Petri Nets

Rebecca Albrecht

Seminar: **Automata Theory**

Chair of Software Engeneering

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Overview

1. **Motivation:**
   - Why not just using finite automata for everything?
   - What are Petri Nets and when do we use them?
2. **Introduction:**
   - How do Petri Nets work?
3. **Languages:**
   - What is the general expressiveness power of Petri Nets?
4. **Decidability and Complexity:**
   - What is the complexity of certain decision problems?
   - What if we assure certain conditions?

Overview

1. **Motivation:**
   - Why not just using finite automata for everything?
   - What are Petri Nets and when do we use them?
2. **Introduction:**
3. **Languages:**
4. **Decidability and Complexity:**

Motivation

Consider:

- Two groups of workers produce goods of type **A** and type **B**.
- Two goods of type A and type B can be combined into type **C**.

The corresponding language:

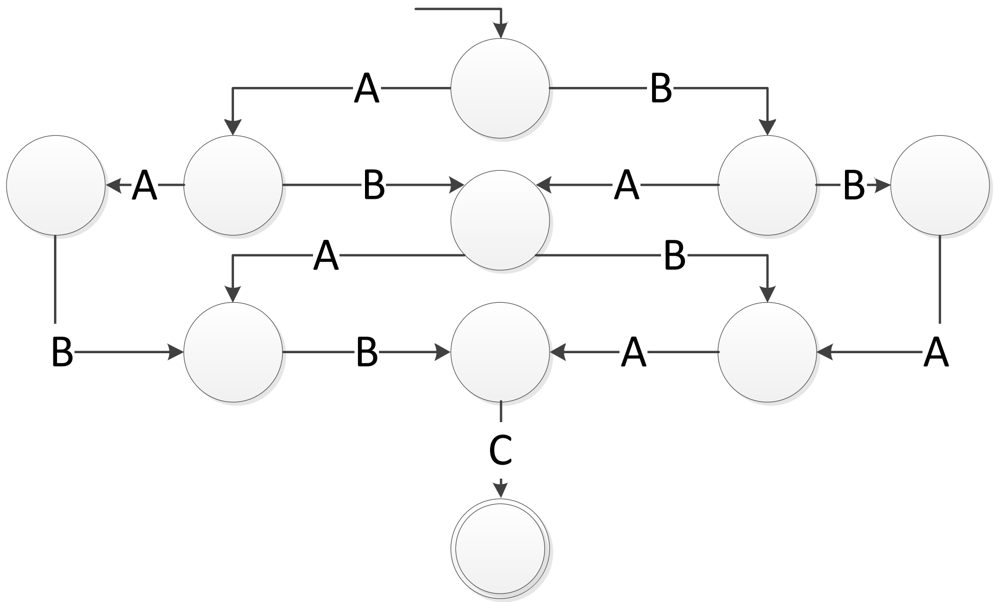
$L = \{aabbc, ababc, abbac, babac, bbaac, baabc\}$

Motivation

The corresponding language:

$L = \{aabbc, ababc, abbac, babac, bbaac, baabc\}$

The corresponding finite automaton:



Problem Analysis

The problem:

- A and B are produced **concurrently** and **asynchronous.**
- The production has to be **synchronized** when producing a C**.**

The solution:

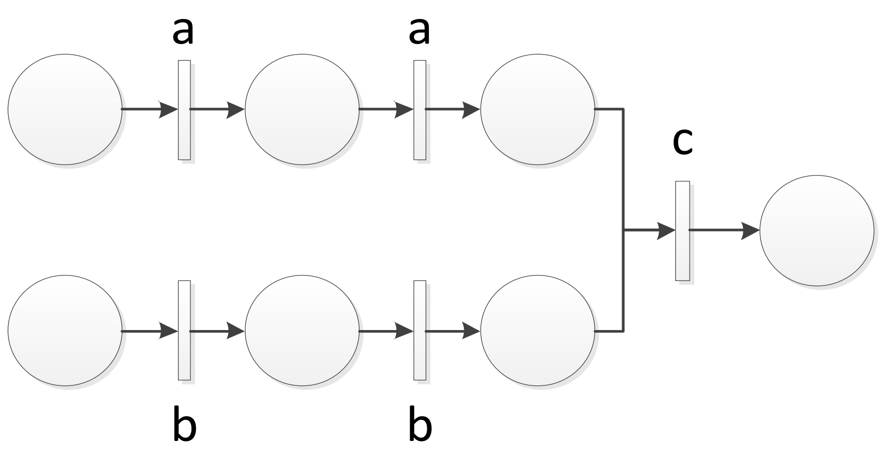
- Automaton model which allows for expressing these properties of a system or process more concise – **Petri Nets.**

Possible Solution

The corresponding language:

$L = \{aabbc, ababc, abbac, babac, bbaac, baabc\}$

A Petri Net for the given problem:

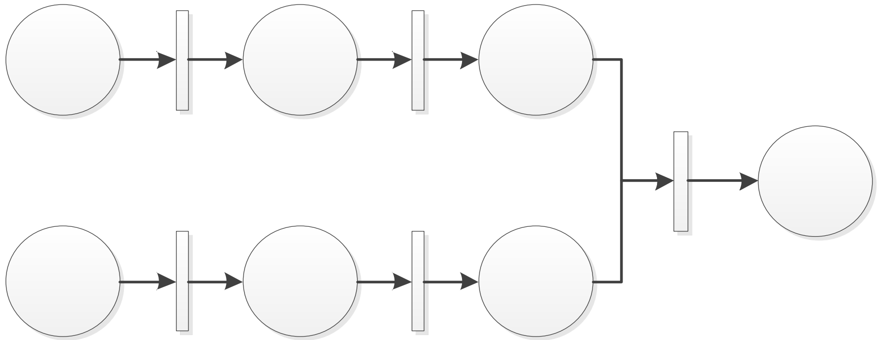


Overview

1. Motivation:
2. **Introduction:**
   - How do Petri Nets work?
3. Languages:
4. Decidability and Complexity:

Structural Definitions

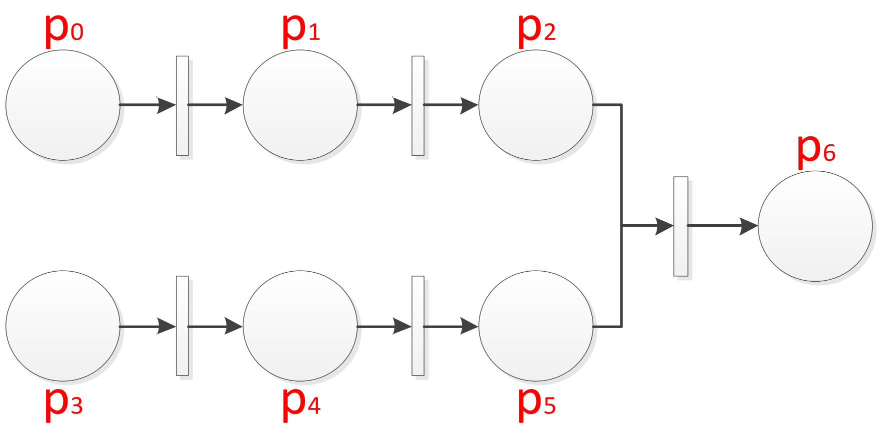
A Petri Net for the given problem:



$$N = (P, T, F)$$

Structural Definitions

A Petri Net for the given problem:

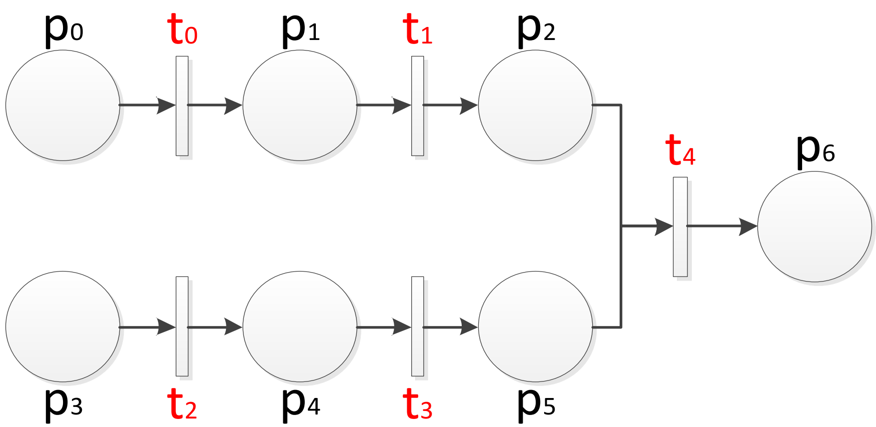


$$N = (P, T, F)$$

$$P = \{p_0, \ldots, p_6\}$$

Structural Definitions

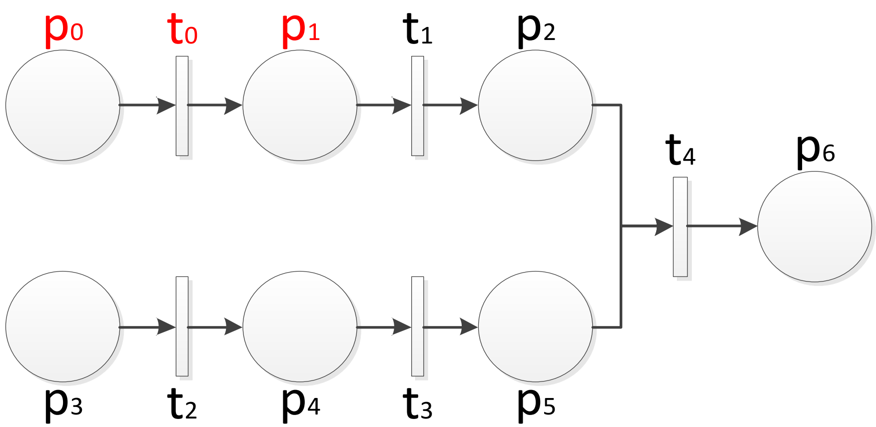
A Petri Net for the given problem:



$$N = (P, T, F)$$
$$P = \{p_0, \dots, p_6\}$$
$$T = \{t_0, \dots, t_4\}$$

Structural Definitions

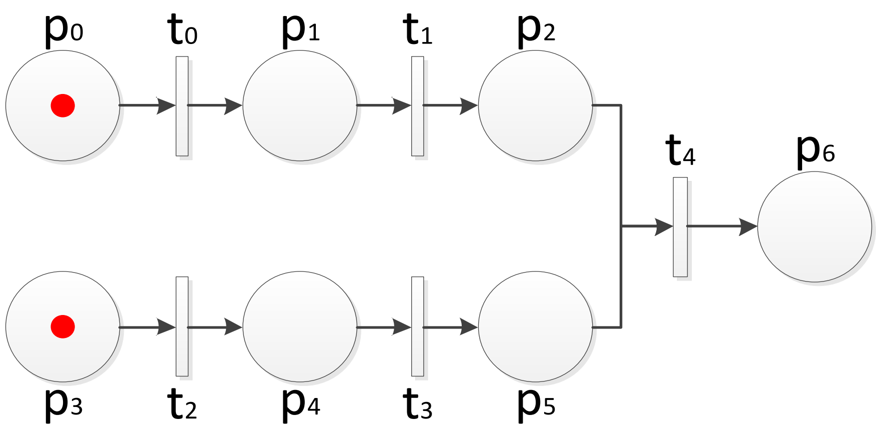
A Petri Net for the given problem:



$$N = (P, T, F)$$
$$P = \{p_0, \dots, p_6\}$$
$$T = \{t_0, \dots, t_4\}$$
$$F \subset (P \times T) \cup (T \times P)$$

Structural Definitions

A Petri Net for the given problem:



$$N = (P, T, F)$$

$$P = \{p_0, \ldots, p_6\}$$

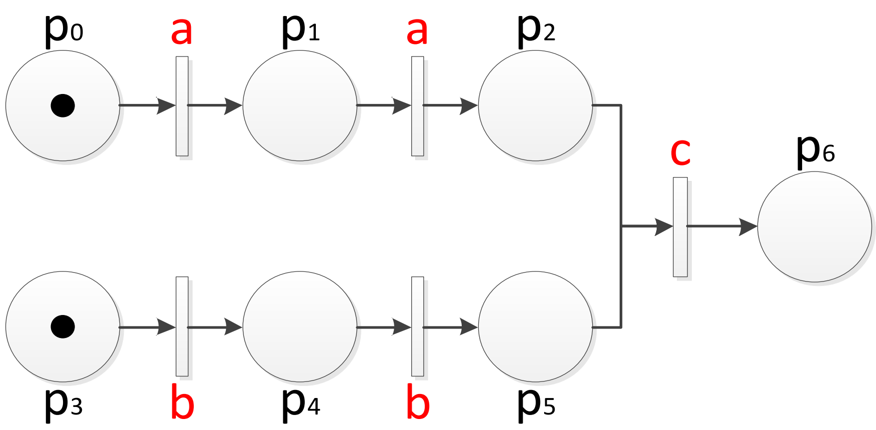
$$T = \{t_0, \ldots, t_4\}$$

$$F \subset (P \times T) \cup (T \times P)$$

$$m: P \rightarrow \mathbb{N}$$

Structural Definitions

A Petri Net for the given problem:



$$N = (P, T, F)$$

$$P = \{p_0, \dots, p_6\}$$

$$T = \{t_0, \dots, t_4\}$$

$$F \subset (P \times T) \cup (T \times P)$$

$$m: P \rightarrow \mathbb{N}$$

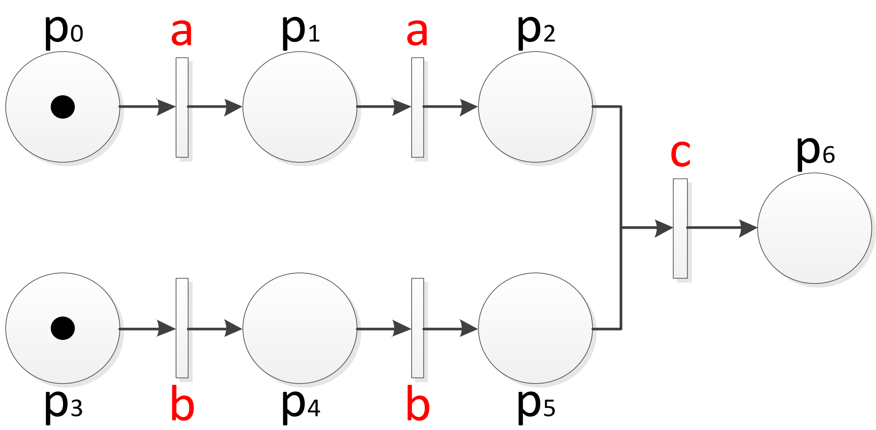
$$\Sigma = \{a, b, c\}$$

$$l: T \rightarrow \Sigma$$

Behavior

$L = \{aabbc, ababc, abbac, babac, bbaac, baabc\}$

A Petri Net for the given language:



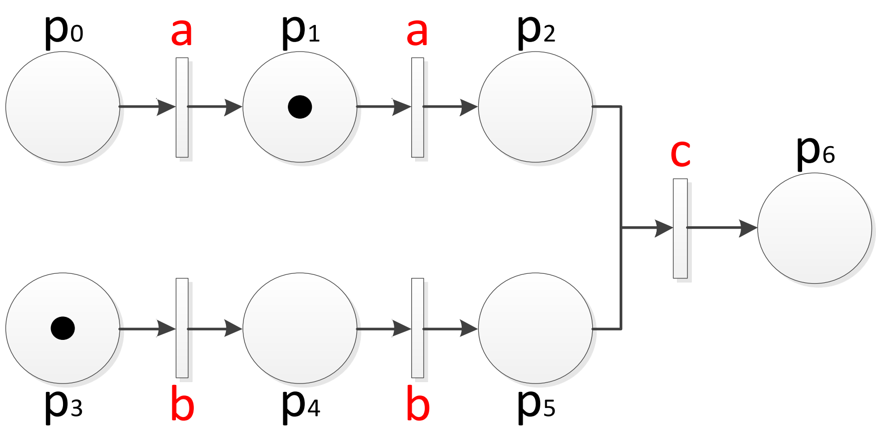
$P = (p_0, \ldots, p_6)$

$M_0 = (1, 0, 0, 1, 0, 0, 0)$

Behavior

$L = \{aabbc, ababc, abbac, babac, bbaac, baabc\}$

A Petri Net for the given language:



$P = (p_0, \ldots, p_6)$

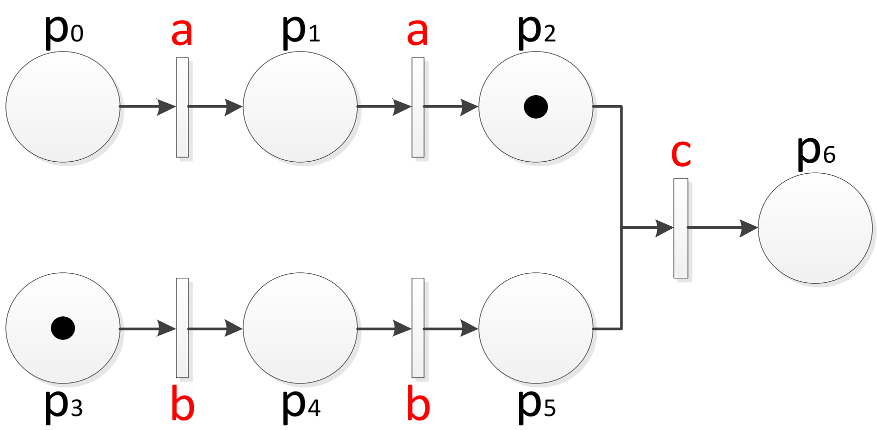
$M_0 = (1, 0, 0, 1, 0, 0, 0)$

$M_1 = (0, 1, 0, 1, 0, 0, 0)$

Behavior

$L = \{aabbc, ababc, abbac, babac, bbaac, baabc\}$

A Petri Net for the given language:



$P = (p_0, \ldots, p_6)$

$M_0 = (1, 0, 0, 1, 0, 0, 0)$

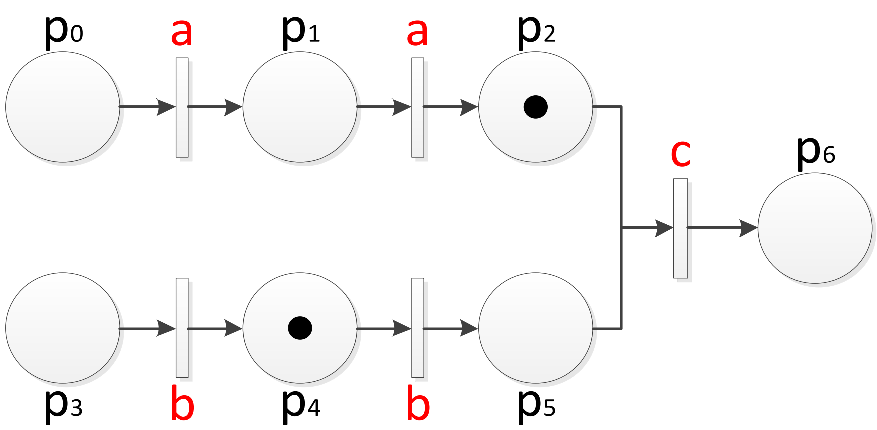
$M_1 = (0, 1, 0, 1, 0, 0, 0)$

$M_2 = (0, 0, 1, 1, 0, 0, 0)$

Behavior

$L = \{aabbc, ababc, abbac, babac, bbaac, baabc\}$

A Petri Net for the given language:



$P = (p_0, \ldots, p_6)$

$M_0 = (1, 0, 0, 1, 0, 0, 0)$

$M_1 = (0, 1, 0, 1, 0, 0, 0)$

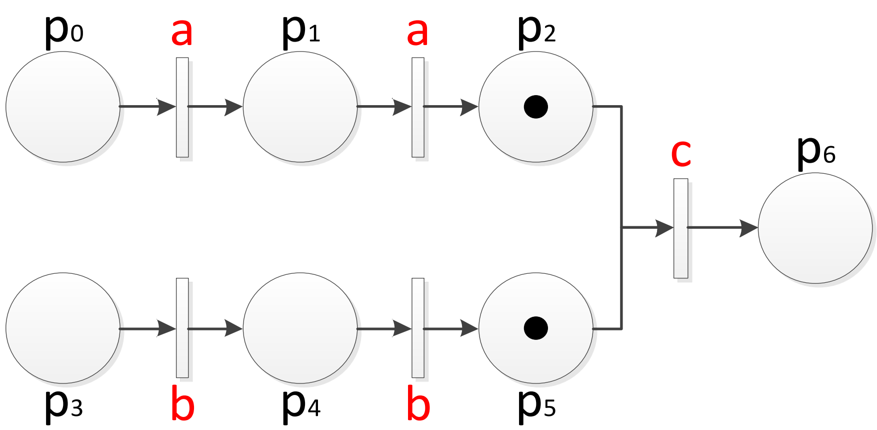
$M_2 = (0, 0, 1, 1, 0, 0, 0)$

$M_3 = (0, 0, 1, 0, 1, 0, 0)$

Behavior

$L = \{aabbc, ababc, abbac, babac, bbaac, baabc\}$

A Petri Net for the given language:



$P = (p_0, \dots, p_6)$

$M_0 = (1, 0, 0, 1, 0, 0, 0)$

$M_1 = (0, 1, 0, 1, 0, 0, 0)$

$M_2 = (0, 0, 1, 1, 0, 0, 0)$

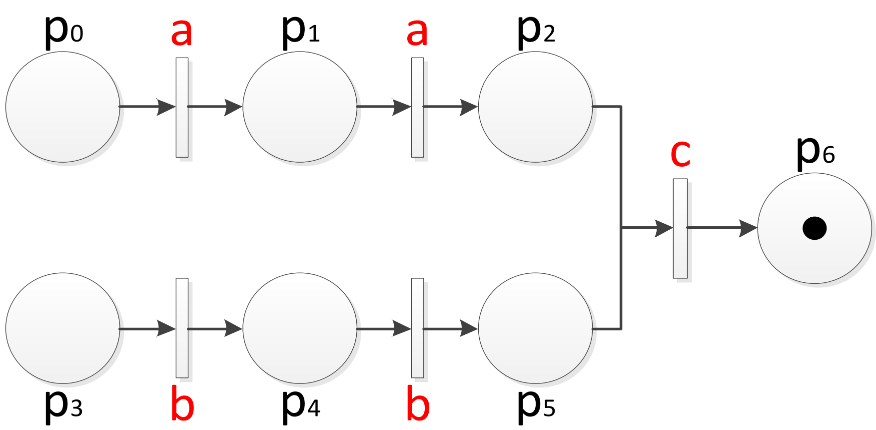
$M_3 = (0, 0, 1, 0, 1, 0, 0)$

$M_4 = (0, 0, 1, 0, 0, 1, 0)$

Behavior

$L = \{aabbc, ababc, abbac, babac, bbaac, baabc\}$

A Petri Net for the given language:



$P = (p_0, \ldots, p_6)$

$M_0 = (1, 0, 0, 1, 0, 0, 0)$

$M_1 = (0, 1, 0, 1, 0, 0, 0)$

$M_2 = (0, 0, 1, 1, 0, 0, 0)$

$M_3 = (0, 0, 1, 0, 1, 0, 0)$

$M_4 = (0, 0, 1, 0, 0, 1, 0)$

$M_5 = (0, 0, 0, 0, 0, 0, 1)$

Overview

1. Motivation:
2. Introduction:
3. **Languages:**
   - What is the general expressiveness power of Petri Nets?
4. Decidability and Complexity:

Language Acceptance

Definition:

$\mathcal{N} = (N, M^I, M^F, l)$ with,

- $N = (P, T, F)$ a petri net,
- $l: T \rightarrow \Sigma$ a labeling function, Σ a non-empty alphabet,
- and M^I, M^F finite sets of initial and final markings respectively,

accepts a word $w \in \Sigma^*$ if there exists a firing sequence

$\tau = t_1, \dots, t_n$ from $m_i \in M^I$ to $m_j \in M^F$.

