{ready}(\alpha_i)$, we only have to consider the latter case. Since e' fires in ρ, some event in $\mathit{necessary}^*(e', \alpha_i)$ must also fire in ρ. If there exists an event e'' in $\mathit{necessary}^*(e', \alpha_i)$ such that e'' is firable in α_i, then e'' is included in $\mathit{ready}(\alpha_i)$. Thus, we have to show that some such e'' is firable: Assume that every event in $\mathit{necessary}^*(e', \alpha_i)$ is disabled in α_i. Since there exists a firing sequence in which e' fires (as in ρ), some conflicting events fire before α_i in ϱ. These events, however, also fire in ρ from the induction hypothesis. This contradicts the assumption that e' fires in ρ. Hence, some $e'' \in \mathit{necessary}^*(e', \alpha_i)$ is enabled in α_i. There exist events e_{parent} and e''_{parent} (before α_i in ϱ) satisfying the following inequalities:

$$\mathit{Eft}(e) \leq \mathit{EVAL}(e) - \mathit{EVAL}(e_{\text{parent}}) \leq \mathit{Lft}(e),$$
$$\mathit{Eft}(e'') \leq \mathit{EVAL}(e'') - \mathit{EVAL}(e''_{\text{parent}}) \leq \mathit{Lft}(e'')$$

Since e' fires in ρ, we have $\mathit{EVAL}(e') \leq \mathit{EVAL}(e)$. Further, $\mathit{EVAL}(e'') \leq \mathit{EVAL}(e')$ from $e'' \in \mathit{necessary}^*(e', \alpha_1)$. Thus, we have

$$\mathit{EVAL}(e''_{\text{parent}}) + \mathit{Eft}(e'') \leq \mathit{EVAL}(e_{\text{parent}}) + \mathit{Lft}(e).$$

From the induction hypothesis, $\mathit{EVAL}(e''_{\text{parent}})$ and $\mathit{EVAL}(e_{\text{parent}})$ satisfy $\mathcal{I}_i$. Hence, e'' can fire earlier than e in α_i (i.e., $\mathcal{I}_i \cup \{$"$e'' \leq e$"$\}$ is consistent). If e'' is firable, then we are done. Otherwise, from the definition of the ready set, some event e_1 which makes e'' non firable (i.e., $\mathcal{I}_i$ implies $e_1 < e''$) is in the ready set (otherwise, e'' must be in the ready set). If we have the same consideration as above for e_1 instead of e, then in this turn we have to show that e''_1 is firable, where $\mathcal{I}_i \cup \{$"$e''_1 \leq e_1$"$\}$ is consistent. If e''_1 is not firable, again there exists e_2 in the ready set, where $\mathcal{I}_i$ implies $e_2 < e''_1$, and the same process can be repeated. Since the ready set is finite, we eventually have $e_k = e_j$ for $0 \leq j < k$ with $e_0 = e$, which implies that $\mathcal{I}_i \cup \{$"$e_k < \cdots \leq e_j = e_k$"$\}$ is consistent. This is, however, a contradiction. Therefore, for some l $(0 \leq l < k)$, e''_l is firable.

Define $\pi(i+1)$ to be the smallest number such that $\sigma_{\pi(i+1)}$ is obtained by firing some $e_f \in \mathit{ready}(\alpha_i)$.

If e_f is invisible, there is nothing to show for the inductive assertion 3. Otherwise, we show by induction on i that all states in $\sigma_0, \sigma_1, ..., \sigma_{\pi(i)}$ obtained by the firing of visible events are consumed. The base case is trivial. Suppose that the induction hypothesis holds for i, i.e., for every state $\sigma_k \notin \{\sigma_{\pi(0)}, ..., \sigma_{\pi(i)}\}$ such that σ_k is obtained by the firing of a visible event we have $k > \pi(i)$. Since e_f is visible and $\sigma_{\pi(i+1)} \notin \{\sigma_{\pi(0)}, ..., \sigma_{\pi(i)}\}$, we have $\pi(i+1) > \pi(i)$ according to the induction hypothesis. Assume that for $\pi(i) < m < \pi(i+1)$, state σ_m is also obtained by a visible event e'. If e' is enabled in α_i, then e' must be in $\mathit{ready}(\alpha_i)$. If e' is not enabled, some event e'' in $\mathit{necessary}^*(e', \alpha_i)$, which is in $\mathit{ready}(\alpha_i)$, must also fire before σ_m (the firability of e' or e'' can be shown similar as above). Either case leads to a contradiction, because $\pi(i+1)$ is defined to be smallest. Hence, the sequences of visible events in $(\alpha_0, \alpha_1, \cdots, \alpha_i)$ and $(\sigma_0, \sigma_1, \cdots, \sigma_{\pi(i)})$ are equal, and we have shown stuttering equivalence of those sequences with respect to propositional variables (i.e., atomic propositions except for atomic inequalities).

Next, we prove inductive assertion 2 together with the remaining part of 3.

