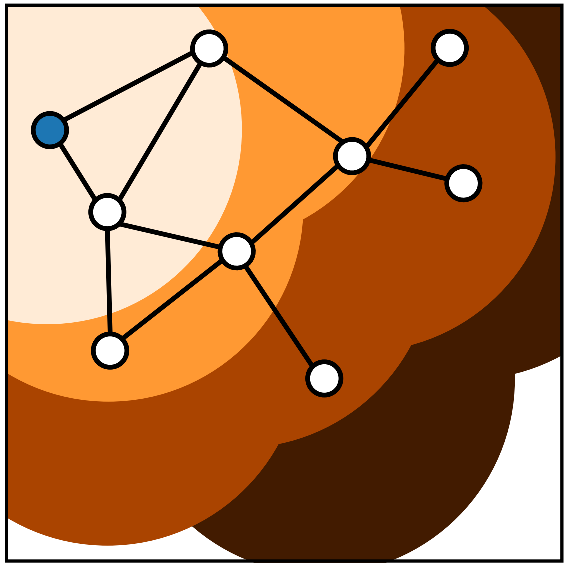
Discrete Event Systems
Time Petri Nets



Romain Jacob

www.romainjacob.net

ETH Zurich (D-ITET)

December 16, 2021

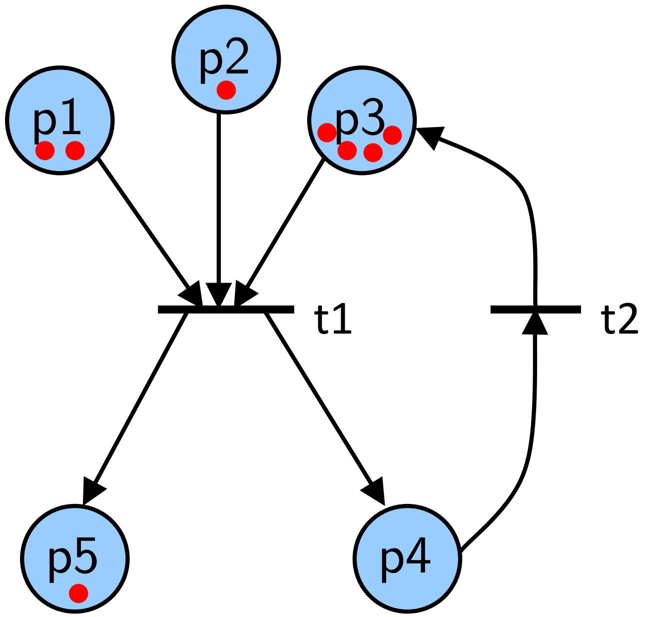
Most materials from Lothar Thiele

Last week in
Discrete Event Systems

Petri Net – Definition

A Petri net is a bipartite, directed graph defined by a 4-tuple (S, T, F, M_0), where

- S is a set of places p
- T is a set of transitions t
- F is a set of edges (flow relations) f
- M0 : S $\rightarrow$ N; the initial marking



$\{p1, p2, p3, p4, p5\} \in$ S
$\{t1, t2\} \in$ T
$\{(p1, t1), (p2, t1), (t1, p5), \ldots\} \in$ F

Definition

- Semantics
- Token game

Properties

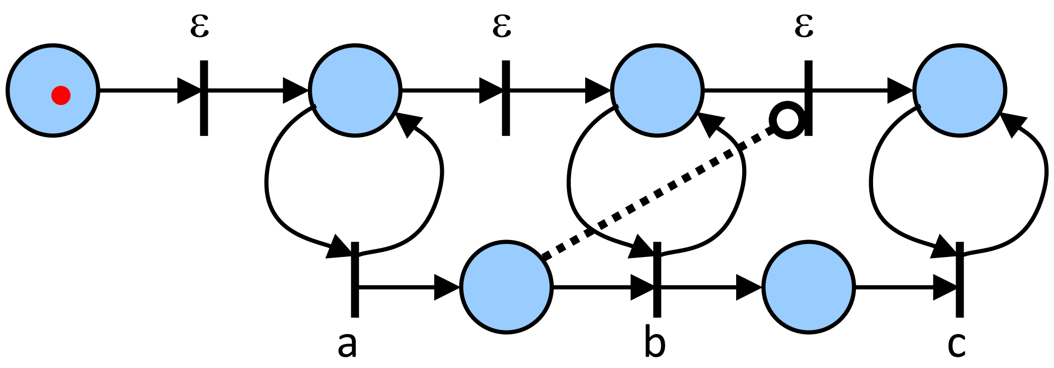
- Safety
- Liveness

Analysis

- Coverability tree
- Incidence matrix

Common Extensions

Colored Petri nets	Tokens carry values (colors). A Petri net with finite number of colors can be transformed into a regular Petri net.
Continuous Petri nets	The number of tokens can be a real number (not only an integer). Cannot be transformed into a regular Petri net.
Inhibitor Arcs	Enable a transition if a place contains no tokens. Cannot be transformed to a regular Petri net