Petri Net Languages

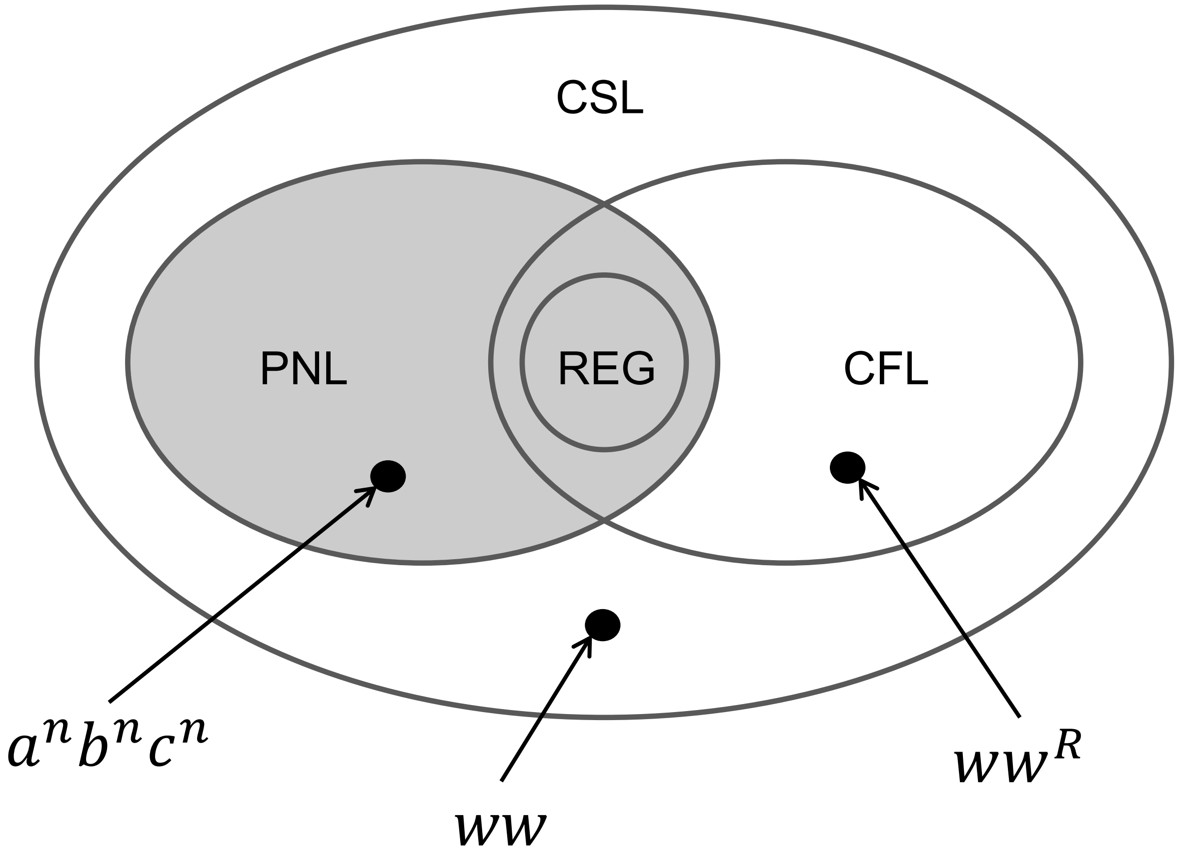
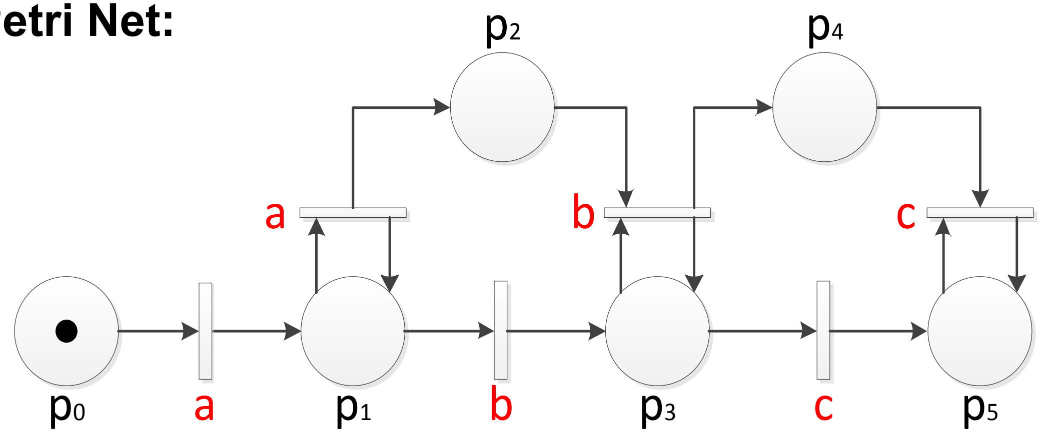


Figure taken from [3]

Languge - Example

Let $\Sigma = \{a, b, c\},\ L = \{a^n b^n c^n |\ n > 0\}$

Petri Net:



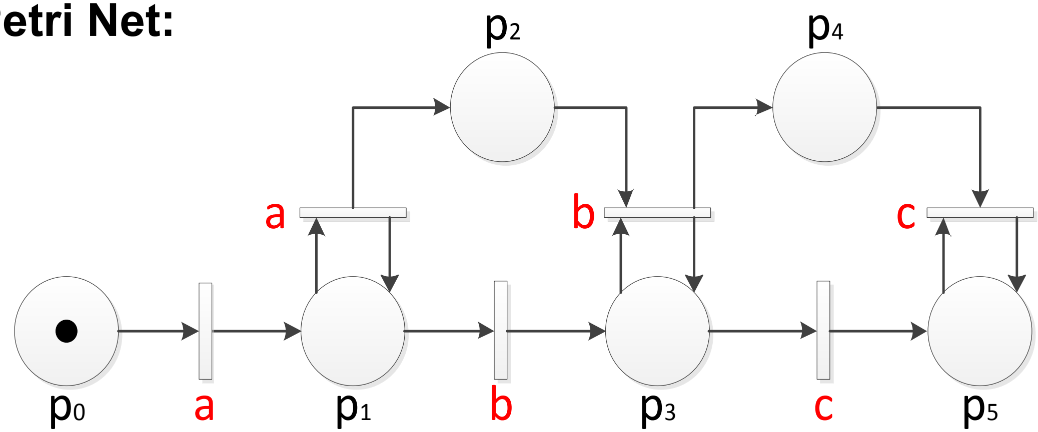
- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Languge - Example

Let $\Sigma = \{a, b, c\},\ L = \{a^n b^n c^n |\, n > 0\}$

Consider: $aaabbbccc$

Petri Net:



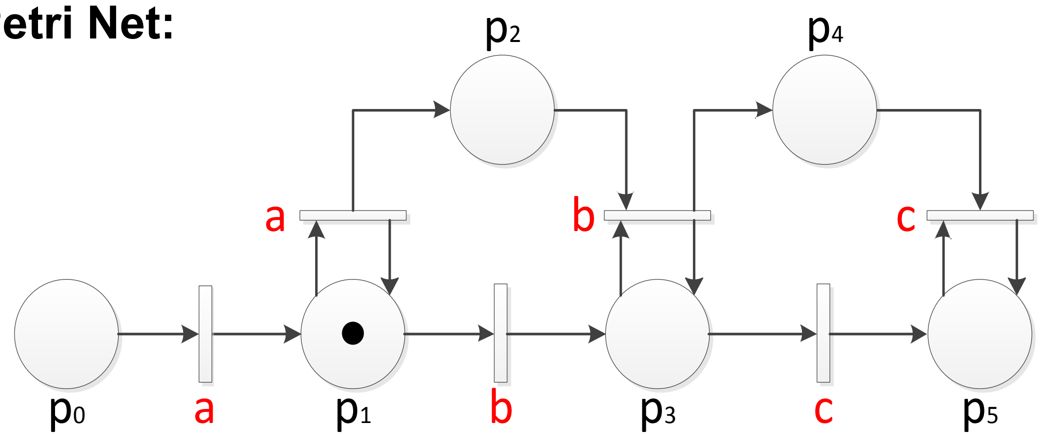
- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Languge - Example

Let $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n | \, n > 0\}$

Consider: $aaabbbccc$

Petri Net:



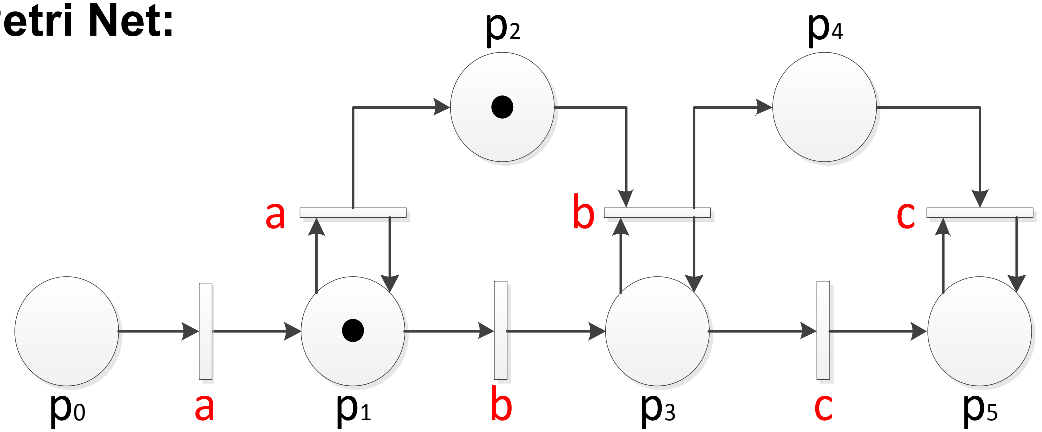
- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Languge - Example

Let $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n | \, n > 0\}$

Consider: $aaabbbccc$

Petri Net:



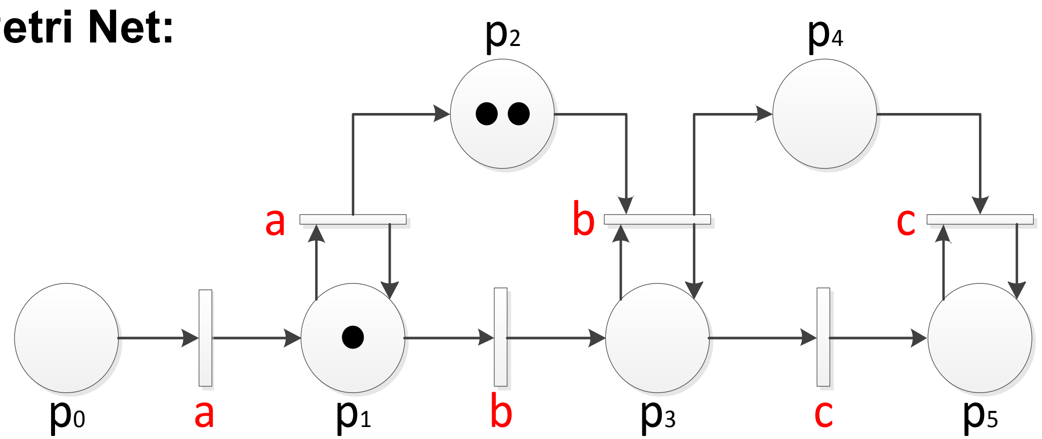
- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Languge - Example

Let $\Sigma = \{a, b, c\}, \ L = \{a^n b^n c^n | \ n > 0\}$

Consider: $aaabbbccc$

Petri Net:



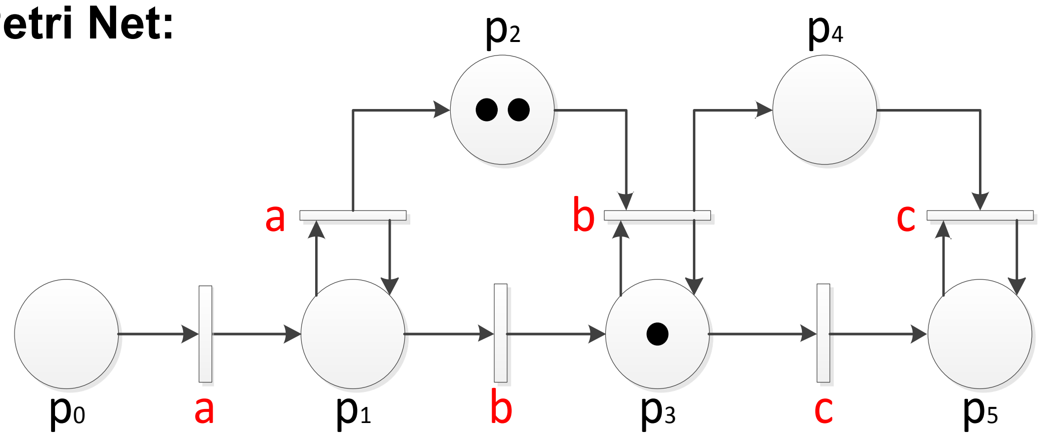
- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Languge - Example

Let $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n \mid n > 0\}$

Consider: $aaabbbccc$

Petri Net:



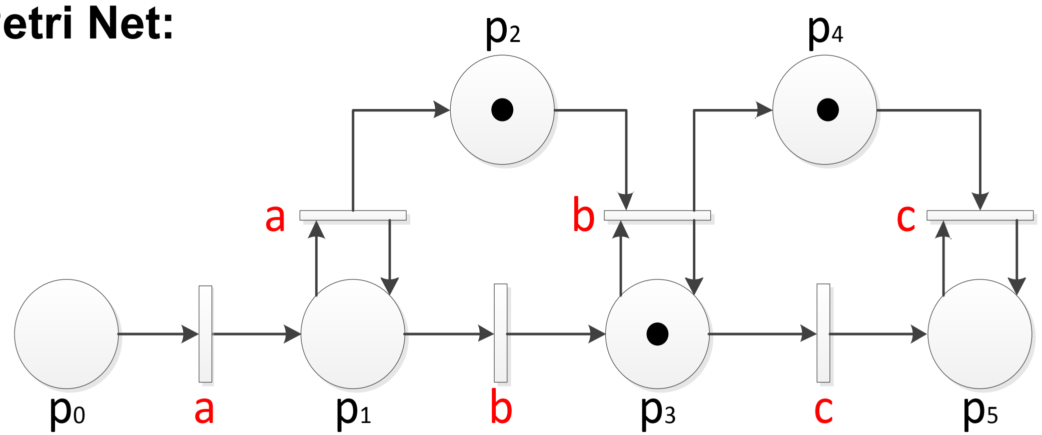
- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Languge - Example

Let $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n | \, n > 0\}$

Consider: $aaabbbccc$

Petri Net:



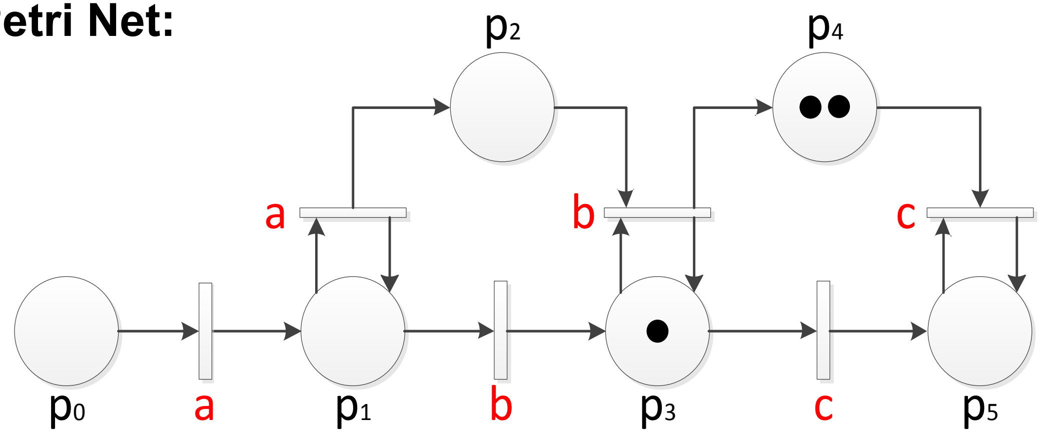
- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Languge - Example

Let $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n | \, n > 0\}$

Consider: $aaabbbccc$

Petri Net:



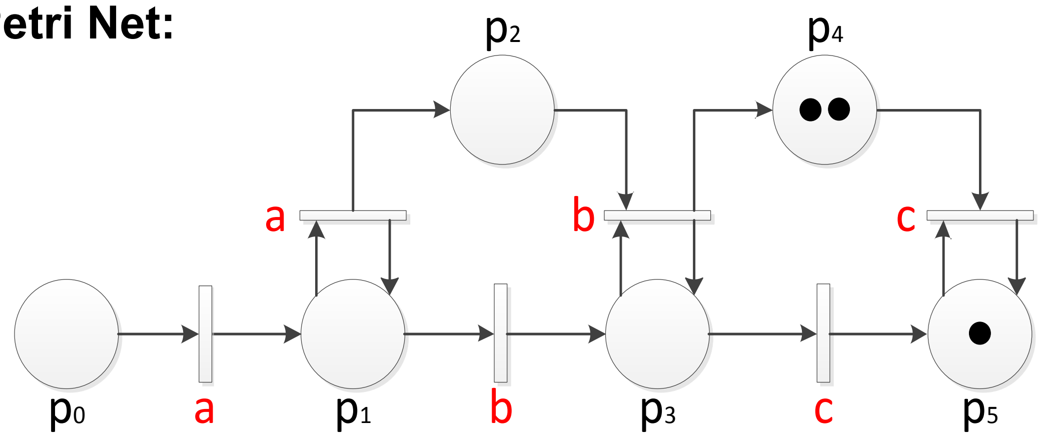
- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Languge - Example