Let $I' = \mathcal{I}_i \cup K'_6$, where K'_6 is modified from K_6 by replacing variables with events. For any newly enabled transition event $e = (t, E)$ in α_{i+1}, every $d \in E$ is valid in ρ. Thus, there exists $d' \in E$ such that for every $d \in E$, $EVAL(d) \leq EVAL(d')$. Then,

$$Eft(e) \leq EVAL(e) - EVAL(d') \leq Lft(e)$$

holds. We define $select(e)$ to be any such p. Therefore, from K_5, I' contains in addition to $\mathcal{I}_i$

$$Eft(e) \leq e - d' \leq Lft(e),$$
$$d \leq d'$$

for every $d \in E$. These inequalites are clearly satisfied by $EVAL$.

Further, from the definition of events, no consumed states in ρ are obtained by firable events in α_i, and the earliest unconsumed state is obtained by e_f. Thus, e_f fires in ρ earliest among the events firable in α_i. That is, $EVAL(e_f)$ satisfies $EVAL(e_f) \leq EVAL(e)$ for $e \in ready(\alpha_i)$. This guarantees that the remaining additional inequalities in I'

$$e_f \leq e$$

for $e \in ready(\alpha_i)$ are satisfied by $EVAL$. Therefore, $EVAL$ is a solution for I'.

Suppose that e_f is visible and an atomic inequality $x - y \sim c$ holds in $\sigma_{\pi(i+1)}$. This means that two visible events e_1 and e_2 fired in ρ with $EVAL(e_1) - EVAL(e_2) \sim c$, and e_l ($l \in \{1,2\}$) fired in α_{k_l} for $k_l \leq i$ (because e_1 and e_2 are visible). If $k_l < i$ for $l \in \{1,2\}$, $EVAL(e_1)$ and $EVAL(e_2)$ satisfy I' from the induction hypothesis. The case that $k_l = i$ for $l \in \{1,2\}$ is trivial. Consider one of the remaining cases in which e_1 is e_f and $\sim$ is $\geq$. As shown above, $EVAL(e_f) \leq EVAL(e)$ holds for $e \in ready(\alpha_i)$. Thus, for some event e_{parent} that fired before α_i in ϱ with $select(e) \in e_{\text{parent}}\bullet$ (e_{parent} made e enabled),

$$EVAL(e_2) + c \leq EVAL(e_f) \leq EVAL(e) \leq EVAL(e_{\text{parent}}) + Lft(e)$$

holds. Note that both e_2 and e_{parent} fired before α_i in ϱ. Thus, from the induction hypothesis, they satisfy I'. Since the constraints for e_f are only the inequalities added in K_1, in I' e_f can be as large as $e_{\text{parent}} + Lft(e)$, which means that $I' \cup \{$"$e_2 + c \leq e_f$"$\}$ is consistent. The remaining cases can be proven similarly. Further, the cases that there exists more than one atomic inequality holding in $\sigma_{\pi(i+1)}$ can be handled analogically. Therefore, there exists an complete extension $\mathcal{I}_{i+1}$ of I' such that $\mathcal{I}_{i+1}$ includes atomic inequalites holding in $\sigma_{\pi(i+1)}$ and $EVAL$ is a solution of $\mathcal{I}_{i+1}$.

We have shown that α_{i+1} satisfies the same atomic propositions as $\sigma_{\pi(i+1)}$ if e_f is visible. Hence, the inductive assertion 3 holds.

Since our logic is stuttering invariant, this assertion ensures that the evaluation of any formula in ρ and ϱ yields the same value, provided that every visible event firing in ρ is eventually consumed.

Therefore, we finally show that every event firing in ρ also fires in ϱ, and thus the set of events valid in ρ is equal to the set of events valid in ϱ.

We will show that for an event $e = (t, E)$ which fires in ρ, if every e' (which fires in ρ) such that $(p^\bullet, \{e'\}) \in E$ for some p fires in ϱ, then e also fires in ϱ. If e is not enabled in ϱ, its conflicting event e'' must fire in ϱ. This, however, means that e'' fires also in ρ as shown above, and contradicts the assumption that e fires in ρ. Thus, we can suppose that e is enabled in α_i. From the construction, "$e - e' \leq Lft(e)$" is in $\mathcal{I}_i$. Since nonzero time passes in every loop, e and its direct conflicting events must eventually become the only firable events. Since those direct conflicting events do not fire in ρ, e must be chosen. If we consider a dummy event of which firing generates the initial atom, then it fires both in ρ and ϱ. Hence, every event is eventually consumed.

The correctness proof would be simple, if the set of the partial order atom sequences is a subset of the set of total order atom sequences. However, this does not hold, because in the partial order atom sequences the transition t_f that finally made some t enabled is not always the parent of t. Thus, we have to consider again the permutation to relate both partial and total atom sequences. In the following, for the correctness proof, we construct for any partial order atom sequence a stuttering equivalent total order atom sequence in a way similar to the above completeness proof.