$L(M_0) = \{a^n\, b^n\, c^n \mid n \geq 0\}$

Common Extensions

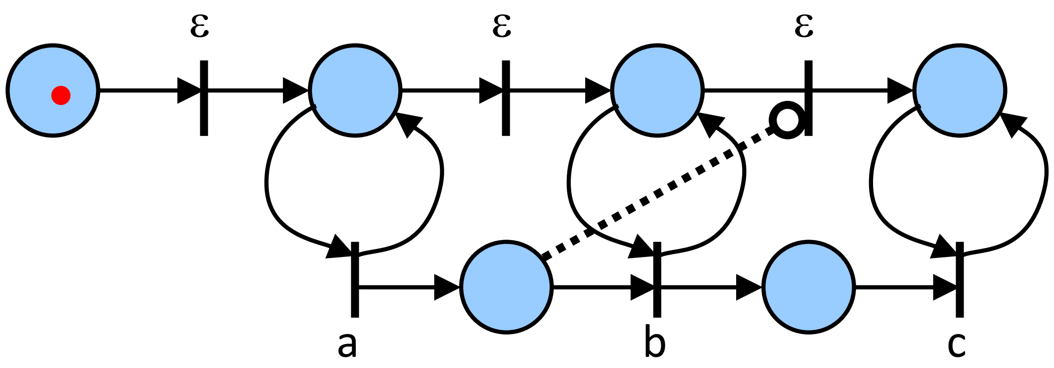
Inhibitor arcs makes PN Turing Complete

Zaitsev, Dmitry A. "Toward the minimal universal Petri net." IEEE Transactions on Systems, Man, and Cybernetics: Systems 44.1 (2013): 47-58.

Inhibitor Arcs

Enable a transition if a place contains no tokens.

Cannot be transformed to a regular Petri net



$L(M_0) = \{a^n\, b^n\, c^n \mid n \geq 0\}$

Behavioral Properties (2)

CTL equivalent formula

f means "fires"

Liveness

A transition t in a Petri net is

dead	iff t cannot be fired in any firing sequence,	$AG\ \neg f$
L_1-live	iff t can be fired at least once in some firing sequence,	$EF\ f$
L_2-live	iff, $\forall$ k $\in \mathbf{N}^+$, t can be fired at least k times in some firing sequence,	$\emptyset$
L_3-live	iff t appears infinitely often in some infinite firing sequence,	$EG\ EF\ f$
L_4-live	(live) iff t is L_1-live for every marking that is reachable from M_0 .	$AG\ EF\ f$

L_{j+1}-liveness implies L_j-liveness.

A Petri net is free of deadlocks iff there is no reachable marking from M_0 in which all transitions are dead.

Any further question?

This week in
Discrete Event Systems

Discrete Event Models with Time

In many discrete event systems, time is an important factor.

- queuing systems
- computer systems
- digital circuits
- workflow management
- business processes

Based on a timed discrete event model, we would like to determine properties:

- delay
- throughput
- execution rate
- resource load
- buffer sizes

▶ There are many ways of adding the concept of time to Petri nets and finite automata.
In the following, we present one specific model.

Discrete Event Models with Time

What can you do with a timed model?

Verify timed properties

- How long does it take until a certain event happens?
- What is the minimum time between two events?

Simulate the model

- Given a specific input, how does the system state evolve over time?
- Is the resulting trace of execution what we had in mind?

Definition

Simulation

Definition

Simulation

Time Petri Net

We define a delay function $d: T \rightarrow R$
that determines the delay between the activation of a transition t and its firing.

