Petri Nets

Albert-Ludwigs-Universität Freiburg

January 27, 2018

Introduction

- Concurrent computing: several computations are executed concurrently and not sequentially.
- Many dependencies can exist in the system.
- A model is required to investigate these systems.
- DFA and NFA can grow very big.

Example for problem of concurrent computing

An example for a problem of concurrent computing is a shared resource:

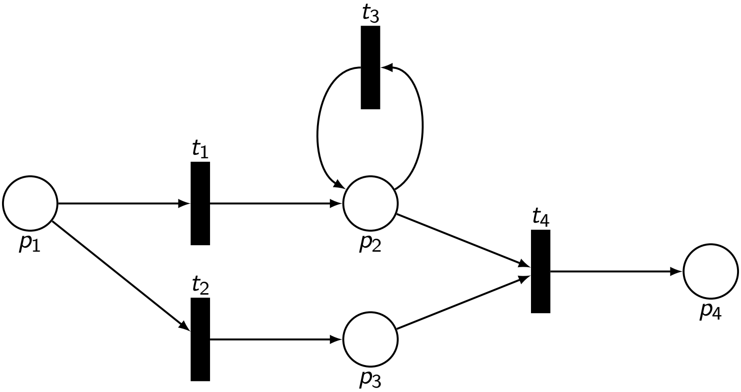
- One resource is shared by several processes.
- Only one process should access the resource at the same time.
- Solution: mutex lock

Definitions

Definition

A Petri Net N consists of a tuple $N = (P, T, F)$ where

- P is a finite, nonempty set of *places*
- T is a finite, nonempty set of *transitions*
- $F \subseteq (P \times T) \cup (T \times P)$ is a *flow relation*



$$P = \{p_1, p_2, p_3, p_4\} \quad T = \{t_1, t_2, t_3, t_4\}$$

Pre- and postset

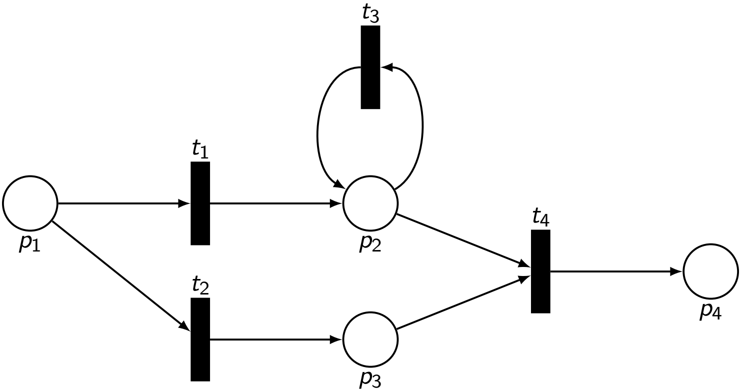
Given a Petri Net $N = (P, T, F)$, it is defined

for every transition $t \in T$

- the *preset* $^{\bullet}t := \{p \in P | (p, t) \in F\}$
- the *postset* $t^{\bullet} := \{p \in P | (t, p) \in F\}$

for every place $p \in P$

- the *preset* $^{\bullet}p := \{t \in T | (t, p) \in F\}$
- the *postset* $p^{\bullet} := \{t \in T | (p, t) \in F\}$



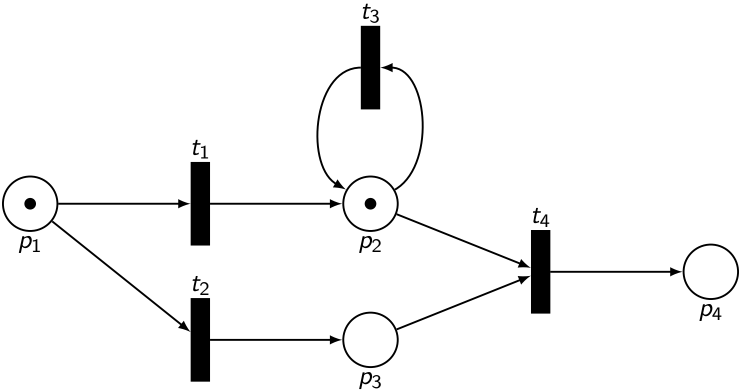
$$^\bullet t_4 = \{p_2, p_3\} \quad t_4^\bullet = \{p_4\} \quad ^\bullet p_2 = \{t_1, t_3\} \quad p_2^\bullet = \{t_3, t_4\}$$

Markings and tokens

Definition

A *marking* of a Petri net N is a function $m : P \rightarrow \mathbb{N}_0$ which assigns a number of *tokens* to every place. The set of all *markings* of a Petri net is M.