Let $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n | \, n > 0\}$

Consider: $aaabbbccc$

Petri Net:



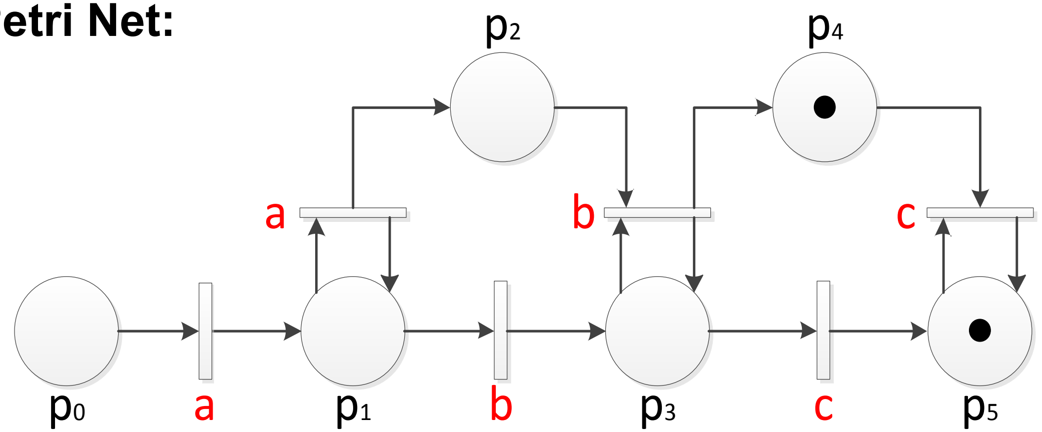
- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Languge - Example

Let $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n | \, n > 0\}$

Consider: $aaabbbccc$

Petri Net:



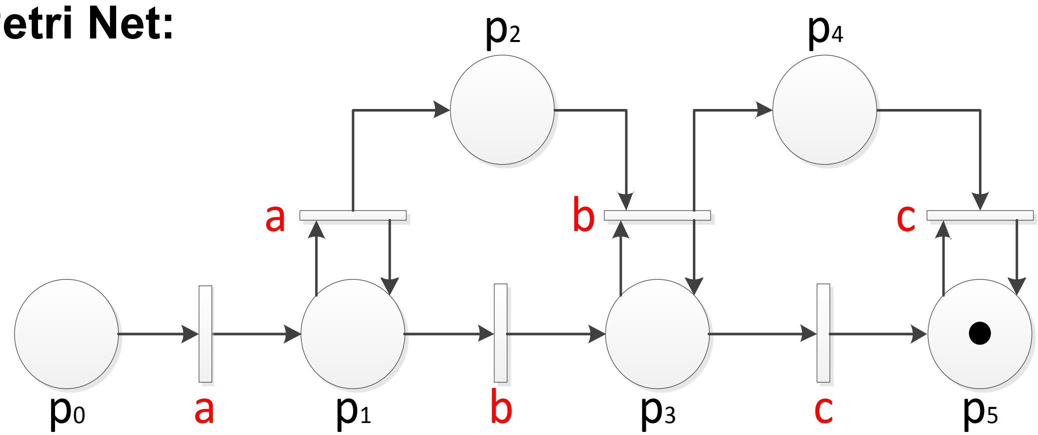
- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Languge - Example

Let $\Sigma = \{a, b, c\},\ L = \{a^n b^n c^n | \, n > 0\}$

Consider: $aaabbbccc$

Petri Net:



- **Initial marking:** (1,0,0,0,0,0)
- **Final marking:** (0,0,0,0,0,1)

Overview

1. Motivation:
2. Introduction:
3. Languages:
4. **Decidability and Complexity:**
   - What is the complexity of certain decision problems?
   - What if we assure certain conditions?

Reachability Problem

Is a certain marking reachable in a Petri Net?

- I.e. can a system reach some bad state?

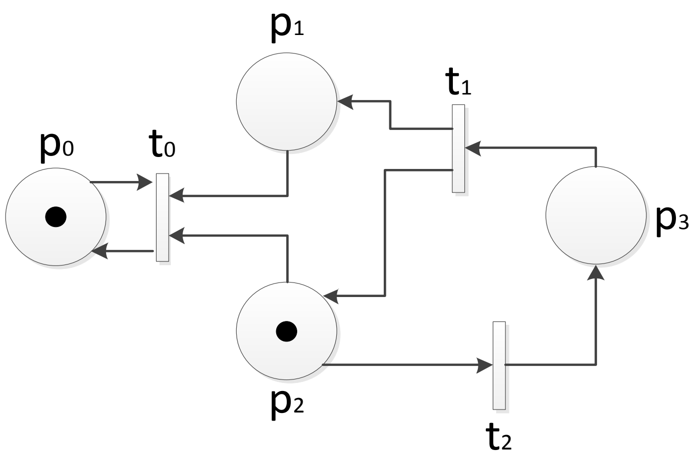
Analysis:

1. What about the reachability problem for Petri Nets in general?
2. Are there certain properties which allow for deciding the reachability problem efficiently?

General analysis tool: The reachability/coverability tree.

Reachability Tree

Petri Net:



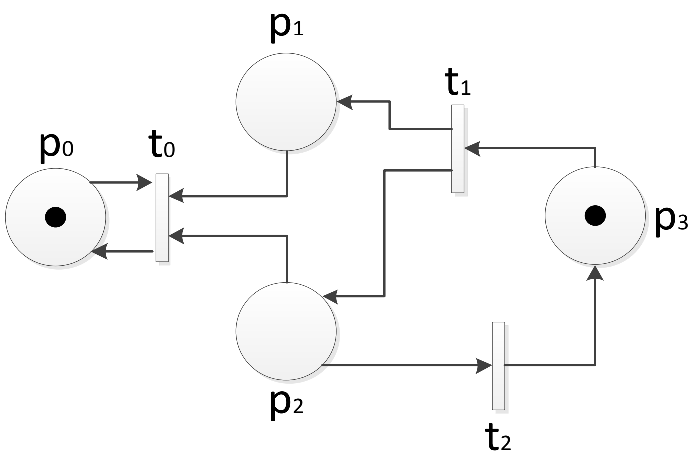
Reachability Tree:

(1, 0, 1, 0)

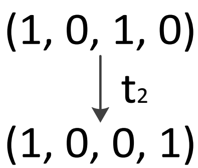
Example taken from [2]

Reachability Tree

Petri Net:



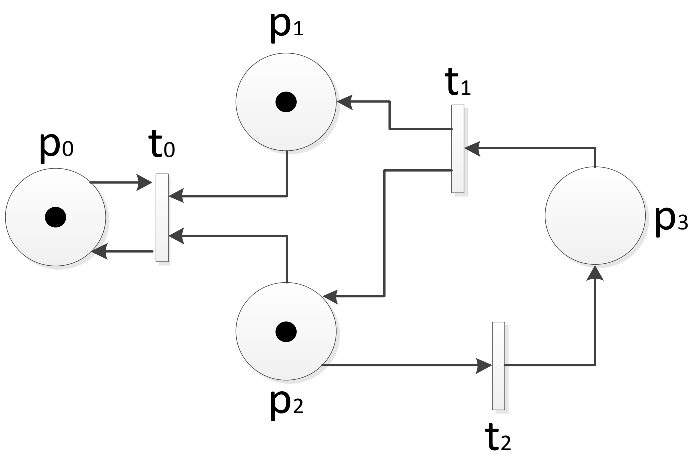
Reachability Tree:



Example taken from [2]

Reachability Tree

Petri Net:



Reachability Tree:

(1, 0, 1, 0)

↓ t_2

(1, 0, 0, 1)