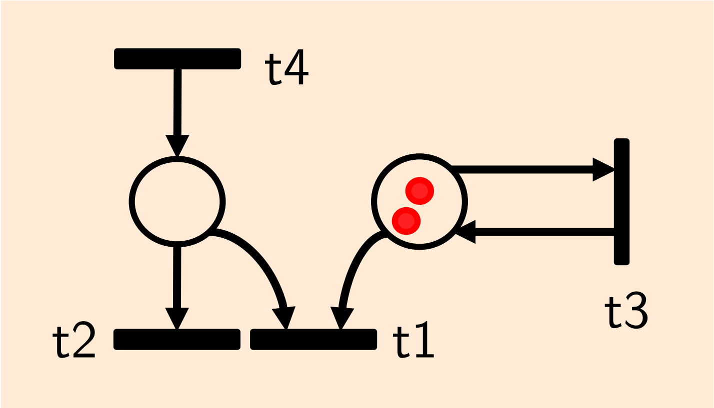
- Repeated calls may lead to the same value — constant delay
  or to different ones every time. — values of some random variable

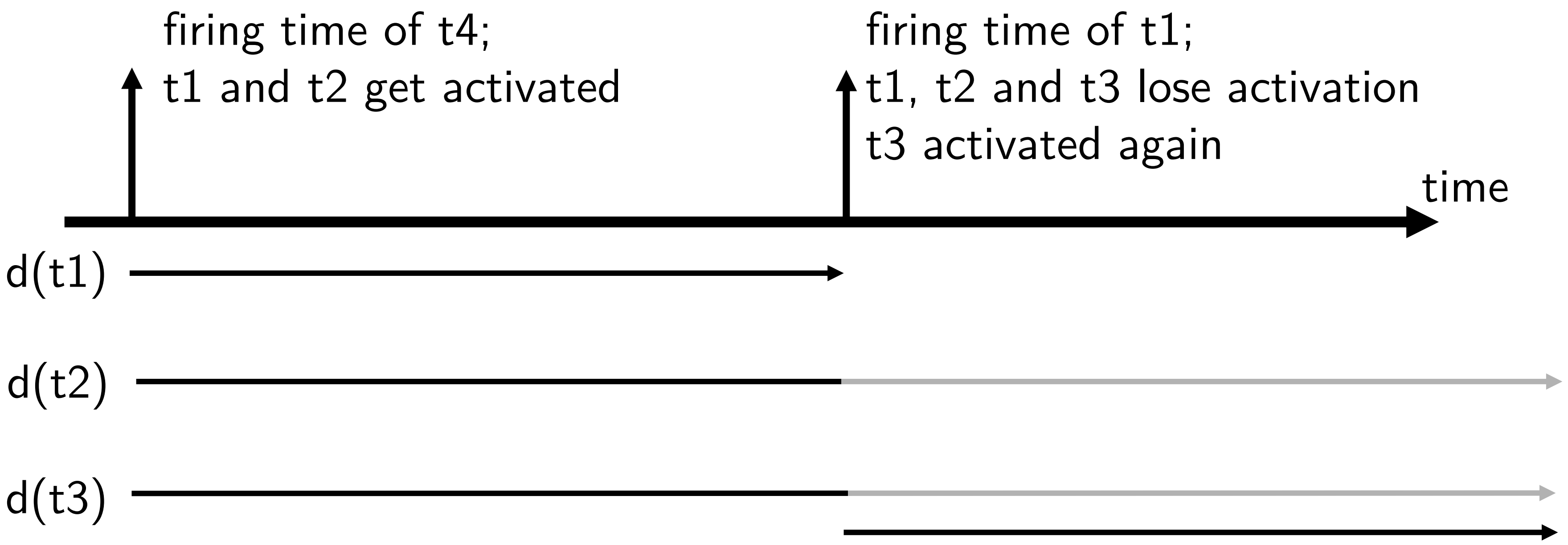
- The function is called for every new activation of transition t
  and determines the time until the transition fires.

- There is a new activation whenever a token is removed from some input place of t.
  - If the transition t loses its activation, then d(t) is called again at the next activation.

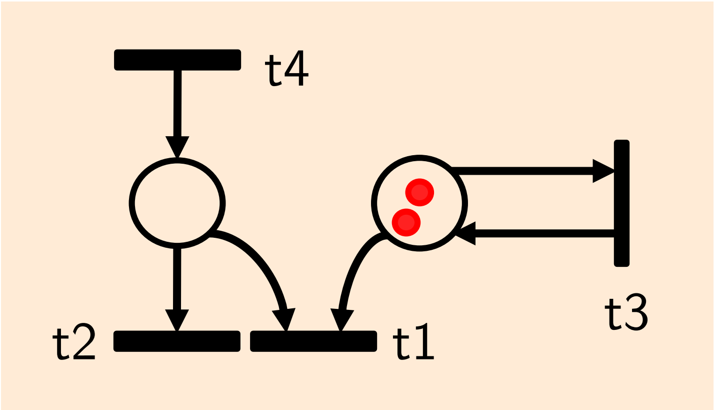
- Only one transition fires at a time (same as with regular Petri nets).
  - If two transitions have the same firing time,
    one of them is chosen non-deterministically to fire first.