Given $p_1, ..., p_n$, we write the marking as a vector of dimension n which is written $\bar{m} = (m_1, ..., m_n)$. Each marking $m_i \in \bar{m}$ is the number of tokens that is assigned to place p_i.



$$m(p_1) = 1 \quad m(p_4) = 0 \quad \bar{m} = (1, 1, 0, 0)$$

Firing of a transition

Let m and m' be a markings of a Petri Net $N = (N, T, F)$ and $t \in T$. We define

$m \triangleright_t m'$ if and only if $\forall p \in {}^\bullet t : m(p) > 0$ and

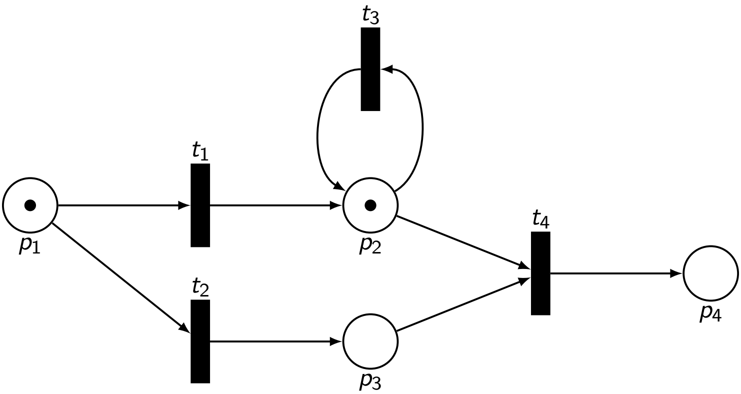
$$\forall p \in P : m'(p) = \begin{cases} m(p) - 1 \text{ if } (p \in {}^\bullet t \wedge p \notin t^\bullet) \\ m(p) + 1 \text{ if } (p \notin {}^\bullet t \wedge p \in t^\bullet) \\ m(p) \quad\quad \text{if } (p \notin {}^\bullet t \wedge p \notin t^\bullet) \vee (p \in {}^\bullet t \wedge p \in t^\bullet) \end{cases}$$

A transition t can be executed, if there is a token on every place in the preset of t. We call this firing of a transition or, t is fired.

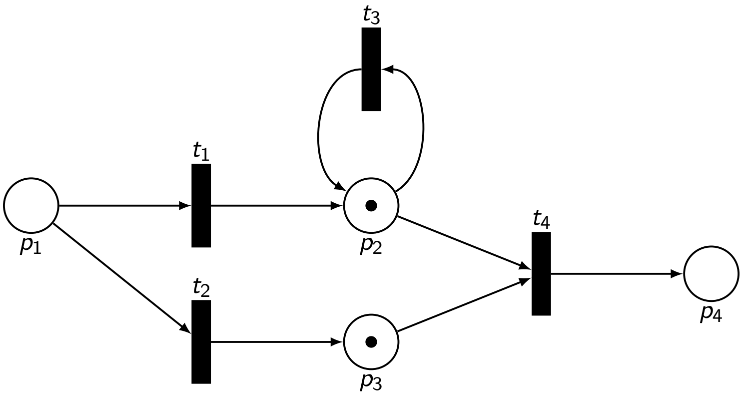
When a transition is fired, one token is removed from every place in the preset of t and one token is added to every place in the postset of t.

If s transitions are fired sequentially, we call this a *sequence of transitions of length* s.

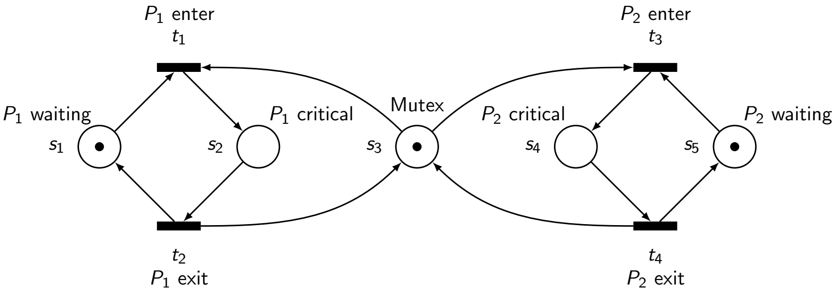
Example of a Petri net



Example of a Petri net



Mutex



Graphic by Dominik Drexler

Extension of definitions for language acceptance

Let $N = (P, T, F)$ be a Petri net and M^-, M^+ be finite sets of *initial and final markings* with $M^- \subseteq M$ and $M^+ \subseteq M$.

Let $\sum$ be a finite set of symbols (*alphabet*) and $\ell : T \rightarrow \sum$ be a labeling function which assigns a symbol from the alphabet to every transition.

Extension of definitions for language acceptance

This modified Petri Net accepts a word $w \in \sum^*$ if there exists a firing sequence τ from $m^- \in M^-$ to $m^+ \in M^+$ with $\ell(\tau) = w$ with $\ell(\tau) := \ell(t_1)...\ell(t_n)$.