Given any partial order atom sequence $\hat{\varrho} = (\hat{\alpha}_0 \xrightarrow{e'_1} \hat{\alpha}_1 \xrightarrow{e'_2} \ldots)$, we define a total order atom sequences $\varrho = (\alpha_0 \xrightarrow{e_1} \alpha_1 \xrightarrow{e_2} \cdots)$ and a permutation π such that the following holds for all $i \geq 0$:

1. $\alpha_{i-1} \xrightarrow{e} \alpha_i$ iff $\hat{\alpha}_{\pi(i)-1} \xrightarrow{e} \hat{\alpha}_{\pi(i)}$ (the same event fires in α_{i-1} and in $\hat{\alpha}_{\pi(i)-1}$),
2. $\hat{\mathcal{I}}_\infty \cup \mathcal{I}_i$ is consistent, where $\hat{\mathcal{I}}_j$ and $\mathcal{I}_i$ are the accumulated sets of inequalites for $\hat{\varrho}$ and ϱ, respectively, and $\hat{\mathcal{I}}_\infty = \bigcup_j \hat{\mathcal{I}}_j$,
3. if $\alpha_{i-1} \xrightarrow{e} \alpha_i$ is visible, then the sequences $(\alpha_0, \alpha_1, \cdots, \alpha_i)$ and $(\hat{\alpha_0}, \hat{\alpha}_1, \cdots, \hat{\alpha}_{\pi(i)})$ are stuttering equivalent.

For events e_i, e_j that fire in an atom sequence $\hat{\varrho} = (\hat{\alpha}_0 \xrightarrow{e'_1} \hat{\alpha}_1 \xrightarrow{e'_2} \cdots)$, we say that e_i *precedes* e_j at k, if $\hat{\mathcal{I}}_\infty \cup \mathcal{I}_k \cup \{$"$e_i > e_j$"$\}$ is inconsistent. If e_i does not precede e_j at k, then we say that e_j is *precedable* to e_i at k.

Let $\alpha_0 = \hat{\alpha}_0$ be the initial atom, and $\pi(0) = 0$. Again, the base case for the induction is immediate. Assume that $\alpha_0, \ldots, \alpha_i$ has been constructed. In the same way as in the completeness proof, we can show that an event $e \in \mathit{firable}(\alpha_i)$ fires in $\hat{\varrho}$.

Define $\pi(i+1)$ such that $\hat{\alpha}_{\pi(i+1)}$ is obtained by firing some $e_f \in \mathit{firable}(\alpha_i)$ and e_f is precedable at i to any event which is in $\mathit{enabled}(\alpha_i)$ and fires in $\hat{\varrho}$.

If e_f is invisible, there is nothing to show for the inductive assertion 3. Otherwise, assume that for $\pi(i) < m < \pi(i+1)$, $\hat{\alpha}_m$ is obtained by a visible event e'. Since every visible event is ordered even in the partial order atom sequences, e' precedes e_f. If e' is enabled in α_i, then this contradicts that e_f is precedable to e' at i. Otherwise, some enabled event e'' that fires in ϱ must precede e', which again causes the contradiction. Therefore, the inductive assertion 3 holds with respect to propositional variables.

If $e \in \mathit{enabled}(\alpha_i)$ fires in $\hat{\varrho}$, then according to its definition, e_f is precedable to e at i. Otherwise, an event e' which is in conflict with e must fire in $\hat{\varrho}$, and such e' must be enabled in α_i or made enabled by the firing of an event enabled in α_i. In any case, e_f is precedable to e' at i, and hence e. Thus, for any $e \in \mathit{enabled}(\alpha_i)$, $\hat{\mathcal{I}}_\infty \cup \mathcal{I}_i \cup \{$"$e_f \leq e$"$\}$ is consistent.

Suppose that an event $e = (t, E)$ is newly enabled by firing e_f in α_i. If e is not enabled in $\hat{\varrho}$, then a conflicting event e' must fire in $\hat{\varrho}$. Let e' be in the ready set in $\hat{\alpha}_m$ for $m < \pi(i+1) - 1$. Then, some event $e_1 \in \mathit{necessary}(\hat{\alpha}_m, e)$ is also in the ready set. Thus, e' precedes e_1. Since e is enabled in α_i, e_1 fired in α_j for $j \leq i$. e' does not fire up to α_i in ϱ, because e is enabled in α_i. If e' is enabled in α_j, this contradicts the assumption that e_1 is precedable (at j) to transitions enabled in α_j. Otherwise, e' is made enabled by firing some e'_{parent} enabled in α_j. From the causality, such e'_{parent} precedes e' at j, and hence e_1. This again contradicts the assumption.

Thus, e is enabled in $\hat{\varrho}$. This implies that

$$\mathit{Eft}(e) \leq e - d' \leq \mathit{Lft}(e)$$
$$d \leq d'$$

are in $\hat{\mathcal{I}}_\infty$ for $d' = (p^\bullet, E') \in E$, $p = \mathit{select}(e)$, $d = (q^\bullet, \{e''\}) \in E$. Since e_f finally made e enabled in ϱ, every such e'' is precedable to e_f at i. Thus, inequalities $d' = e_f$ must hold. Hence, $\hat{\mathcal{I}}_\infty \cup \mathcal{I}_i \cup \{$"$\mathit{Eft}(e) \leq e - e_f \leq \mathit{Lft}(e)$"$\}$ is consistent. Note that "$\mathit{Eft}(e) \leq e - e_f \leq \mathit{Lft}(e)$" is equivalent to the inequality added in J_5.