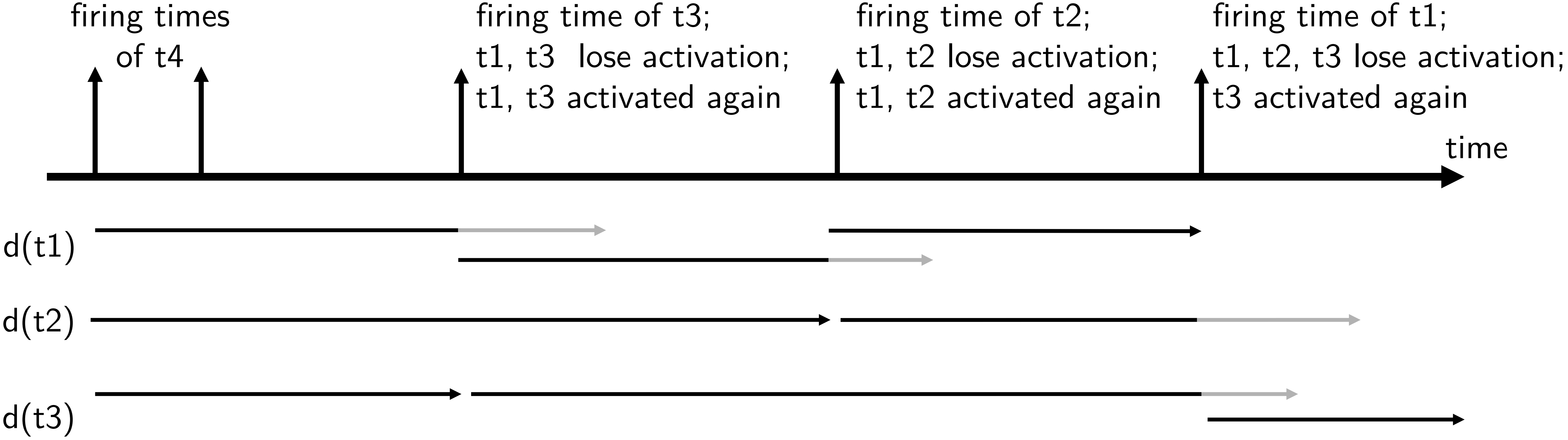
Time Petri Net





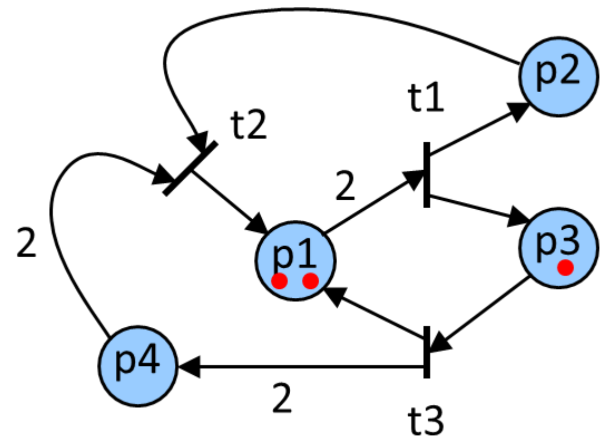
Time Petri Net





Time Petri Net

- The time when a transition t fires is called the firing time.
- A time Petri net can be regarded as a generator for firing times of its transitions.



$d(t1) = 1 \xrightarrow{t1} \{1, 6, 9, 12, \ldots\}$

$d(t2) = 2 \xrightarrow{t2} \{5, 8, 11, 13, \ldots\}$

$d(t3) = 3 \xrightarrow{t3} \{3, 6, 9, 12, \ldots\}$

initialization time 0

firing time sequences for transitions t1, t2 and t3

- How do we get the firing times? By simulation!

Time Petri Net

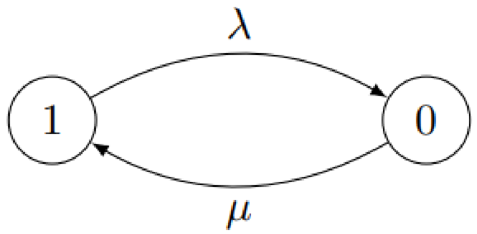
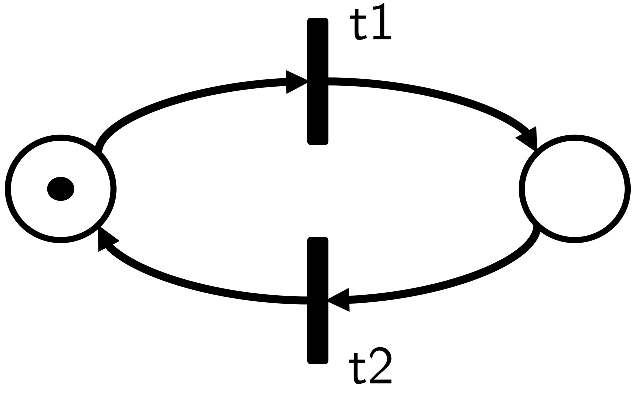


Figure 4.5: A CTMC modeling an unreliable system. In state 1 the system is working, in state 0 the system is faulty. The *failure rate*, i.e., the time until the system fails, is exponentially distributed with parameter λ. After a failure, the repair takes some time, exponentially distributed with parameter μ.

