Undecidability of Coverability and Boundedness for Timed-Arc Petri Nets with Invariants

Lasse Jacobsen, Morten Jacobsen, Mikael H. Møller
{lassejac,mortenja,mikaelhm}@cs.aau.dk

Department of Computer Science
Aalborg University

14. november 2009
Example: Petri Net

![Petri Net Diagram](image)

- **Light** (off)
- **Light** (on)
- **Reset** (timer)
- **Turn light on**
- **Turn light off**

Transition: $M_0 \xrightarrow{\text{Turn light on}} M_1 \xrightarrow{d} M_2$
Example: Timed-Arc Petri Net

\[
\begin{align*}
M_0 & \xrightarrow{\text{Turn}_{\text{light off}}}[5,5] M_1 \\
M_1 & \xrightarrow{\text{Turn}_{\text{light on}}}[0,\infty) M_0 \\
M_1 & \xrightarrow{\text{Reset}_{\text{timer}}}[0,5] M_2 \\
M_2 & \xrightarrow{\text{Reset}_{\text{timer}}}[5,5] M_1
\end{align*}
\]
Example: Timed-Arc Petri Net

\[ \text{Turn}_{\text{light on}} \]

\[ \text{Turn}_{\text{light off}} \]

\[ \text{Light}_{\text{off}} \quad [0, \infty) \]

\[ \text{Light}_{\text{on}} \quad [0, 5) \]

\[ \text{Reset}_{\text{timer}} \]

\[ M_0 \xrightarrow{\text{Turn}_{\text{light on}}} M_1 \]
Example: Timed-Arc Petri Net

M₀ →

Turn light on → M₁
d : 5 → M₂

Turn light off

[0, ∞)

[0, 5)

[5, 5]

Reset timer

Light on

Light off

Light on

Light off

Turn light on

Turn light off

Overview

Introduction

Boundedness and Coverability

Undecidability

Our Results
Timed-Arc Petri Net was first introduced by T. Bolognesi, F. Lucidi, and S. Trigila in “From timed Petri nets to timed LOTOS”, 1990.

Overview


Reachability was shown **undecidable** by V. V. Ruiz, F. C. Gomez, and D. de Frutos-Escrig in “On Non-Decidability of Reachability for Timed-Arc Petri Nets“, 1999.

Reachability was shown **undecidable** by V. V. Ruiz, F. C. Gomez, and D. de Frutos-Escrig in “On Non-Decidability of Reachability for Timed-Arc Petri Nets“, 1999.

Coverability was shown **decidable** by P. A. Abdulla and A. Nylen in ”Timed Petri Nets and BQOs“, 2001.
Overview

Timed-Arc Petri Net was first introduced by T. Bolognesi, F. Lucidi, and S. Trigila in “From timed Petri nets to timed LOTOS”, 1990.

Reachability was shown **undecidable** by V. V. Ruiz, F. C. Gomez, and D. de Frutos-Escrig in “On Non-Decidability of Reachability for Timed-Arc Petri Nets“, 1999.

Coverability was shown **decidable** by P. A. Abdulla and A. Nylen in ”Timed Petri Nets and BQOs“, 2001.

Boundedness was shown **decidable** by P. A. Abdulla, P. Mahata, and R. Mayr in “Dense-Timed Petri Nets: Checking Zenoness, Token liveness and Boundedness.“, 2007.
Introduction

Our Results

Undecidability

Example

Overview

Introduction

Boundedness and Coverability

www.tapaal.net
Example: Timed-Arc Petri Net with Invariants

\[ M_0 \xrightarrow{\text{Turn}_{\text{light}} \text{on}} M_1 \xrightarrow{d:5} M_2 \]
Boundedness

A marked ITAPN \((N, M_0)\) is said to be \textit{bounded}, if there exist a number \(k\) such that the number of tokens at any place in any reachable marking is bounded by \(k\).
Boundedness and Coverability

**Boundedness**

A marked ITAPN \((N, M_0)\) is said to be *bounded*, if there exist a number \(k\) such that the number of tokens at any place in any reachable marking is bounded by \(k\).

**Coverability**

A marking \(M\) is said to be *coverable* from \(M_0\) if there exists a reachable marking \(M'\) which covers \(M\).
Our results

- Boundedness is undecidable for ITAPN
- Coverability is undecidable for ITAPN.

The idea is to make a reduction from a Minsky Two-counter Machine.
Definition: Minsky Two-counter Machine

A Minsky Two-counter Machine (2-CM) with two non-negative registers \( r_1 \) and \( r_2 \), is a sequence of instructions,

\[
\begin{align*}
I_1 : & \text{Ins}_1 \\
I_2 : & \text{Ins}_2 \\
& \quad \vdots \\
I_e : & \text{HALT}
\end{align*}
\]

where each instruction \( \text{Ins}_j \) is one of the two types:

**Inc** \( r_i := r_i + 1; \text{goto } I_k; \)
where \( i \in \{1, 2\} \) and \( k \in \{1, 2, \ldots, e\} \).

**TD** if \( r_i > 0 \) then \( r_i := r_i - 1; \text{goto } I_k; \) else \( \text{goto } I_\ell; \)
where \( i \in \{1, 2\} \) and \( k, \ell \in \{1, 2, \ldots, e\} \).
The halting problem for 2-CM was shown to be undecidable by M. L. Minsky in 1967.
The halting problem for 2-CM was shown to be undecidable by M. L. Minsky in 1967.

The basics of the reduction is:

- A place $p_{r_i}$ for each register, the number of tokens of age zero simulates the value of $r_i$. 
The halting problem for 2-CM was shown to be undecidable by M. L. Minsky in 1967.