The set of all words accepted by N is denoted as $L(N)$.
$L(N)$ is called the *language of* N.

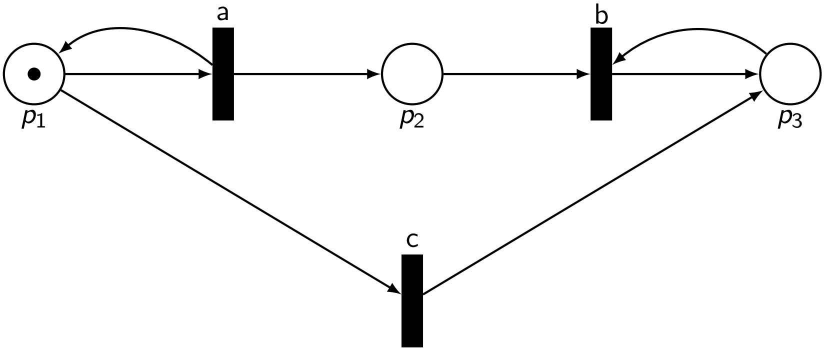
Petri nets and regular languages

For every regular language L, there exists a Petri net N such that $L = L(N)$.

$\rightarrow$ Petri nets are at least as expressive as DFA/NFA.

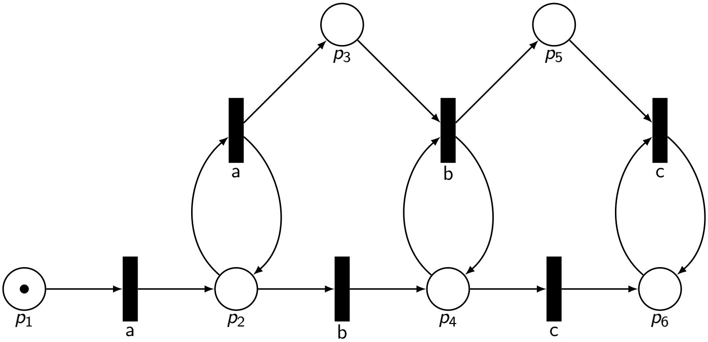
Example for acceptance of contextfree language

$L = \{a^n c b^n | n \geq 0\}$ $m^- = (1, 0, 0)$ $m^+ = (0, 0, 1)$



Example for acceptance of contextsensitive language

$$L = \{a^n b^n c^n | n > 0\} \quad m^- = (1,0,0,0,0,0) \quad m^+ = (0,0,0,0,0,1)$$



Theorem

Theorem

The context-free mirror-language $L = \{ww^R | w \in \{a, b\}^*\}$ *is not a Petri net language.*

Examples

- $\in L$: *abba*, *aaaaaa*
- $\notin L$: *aababaa*, *aabb*

Lemma

Lemma

For every Petri net recognizing L, the following applies:
for a sufficiently large s, only less than 2^s markings are reachable.

Sketch of proof:

- Assume that the Petri net has got r transitions.
- We consider a sequence of s transitions.
- Let transition t_i be fired k_i times.
- We can reach as many markings as there are tuples $(k_1, ..., k_r)$.
- Therefore, there are $(s + 1)^r$ such tuples.

The number of tuples reachable is polynomial in growth while 2^s is exponential in growth.

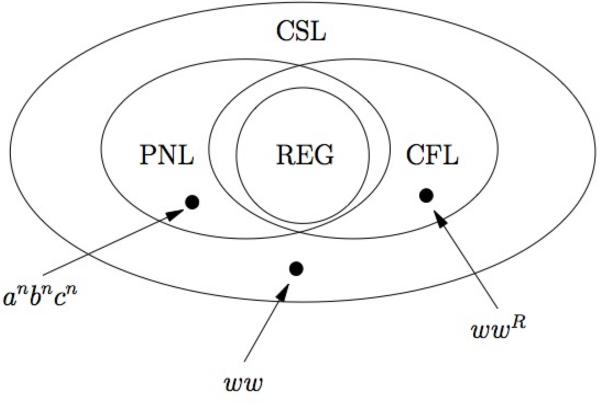
Proof of Theorem

Theorem

The context-free mirror-language $L = \{ww^R | w \in \{a, b\}^*\}$ *is not a Petri net language.*

- We assume that there is a Petri net N with r transitions accepting L.
- After reading s letters, there must be as many different markings as words of length s which is 2^s.
- There are words w and w' which lead N to the same marking which means that N can't distinguish between ww^R and $w'w^R$.

$\rightarrow$ Contradiction, L is not a Petri net language.



Applied Automata Theory Script RWTH Aachen, Page 171

Source

- Applied Automata Theory Script RWTH Aachen.
  Prof. Dr. Wolfgang Thomas. Accessed, November 11, 2017