Example

Continuous Time Markov Chain

Equivalent time Petri net



d(t1) returns a sample of an exponentially distributed random variable with parameter λ

d(t2) returns a sample of an exponentially distributed random variable with parameter μ

Definition

Simulation

Simulation Principle

This simulation principle holds in one form or another for any simulator of timed discrete event models.

The simulation is based on the following basic principles.

$L = \{ (\mathrm{t_i}, \tau_i) \}$

1. The simulator maintains a set L of currently activated transitions and their firing times. We call L the event list from now on.

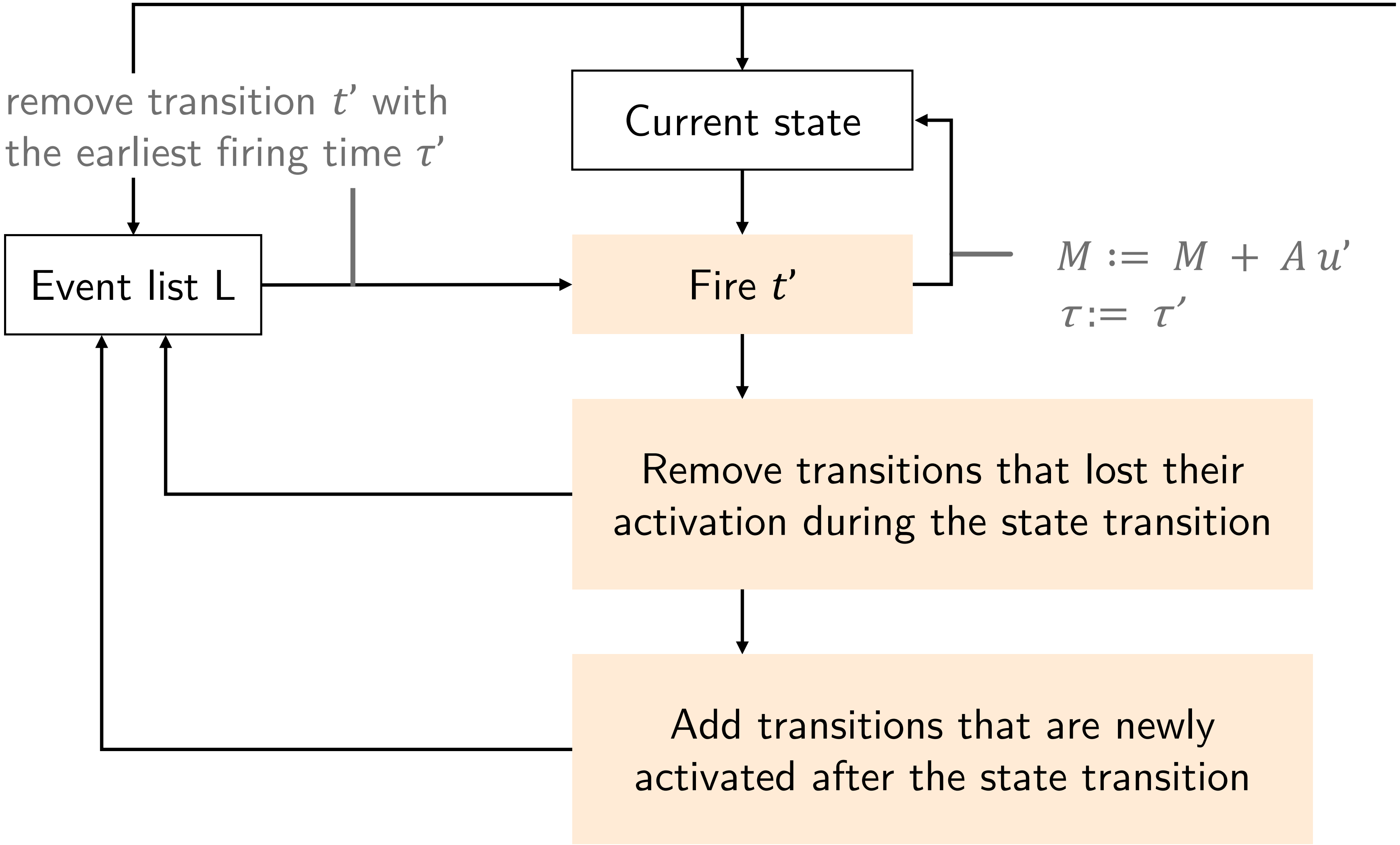
2. A transition with the earliest firing time is selected and fired. The state of the Petri net as well as the current simulation time is updated accordingly.