The basics of the reduction is:

- A place $p_{r_i}$ for each register, the number of tokens of age zero simulates the value of $r_i$.
- A place $p_j$ for each instruction, simulating which instruction is about to be executed.
The halting problem for 2-CM was shown to be undecidable by M. L. Minsky in 1967.

The basics of the reduction is:

- A place $p_{ri}$ for each register, the number of tokens of age zero simulates the value of $r_i$.
- A place $p_j$ for each instruction, simulating which instruction is about to be executed.
- A place $p_{count}$ to count the number of instructions performed.
Simulation of the registers and the Halt instruction

Simulation of register \( r_1 \) with value 2.

\[ p_{r_1}^{\text{reset}} \quad [0, 0] \quad t_{r_1}^{\text{reset}} \quad [1, 1] \quad p_{r_1} \quad \{0, 0\} \]

\[ p_{r_1}^{\text{reset}} \quad [0, 0] \quad t_{r_1}^{\text{reset}} \quad [1, 1] \quad p_{r_1} \quad \{0, 0\} \]
Simulation of the registers and the Halt instruction

Simulation of register $r_1$ with value 2.

The HALT instruction is simulated by:
Simulation of an increment instructions

\[ l_j: r_1 := r_1 + 1; \text{ goto } l_k; \]
Simulation of an increment instructions

\( l_j : r_1 := r_1 + 1; \text{ goto } l_k; \)
Simulation of an increment instructions

\[ l_j: r_1 := r_1 + 1; \text{ goto } l_k; \]
Simulation of an increment instructions

\[ l_j: r_1 := r_1 + 1; \text{ goto } l_k; \]
Simulation of an increment instructions

\[ l_j: \ r_1 := r_1 + 1; \ \text{goto} \ l_k; \]
Simulation of an increment instructions

\[ l_j: r_1 := r_1 + 1; \text{ goto } l_k; \]
Simulation of an increment instructions

\[ l_j: \ r_1 := r_1 + 1; \ \text{goto} \ l_k; \]
Simulation of a test & decrement instructions

\[ l_j: \text{ if } r_1 > 0 \text{ then } r_1 := r_1 - 1; \text{ goto } l_k; \text{ else goto } l_\ell; \]
Simulation of a test & decrement instructions

$l_j$: if $r_1 > 0$ then $r_1 := r_1 - 1$; goto $l_k$; else goto $l_\ell$;
Simulation of a test & decrement instructions

\[ l_j: \text{if } r_1 > 0 \text{ then } r_1 := r_1 - 1; \text{ goto } l_k; \text{ else goto } l_\ell; \]
Simulation of a test & decrement instructions

\( l_j: \) if \( r_1 > 0 \) then \( r_1 := r_1 - 1; \) goto \( l_k; \) else goto \( l_\ell; \)
Simulation of a test & decrement instructions

\[ l_j: \text{if } r_1 > 0 \text{ then } r_1 := r_1 - 1; \text{ goto } l_k; \text{ else goto } l_\ell; \]
Simulation of a test & decrement instructions

\[ l_j: \text{ if } r_1 > 0 \text{ then } r_1 := r_1 - 1; \text{ goto } l_k; \text{ else goto } l_\ell; \]
Simulation of a test & decrement instructions

\[ I_j: \text{ if } r_1 > 0 \text{ then } r_1 := r_1 - 1; \text{ goto } I_k; \text{ else goto } I_\ell; \]
Simulation of a test & decrement instructions

\[ l_j: \text{ if } r_1 > 0 \text{ then } r_1 := r_1 - 1; \text{ goto } l_k; \text{ else goto } l_\ell; \]
Lemma

Given a 2-CM $CM$ and the associated ITAPN $(N, M_0)$,

$$CM \text{ halts } \iff N \text{ is bounded}.$$
Undecidability of Boundedness

Lemma
Given a 2-CM CM and the associated ITAPN \((N, M_0)\),

\[ CM \text{ halts } \iff N \text{ is bounded}. \]

Theorem
The *Boundedness* problem is undecidable for Timed-Arc Petri Nets with Invariants.
Lemma

Let $M$ be a marking such that there is a token at $p_{\text{halt}}$ and none in any other place. Given a 2-CM $CM$ and the associated marked ITAPN $(N, M_0)$,

$$CM \text{ halts} \iff M \text{ is coverable from } M_0.$$
Lemma
Let $M$ be a marking such that there is a token at $p_{halt}$ and none in any other place. Given a 2-CM $CM$ and the associated marked ITAPN $(N, M_0)$,

$$CM \text{ halts } \iff M \text{ is coverable from } M_0.$$

Theorem
The Coverability problem is undecidable for Timed-Arc Petri Nets with Invariants.
Conclusion

We have shown that *Boundedness* and *coverability* is both *undecidable* in Timed-arc Petri Nets with Invariants.

<table>
<thead>
<tr>
<th></th>
<th>Reachability</th>
<th>Boundedness</th>
<th>Coverability</th>
</tr>
</thead>
<tbody>
<tr>
<td>PN</td>
<td>decidable</td>
<td>decidable</td>
<td>decidable</td>
</tr>
<tr>
<td>TAPN</td>
<td>undecidable</td>
<td>decidable</td>
<td>decidable</td>
</tr>
<tr>
<td>ITAPN</td>
<td>undecidable</td>
<td><em>undecidable</em></td>
<td><em>undecidable</em></td>
</tr>
</tbody>
</table>
Discussion

How does TAPAAL deal with this result.

- TAPAAL can verify $k$ – *boundedness* as it is decidable.
- TAPAAL supports optimization of the bound $k$ if the net is bounded.
- If unbounded, then TAPAAL makes an under approximation.

www.tapaal.net