We have shown that $\hat{\mathcal{I}}_\infty \cup \mathcal{I}_i \cup J'_6$ is consistent for J'_6 modified from J_6 by using events instead of variables. Suppose an atomic inequality "$x - y \sim c$" holds in $\alpha_{\pi(i+1)}$. Then, for visible event e and e', "$e - e' \sim c$" is in $\hat{\mathcal{I}}_\infty$. Thus, $\{$"$e - e' \sim c$"$\} \cup I'$ is consistent. Hence, for the related complete extension I_{i+1}, $\hat{\mathcal{I}}_\infty \cup \mathcal{I}_{i+1}$ is consistent.

Again, an event finally precedes some other event which fires in $\hat{\varrho}$, because nonzero time passes in every loop. Thus, every atom in $\hat{\varrho}$ is eventually consumed. **(end of proof)**

We have shown that our partial order method captures the behavior of the net correctly and completely. However, in contrast to other partial order methods proposed for untimed systems, our partial order state space is not a subset of the total order state space, since there are different sets of inequalities involved. In particular, we can not conclude that the termination proof of Theorem 2 still holds; in fact, there are cases when the above partial order method does not terminate.

If the net contains several independent loops, transitions in these loops are handled completely in parallel, that is, inequalities which restrict the difference of future firing times of those transitions are not involved. Hence, the time difference of those transitions may not be bounded. To deal with this problem, we require that all transitions for which the time difference to transitions in the ready set exceeds a certain maximum are also fired.

For two enabled transitions t_1 and t_2 in $\alpha = (\mu, I)$, if both $I \cup \{$"$t_1 - t_2 > a$"$\}$ and $I \cup \{$"$t_2 - t_1 > a$"$\}$ are inconsistent, then we say that t_1 and t_2 are *a-bounded* at α. Let $c \geq 1$ be any constant. In addition to the conditions given above, we require the ready sets to satisfy the following property:

- For $t \in \mathit{ready}(\alpha)$ and all firable transition t' at α, if t and t' are not $c \cdot \mathit{Lft}_{\max}$-bounded at α, then $t' \in \mathit{ready}(\alpha)$

Intuitively, this modification guarantees that when the time difference between firable transitions t and t' exceeds some constant value, then for both cases that t fires before t' and that t' fires before t atoms are generated, and in both these atoms the difference between t and t' is restricted.

We say that α is a-bounded, if for all enabled transitions t_1 and t_2 at α, t_1 and t_2 are a-bounded.

Lemma 1 *For any reachable atom $\alpha = (\mu, I)$ and successor $\alpha' = (\mu', I')$ obtained by firing $t \in \mathit{ready}(\alpha)$, if α is $(c+1) \cdot \mathit{Lft}_{\max}$-bounded, then α' is also $(c+1) \cdot \mathit{Lft}_{\max}$-bounded.*

Proof: Let t_1 and t_2 be transitions enabled in α'. We have the following three cases :

- both t_1 and t_2 are enabled also in α,
- both t_1 and t_2 are newly enabled in α', and
- t_1 is enabled in α and t_2 is newly enabled in α'.

In the first case, from the hypothesis t_1 and t_2 are $(c+1)Lft_{\max}$-bounded in α. When firing t in α, inequalities like $t \leq t'$ representing the condition that t fires earlier than the other firable transitions t' in the ready set are added to I. These inequalities only give the restriction to the lower bound of the firing times of continuously enabled transitions. Thus, t_1 and t_2 are also $(c+1)Lft_{\max}$-bounded in α'.

In the second case, some inequalities

$$\begin{aligned} Eft(t_1) \leq t_1 - x' &\leq Lft(t_1), \\ Eft(t_2) \leq t_2 - x'' &\leq Lft(t_2), \\ t &\leq x', \\ t &\leq x'' \end{aligned}$$

are added to I, where x' and x'' are the parents of t_1 and t_2. Let t' be a transition such that $t' = x'$, and assume that t' fired in α_1. We consider some ancestor t_x of t which is firable in α_1. Then, $t_x \leq t$ holds. If t' and t_x are not $c \cdot Lft_{\max}$-bounded, then from the definition of the ready set "$t' \leq t_x$" must hold. This implies

$$t' \leq t_x \leq t \leq x' = t',$$

and hence $t' = t_x$, which contradicts the assumption that t' and t_x are not $c \cdot Lft_{\max}$-bounded. Thus, $t' - t_x \leq c \cdot Lft_{\max}$. From $t_x \leq t$, $x' = t'$, and $t \leq x'$, we have $0 \leq x' - t \leq c \cdot Lft_{\max}$. Similarly, we have $0 \leq x'' - t \leq c \cdot Lft_{\max}$. Hence, we have

$$-c \cdot Lft_{\max} \leq x' - x'' \leq c \cdot Lft_{\max},$$

and thus,

$$-c \cdot Lft_{\max} + Eft(t_1) - Lft(t_2) \leq t_1 - t_2 \leq c \cdot Lft_{\max} + Lft(t_1) - Eft(t_2),$$

which implies that t_1 and t_2 are $(c+1)Lft_{\max}$-bounded in α'.