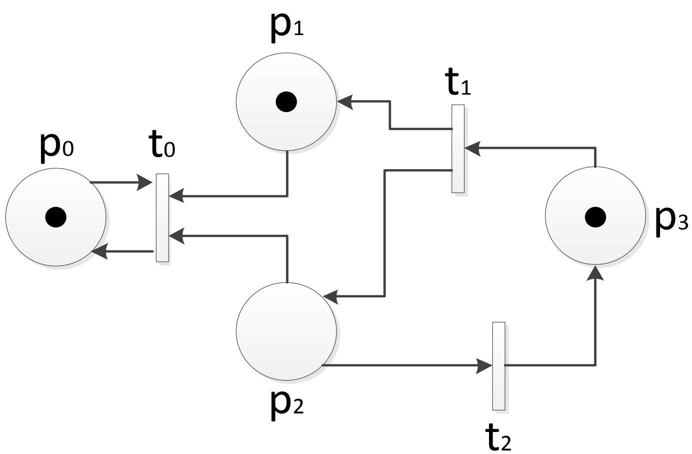
↓ t_1

(1, 1, 1, 0)

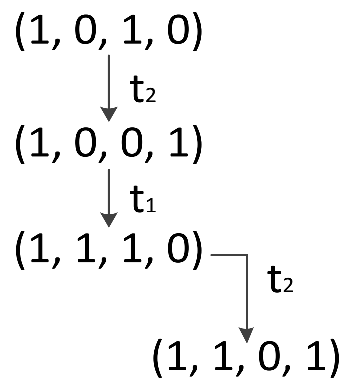
Example taken from [2]

Reachability Tree

Petri Net:



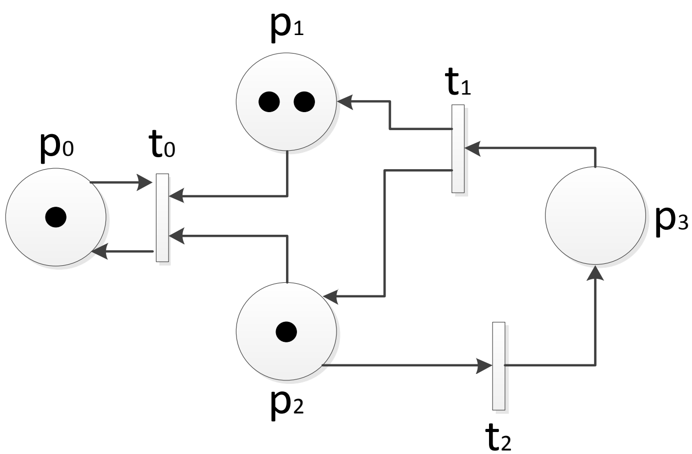
Reachability Tree:



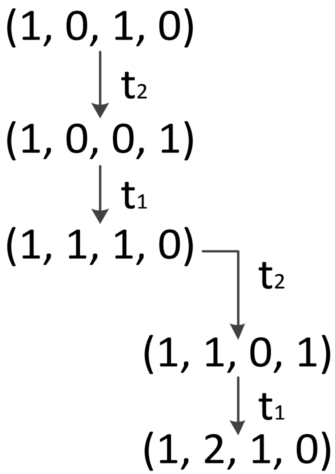
Example taken from [2]

Reachability Tree

Petri Net:



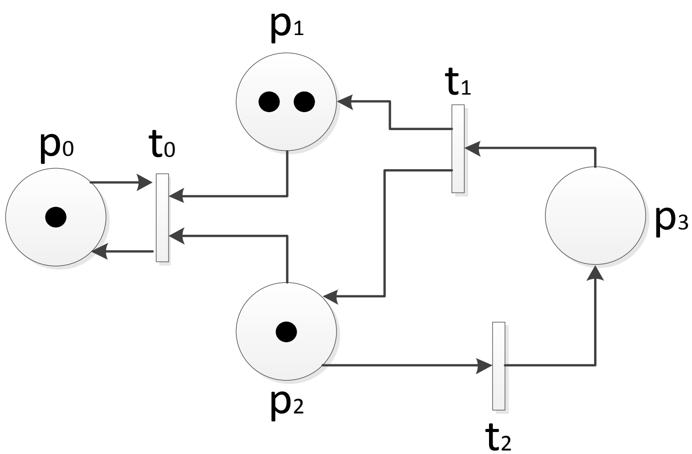
Reachability Tree:



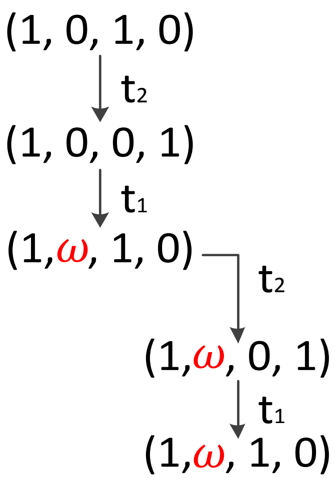
Example taken from [2]

Coverability Tree

Petri Net:



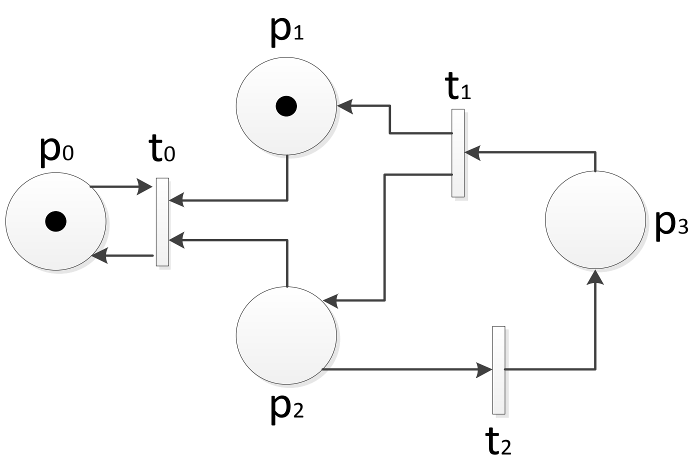
Coverability Tree:



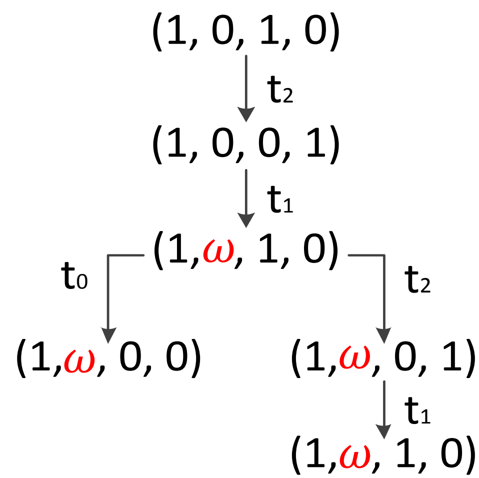
Example taken from [2]

Coverability Tree

Petri Net:



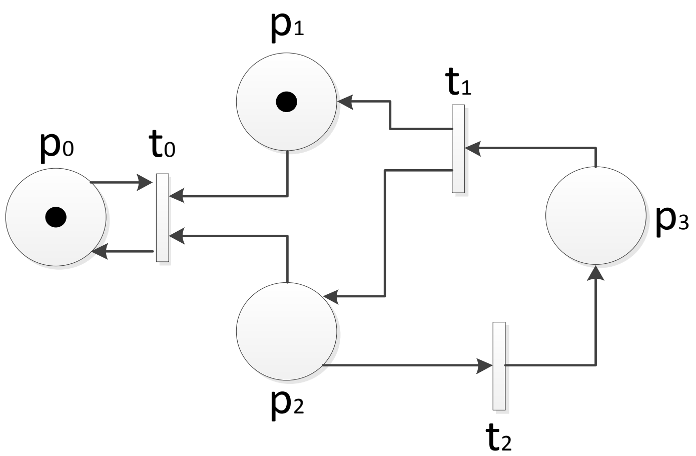
Coverability Tree:



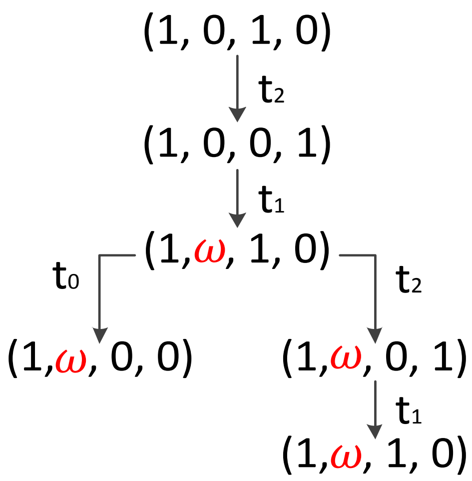
Example taken from [2]