For the third case, we define x'' and t'' similarly to the above. If $t_1 \leq x''$, then similarly to the above, we have $0 \leq x'' - t_1 \leq c \cdot Lft_{\max}$. Hence, we have

$$Eft(t_2) \leq t_2 - t_1 \leq c \cdot Lft_{\max} + Lft(t_2).$$

If $x'' \leq t_1$, then from the hypothesis,

$$-(c+1)Lft_{\max} \leq t_1 - t \leq (c+1)Lft_{\max}.$$

From $t \leq x''$ and $x'' \leq t_1$, we have

$$0 \leq t_1 - x'' \leq (c+1)Lft_{\max}.$$

Thus, we have

$$-(c+1)Lft_{\max} + Eft(t_2) \leq t_2 - t_1 \leq Lft(t_2).$$

Therefore, in any case it is implied that t_1 and t_2 are $(c+1)Lft_{\max}$-bounded in α'.

(end of proof)

The initial atom is clearly $(c+1)Lft_{\max}$-bounded. Thus, from the above lemma, every reachable atom is $(c+1)Lft_{\max}$-bounded.

Let G' be the quotient of the reduced atom graph G'_α under the equivalence relation induced by the deletion of saturated time variables. Then the proof of bisimilarity given in Section 4 is valid also for G'.

It remains to show that the differences between firable transitions and unsaturated variables are bounded. The proof for this is also very similar to the proof for the total order method given in Section 4. However, the partial order method involves other constants:

Given i and x, let again $j \leq i$ be the maximal index such that x is updated in α_j. By a similar induction as in the total order case we obtain $t - x \leq (i-j+1) \cdot (c+2) \cdot \mathit{Lft}_{\max}$ for all t enabled in μ_i: As shown above, for all enabled transitions t and t', I_i implies $t - t' \leq (c+1) \cdot \mathit{Lft}_{\max}$, thus the base case $(i-j=0)$ follows analogously. For transitions t remaining enabled from α_{i-1} to α_i, we can again rely on the induction hypothesis. For transitions t newly enabled in α_i, we have to distinguish two cases: If *select*(t) is among the updated variables, then we can conclude as in Section 4. Else, there exists a place $p = \mathit{select}(t)$ such that $p \in \mu_{i-1}$. Let $k < i$ be the biggest index such that p was updated in α_k. Then, we can refer to the induction hypothesis that I_{i-1} implies that $t_i - x \leq (i-j) \cdot (c+2) \cdot \mathit{Lft}_{\max}$ (recall that α_i is obtained by firing t_i). Suppose $k > 0$. If t_i was enabled in μ_{k-1}, the upper bound of $t_k - t_i$ is $(c+1)\mathit{Lft}_{\max}$. Otherwise, some transition t'_i was enabled in μ_{k-1} and $t'_i \leq t_i$. Since the upper bound of $t_k - t'_i$ is again $(c+1)\mathit{Lft}_{\max}$, the upper bound of $t_k - t_i$ is less than it. In any case, $t_k - t_i$ is bounded by $(c+1)\mathit{Lft}_{\max}$. From $t - p \leq \mathit{Lft}(t)$ and $p = t_k$, I_i implies that $t - x \leq (i-j+1) \cdot (c+2) \cdot \mathit{Lft}_{\max}$. The remaining cases that $k = 0$ is proven similarly from $p - t_i \leq -\mathit{Eft}(t_i)$.

A lower bound for the partial order method is given by $r \cdot \mathit{Eft}_{\min} - (c+1) \cdot \mathit{Lft}_{\max}$. If x is updated in α_i, again for all newly enabled transitions t we have $\mathit{Eft}(t) \leq t - x$, since *parent*$(t,p)$ implies $\mathit{Eft}(t) \leq t - p^{\bullet}$ and $p^{\bullet} \geq x$ and hence $t - x \geq \mathit{Eft}(t)$, and for all transitions t enabled in μ_{i-1} and not disabled in μ_i, we have $x = t_i \leq t + (c+1) \cdot \mathit{Lft}_{\max}$, since $t_i - t$ is $(c+1) \cdot \mathit{Lft}_{\max}$-bounded in α_{i-1}. The rest of the proof is completely analogous to the total order case.

Thus we have shown:

Theorem 4 *The partial order analysis generates only a finite number of different atoms.*

There are a number of improvements to our method which we omitted in the above presentation to simplify it. The static timing intervals of the transitions in the net can be used to reduce the size of the dependency set of a transition.

In our introductory example, we mentioned that although t_6 is disabled, it may inhibit the firing of t_3, if t_4 and t_5 fire. Therefore, we included t_4 into *dependency*(t_3). However, t_6 will not inhibit the firing of t_3 if t_6 becomes enabled too late. This can be checked by examining the minimal time difference between the next firing times of t_4 and t_3. It takes at least $\mathit{Eft}(t_5) + \mathit{Eft}(t_6) = 3 + 4$ time units to fire t_6 after the firing of t_4. Thus, t_4 can only inhibit the firing of t_3, if t_4 can fire 7 time units earlier than t_3. Hence, we include t_4 in the dependent set of t_3 only if $I \cup \{\text{“}t_3 - t_4 \geq 7\text{”}\}$ is consistent, where $7 = \mathit{diff}(t_4, t_6)$ is the sum of earliest firing times in the path from t_4 to t_6.

Formally, let $\mathit{diff}(t,t')$ be the minimal value of sums of earliest firing times in all paths from t to t', with $\mathit{Eft}(t)$ not included. A transition t_h in *necessary*$^*(t,\alpha)$ is *harmful for* t_f, if it is enabled, and $I \cup \{\text{“}t_f - t_h \geq \mathit{diff}(t_h, t)\text{”}\}$ is consistent. Instead of including all enabled transitions in *necessary*$^*(t,\alpha)$ for all conflicting t into the dependency of t_f it is sufficient to include those which are harmful; harmless transitions can never inhibit the firing of t_f since the conflicting transition they enable becomes enabled “too late”.

Another improvement concerns the deletion of “aged” variables in K_6. In the current definition of K_6, all time variables $p^{\bullet}$ for all marked input place p of a disabled transition t are left in the set of inequalities. Thus, if t is continuously disabled, the difference of these variables to other (transition) variables becomes larger and larger until they are saturated. This is not a problem from the view of correctness, completeness, or termination of the algorithm. However, it contains some redundancy, because time variables which are too old can not be the parent of newly enabled transitions. Here, “too old” means that $p^{\bullet}$ can not be greater or equal to the earliest time when t gets enabled, and such earliest time can be guessed with $t' + \mathit{diff}(t',t) - \mathit{Eft}(t)$ for some enabled transition t'. Thus, if we define D in K_6 as

$$D \;=\; \{p^\bullet \mid p \in \mu' \cap \bullet t, K_5 \cup \{p^\bullet \geq t' + \mathit{diff}(t',t) - \mathit{Eft}(t)\} \text{ is consistent} \\ \text{for some disabled } t \text{ and enabled } t' \text{ in } \mu'\},$$

the algorithm is still correct, and in general more efficient.

6 Experimental Results

We have implemented both the basic model checking algorithm and its partial order improvement on a 17 MIPS UNIX workstation in C++. In this section, the performance of both algorithms with an example from [RB86, YNT89] is demonstrated.

The verified system called PROWAY is a local area network linking stations by a shared hardware bus. The bus allocation procedure is based on a token bus access technique. Fig. 4 shows a Time Petri net model for station 1 of the PROWAY system in a four-station configuration.

Stations are logically distributed on a ring, and a baton goes around on the ring. When a station has the baton, it can transmit application messages, whereas the other stations can only listen to them. A token in p_1 means that the station is in the listening mode. A token in p_3 means that the station has a baton. If transition t_4 fires, the station first transmits application messages and then it passes a baton to the next station on the logical ring. Otherwise, the station only passes a baton without message transmission. On the transmission of messages, the station holds a baton for a longer time. (Compare firing intervals associated with t_9 and t_{10} in Table 1).

Each station has a recovery mechanism against a single fault. A station sets its frame interval timer $T1$ (represented by t_{17}) when it transmits a baton. If any activity on the bus (i.e., baton or message transmission from other stations) is listened a certain time later, the station gets into listening mode, resetting the timer. Otherwise, the frame interval timer times out. Suppose the station S_a transmits a baton to the station S_b. Time-out of the S_a's frame interval timer occurs when (i) a baton from S_a is lost, (ii) S_b is faulty, or (iii) the baton or messages from S_b are lost. In these cases, S_a transmits a new baton to another station S_c. Next time S_a has a baton, S_a tries to transmit the baton to S_b. If $T1$ of S_a times out again, S_a will ignore S_b from now on. p_8, p_9 and p_{10} represent how many times this time-out of $T1$ occurs.

A station sets its lost baton timer $T2$ (represented by t_2) when it gets into listening mode. The purpose of this timer is to initiate a new baton when a baton holder goes faulty, holding the current baton, and all other live stations are in the listening mode. The value of $T2$ is indexed with the station's address as shown in Table 1, in such a way that the live station with the smallest address monitors the recovery.

As example property, we verify if the next activity will always occur within some constant time units, say max, after a station finishes sending its message. This property holds in the system if the **TNL** formula

$$\neg\Box[\mathit{finish} \rightarrow (\neg \mathit{activity})\,\mathcal{U}\,(\mathit{activity}^\bullet - \mathit{finish}^\bullet \;\leq\; max)]$$

is not satisfiable. More concretely, we have checked the formula

$$\neg\Box[p_{12} \rightarrow (\neg p_2)\,\mathcal{U}\,(p_2^\bullet - \; p_{12}^\bullet \;\leq\; 100].$$

The Fig. 5 shows the CPU times for both implemented algorithms with this example. The size of the net is linear in the number n of stations; thus the basic