Boundedness

Unbounded Petri Net:



Coverability Tree:



A Petri net ist unbounded if and only if there exists a reachable marking M and a sequence of transitions σ such that:

- $M \rightarrow_{\sigma} M + L$, where L is a non-zero label

Boundedness

A Petri net ist **unbounded** if and only if there exists a reachable marking M and a sequence of transitions σ such that:

$M \rightarrow_{\sigma} \; M + L$, where L is a non-zero label

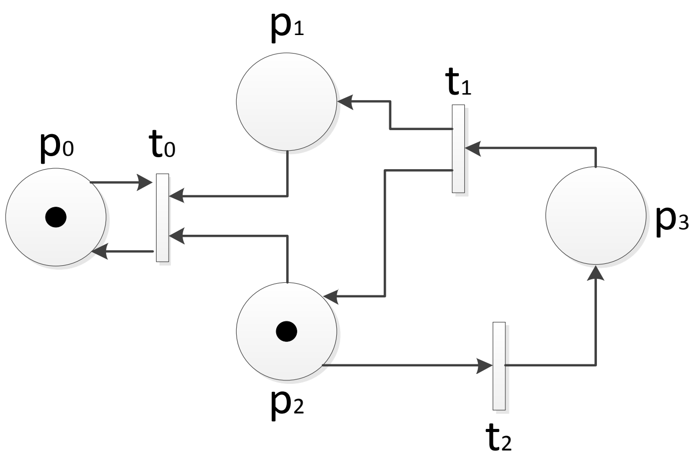
- Decidable in $2^{cn \log n}$

Subclasses of Petri Nets wrt. boundness:

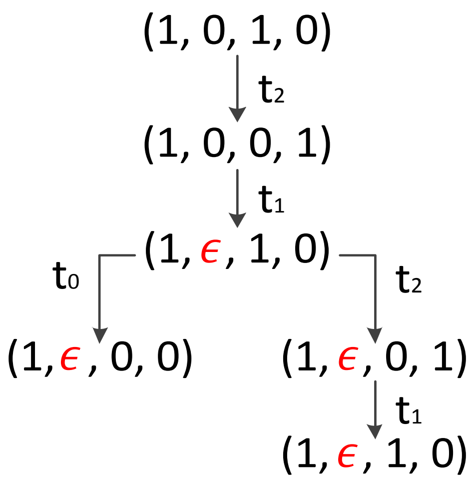
- 1-bounded (or 1-safe)
- k-bounded (or bounded)

Conflicts

Petri Net:

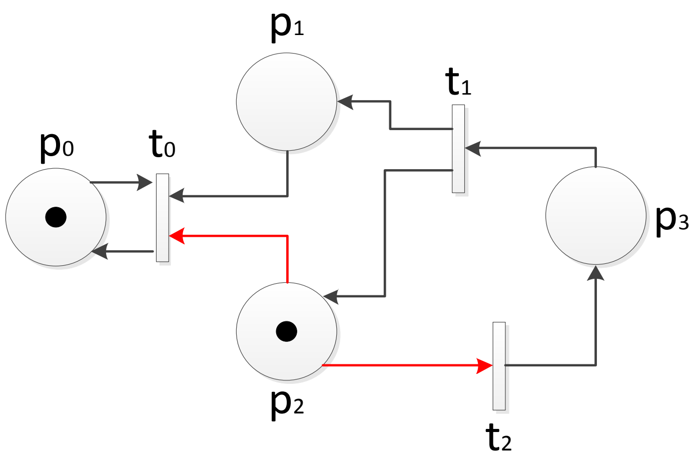


Coverability Tree:

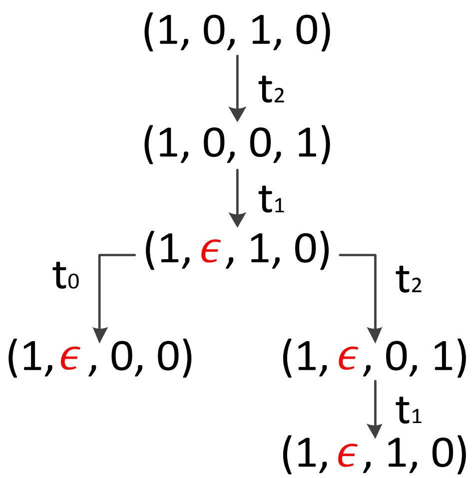


Conflicts

Conflicted Petri Net:



Coverability Tree:

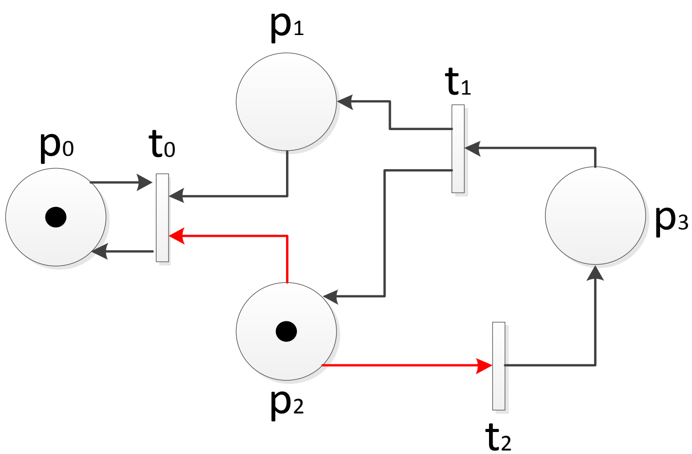


Informal Definition:

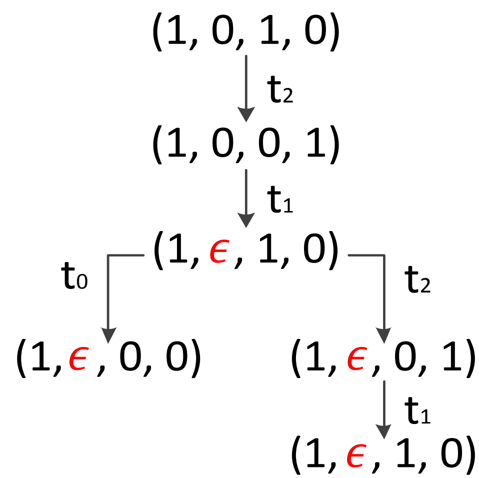
A Petri has a **conflicting configuration** if two or more transitions are enabled and firing one transition disables the other transitions.

Conflicts

Conflicted Petri Net:



Coverability Tree:



Informal Definition:

A Petri is **conflict free** if it has no conflicting configurations.

Reachability Problem

Is a certain marking reachable in a Petri Net?

- I.e. can a system reach some bad state?

Analysis:

1. What about the reachability problem for Petri Nets in general?
2. Are there certain properties which allow for deciding the reachability problem efficiently?

General analysis tool: The reachability/coverability tree.

Reachability Problem

Decide whether a marking M can be reached for a Petri Net N.

- **In general**, double exponential time.
- **1-safe**, PSPACE-complete.
- **Conflict-free**, NP-complete.
- **Bounded and conflict-free**, polynominal.

Conclusion

- Petri Nets allow for a **concise representation** of concurrent processes.
- In general, Petri Nets recognize **all regular languages**, **some context-free languages** and **some context-sensitive languages**.
- In general, certain **decision problems** are **harder** than in other automata models.
- The **expressiveness power** of Petri Nets can be **restricted** in order to **decide** certain **decision problems efficiently**.

References

[1] Murata, T. (1989, April). **Petri Nets: Properties, Analysis and applications.** *In Proceedings of the IEEE* (Vol. 77, pp. 541-580).

[2] Peterson, J. L. (1977, September). **Petri Nets.** *ACM Computing Surveys*, 9 (3), 223-252.

[3] Thomas, W. (2005) **Applied Automata Theory.** *Script available: http://drona.csa.iisc.ernet.in/~deepakd/atc-common/wolfgang-aat.pdf*

[4] Esparza, J. and Nielsen, M. (1995) **Decidability issues for Petri nets - a survey.** *In Bulletin of the EATCS, 1994. Revised, 1995.*

The End

Thank you for your attention!