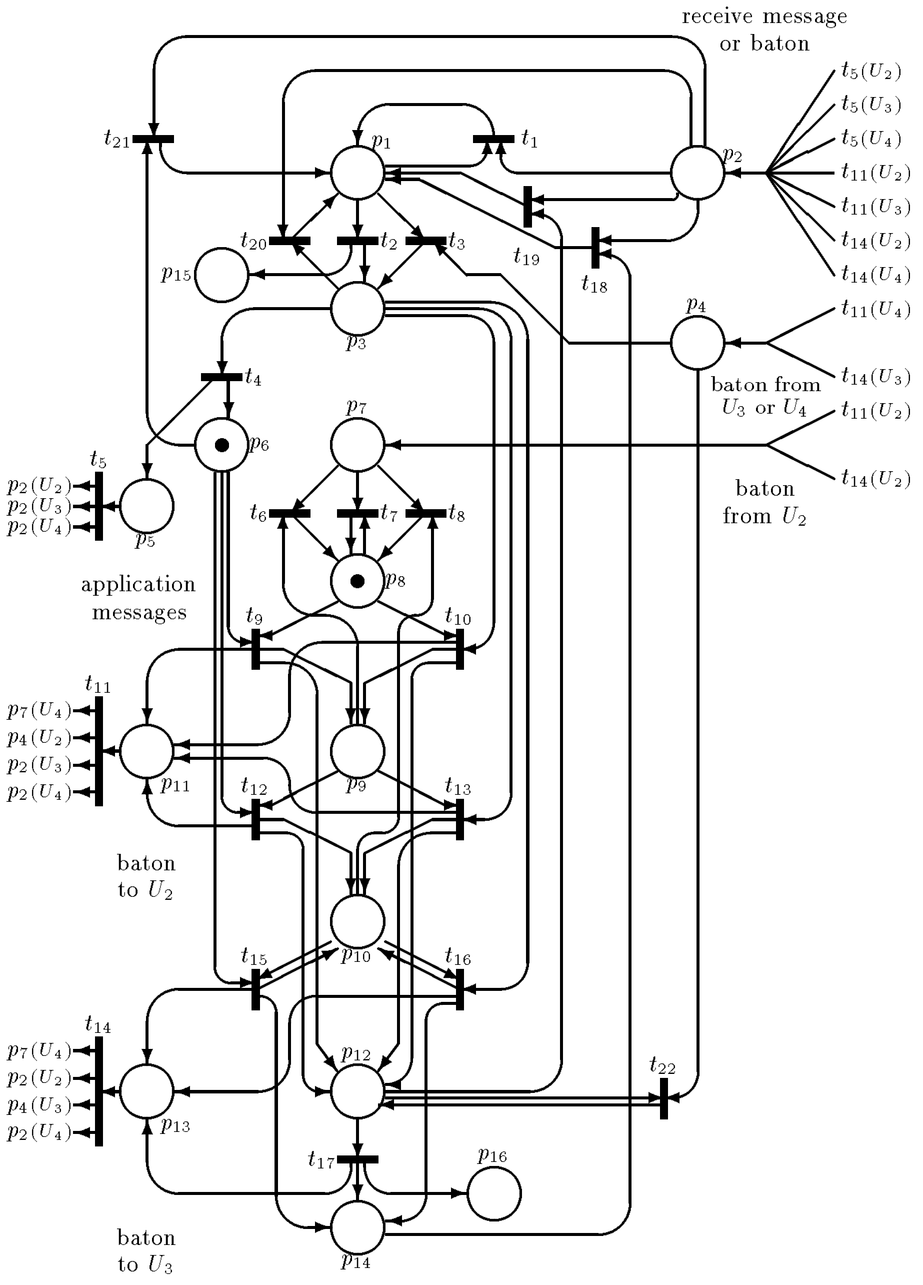

Figure 4: A time Petri net model for station 1 (U_1) of the PROWAY system in a four-station configuration.

Table 1: Timing constraints for transitions ($TC1$).

t_1	[0,0]	t_7	[0,0]	t_{13}	[16,24]	t_{19}	[0,0]
t_2	$[260, 300]^{\dagger}$	t_8	[0,0]	t_{14}	[0,10]	t_{20}	[0,0]
t_3	[0,0]	t_9	[50,100]	t_{15}	[50,100]	t_{21}	[0,0]
t_4	[16,24]	t_{10}	[16,24]	t_{16}	[16,24]	t_{22}	[0,0]
t_5	[0,10]	t_{11}	[0,10]	t_{17}	[50,53]		
t_6	[0,0]	t_{12}	[50,100]	t_{18}	[0,0]		

† [80i+180,80i+220] for station i

algorithm is exponential in n. Since all stations operate more or less independently, parallelism also increases with n; therefore, the partial order method succeeds in reducing the complexity. This result is typical for a number of similar examples.

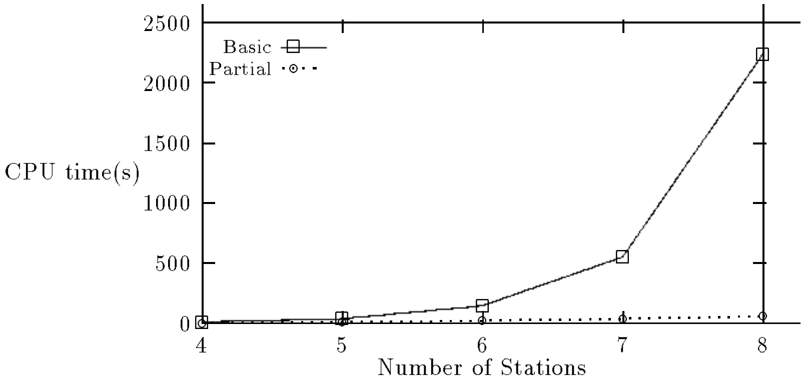

Figure 5: Performance of the basic/partial order methods

7 Conclusion

In this paper, we have proposed a timed temporal logic for time Petri nets which is expressive enough to formalize quantitative timing properties and yet it is stuttering invariant, so that the parallelism in the nets can be used to avoid the state explosion problem during verification.

Then, we have developed a model checking algorithm for our logic. We constructed for the infinite state space of the net a finite representation, the atom graph, such that every atom sequence represents a set of runs, and satisfies the formula iff the corresponding runs satisfy the formula.

Since the complexity of the consequent model checking algorithm depends on the number of atoms, we have shown how to reduce this number by elimination of redundant interleavings. In our method, for every firable transition all dependent sets, i.e., sets of firable transitions whose firings are relevant for the evaluation of the given formula, are computed. From the smallest set of firable transitions which is closed under dependency the reduced atom graph is generated. Since this set is usually much smaller than the set of all firable transitions, a considerable reduction of the state space is achieved.

Although the worst case complexity of the problem is exponential, experimental results from several examples show that the proposed algorithm successfully reduces the average complexity of the model checking.

In the future we intend to combine our method with symbolic model checking techniques (which represent state spaces as binary decision diagrams), and to find similar efficient model checking algorithms for other kinds of temporal logics such as branching time temporal logics and timed μ–calculi.

Acknowledgment:

We would like to thank E. Clarke for his contribution to the preliminary version of this paper and his continuous support and encouragement. We would also like to thank K. McMillan for many helpful discussions, and A. Shibayama for his help in implementing the proposed method.

References

[ACD90] R. Alur, C. Courcoubetis, and D. Dill. Model-checking for real-time systems. *Proc. of 5th IEEE LICS*, 1990.

[AH89] R. Alur and T. A. Henzinger. A really temporal logic. *Proc. of 30th IEEE FOCS*, 1989.

[BCD$^+$92] J. R. Burch, E. M. Clarke, D. L. Dill, L. J. Hwang, and K. L. McMillan. Symbolic model checking: 10^{20} states and beyond. *Academic Press*, 98(2):142–170, 1992.

[BD91] B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time Petri nets. *IEEE Trans. on Software Eng.*, 17(3):259–273, 1991.

[CES86] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. on Programming Languages and Systems*, 8(2):244–263, 1986.

[dBak92] J. W. de Bakker et al. (ed), editor. *Real time - Theory in Practice, Proc. REX Workshop.* Springer LNCS 600, 1992.

[GKPP94] R. Gerth, R. Kuiper, D. Peled, and W. Penczek. A partial order approach to branching time logic model checking. *Internal report*, 1994.

[God90] P. Godefroid. Using partial orders to improve automatic verification methods. *Proc. of Workshop on Computer Aided Verification*, 1990.

[HNSY92] T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *7th IEEE LICS*, 1992.

[JM87] F. Jahanian and A. K. Mok. A graph-theoretic approach for timing analysis and its implementation. *IEEE Trans. Comput.*, C-36(8):961–975, 1987.

[KP90] S. Katz and D. Peled. Defining conditional independence using collapses. *Semantics for concurrency, BCS-FACS Workshop, M. Kwiatkowska (ed.), Springer*, 1990.

[LP85] O. Lichtenstein and A. Pnueli. Checking that finite state concurrent programs satisfy their linear specification. *Proc. 12th POPL*, pages 97–107, 1985.

[MF76] P. Merlin and D. J. Faber. Recoverability of communication protocols. *IEEE Trans. on Communication*, COM-24(9), 1976.

[RB86] J-L. Roux and B. Berthomieu. Verification of a local area network protocol with Tina, a software package for time Petri nets. *7th European Workshop on Application and Theory of Petri Nets*, pages 183–205, 1986.

[Sta90] P. Starke. *Analyse von Petri-Netz Modellen.* Teubner, Stuttgart, 1990.

[Val90] A. Valmari. A stubborn attack on state explosion. *Proc. of Workshop on Computer-Aided Verification*, 1990.

[YNT89] T. Yoneda, K. Nakade, and Y. Tohma. A fast timing verification method based on the independence of units. *Proc. of 19th International Symposium on Fault-tolerant Computing*, pages 134–141, 1989.

[YTK91] T. Yoneda, Y. Tohma, and Y. Kondo. Acceleration of timing verification method based on time Petri nets. *Systems and Computers in Japan*, 22(12):37–52, 1991.