3. All transitions that lost their activation during the state transition are removed from the event list L.

4. Afterwards, all transitions that are newly activated are added to the event list L together with their firing times.

5. Then we continue with 2. unless the event list L is empty.

Simulation Principle



Initialization

- Event list L
- State M
- Simutation time τ

Update

- state
- simulation time

Iterate until L is empty

Time Petri Net – Simulation Steps

Initialization:

- Set the initial simulation time $\tau := 0$
- Set the current state to $M := M_0$
- For each activated transition t, add the event $(t, \tau + d(t))$ to the event list L

Determine and remove current event:

- Determine a firing event (t', τ') with the earliest firing time:

$$\forall 1 \leq i \leq N \ : \ \tau' \leq \tau_i \quad \text{where} \quad L = \{(t_1, \tau_1), (t_2, \tau_2), \cdots, (t_N, \tau_N)\}$$

- Remove event (t', τ') from the event list L:

$$L := L \setminus \{(t', \tau')\}$$

Time Petri Net – Simulation Steps

Update current simulation time:

- Set current simulation time $\tau := \tau'$

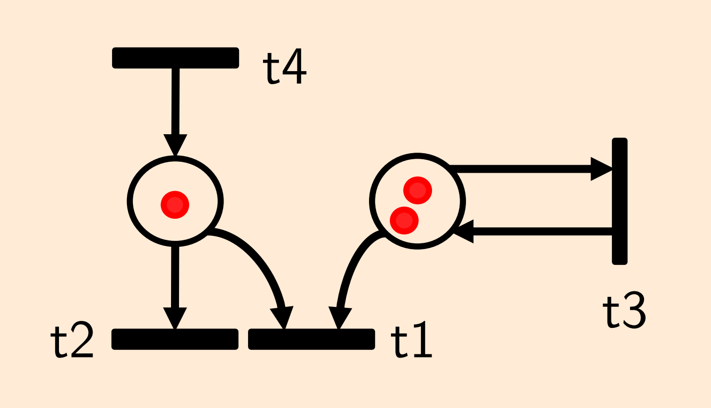
Update token distribution M

- Suppose that the firing transition has index j, i.e. $t_j = t'$.
  Then, the firing vector is

$$\begin{array}{c} u' = [\ 0 \quad \cdots \quad 0 \quad \underset{j}{1} \quad 0 \quad \cdots \quad 0\ ]^t \end{array}$$

- Update current state $M := M + A\,u'$

Time Petri Net – Simulation Steps



Remove transitions from L that lost activation:

- Determine the set of places S' from which at least one token was removed during the state transition caused by t':

$$S' = \{p \,|\, (p, t') \in F\}$$

- Remove from event list L all transitions in T' that lost their activation due to this token removal:

$$T' = \{t \,|\, (p, t) \in F \wedge p \in S'\}$$

Add all transitions to event list L that are activated but not in L yet:

- If some transition t with $M(p) \geq W(p, t)$ for all $(p, t) \in F$ is not in L, then add $(t, \tau + d(t))$ to the event list:

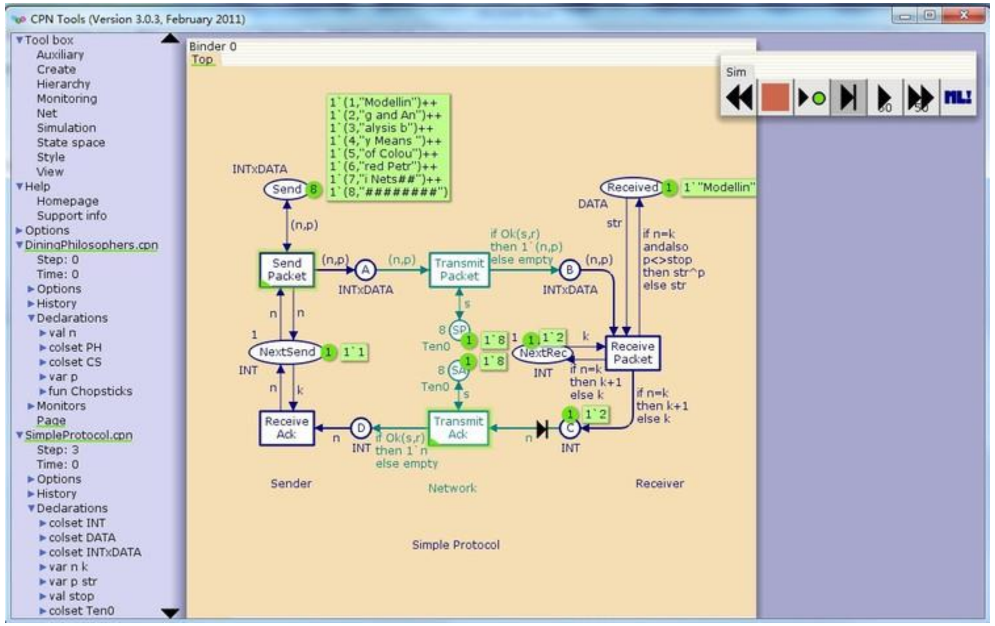
$$L := L \cup \{(t, \tau + d(t))\}$$

Petri Net Simulators

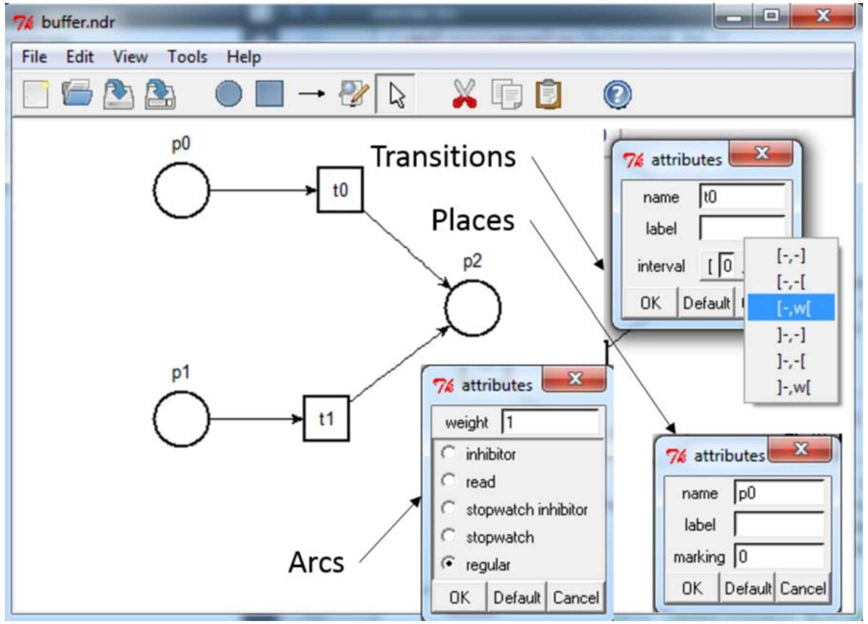
There are many simulators available

An overview
www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html

Examples



CPN Tools



TINA

Discrete Event Models with Time

In many discrete event systems, time is an important factor.

- queuing systems
- computer systems
- digital circuits
- workflow management
- business processes

Based on a timed discrete event model, we would like to determine properties:

- delay
- throughput
- execution rate
- resource load
- buffer sizes

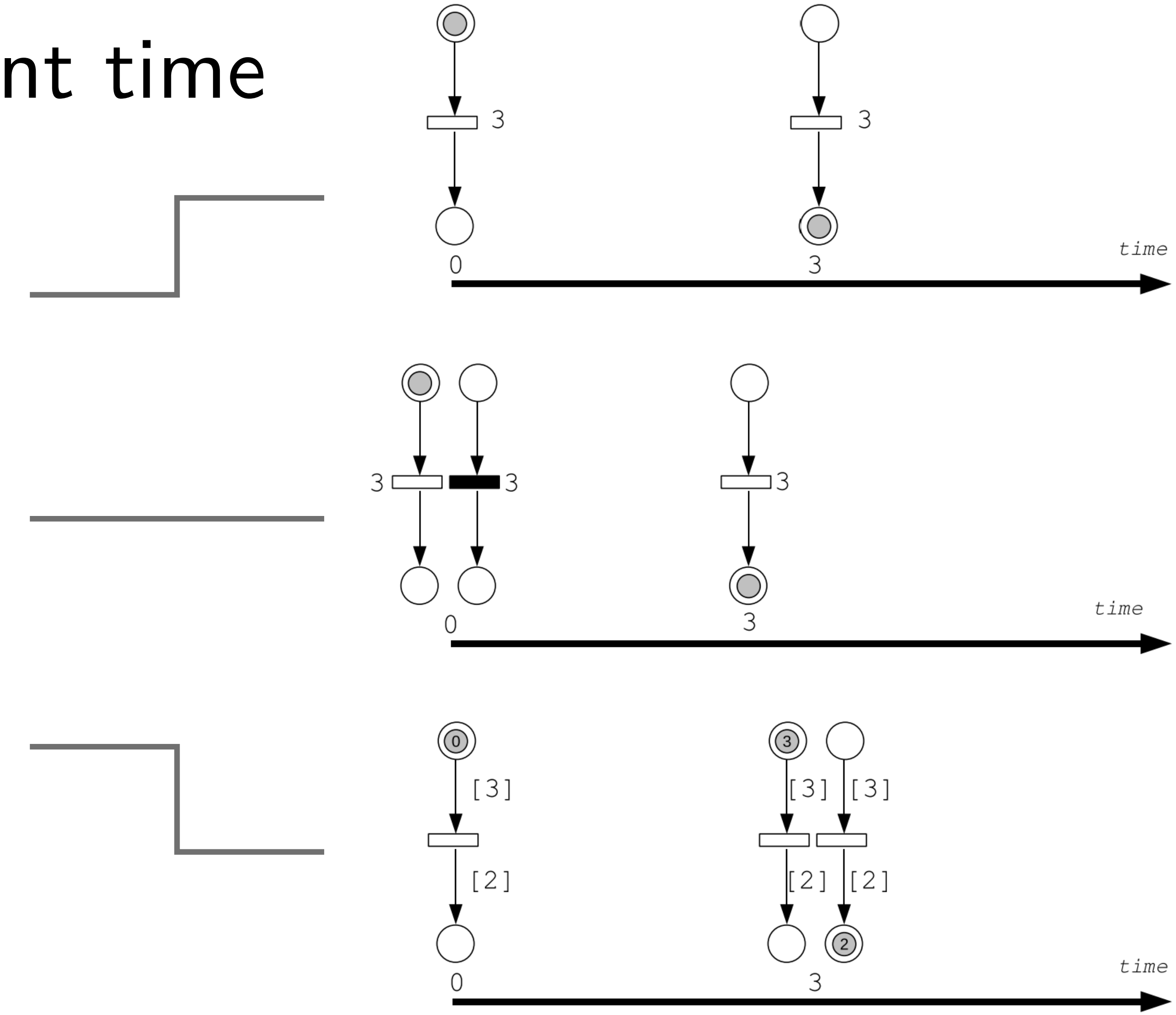
▶ There are many ways of adding the concept of time to Petri nets and finite automata.
In the following, we present one specific model. — What are the others?

There are mainly three ways to count time

Delay on the transition firing

Duration of the transition

Age of the tokens



www.lsv.fr/~haddad/disc11-part1.pdf

There are mainly three ways to count time

Delay on the transition firing	▶	Time Petri nets	Covered here
Duration of the transition			
Age of the tokens	▶	Timed Petri nets	

Expressivity and analysis feasibility vary between the models

Decidable means that

the problem can be solved on all inputs in a finite number of steps.

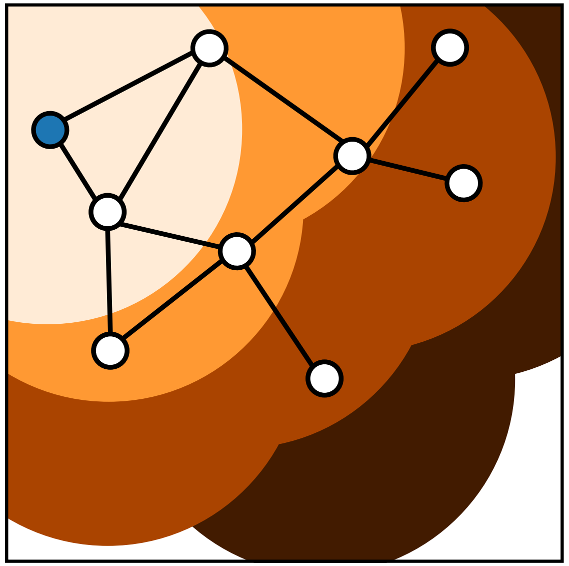
Time and Timed Petri nets are not equivalent.
There exists behavior that only each can model.

	Coverability	Reachability	Non-termination	Deadlock
Time Petri nets	Undecidable	Undecidable	Undecidable	Undecidable
Bounded Time Petri nets	Decidable	Decidable	Decidable	Decidable
Timed Petri nets	Decidable	Undecidable	?	?

Your turn to practice!
after the break

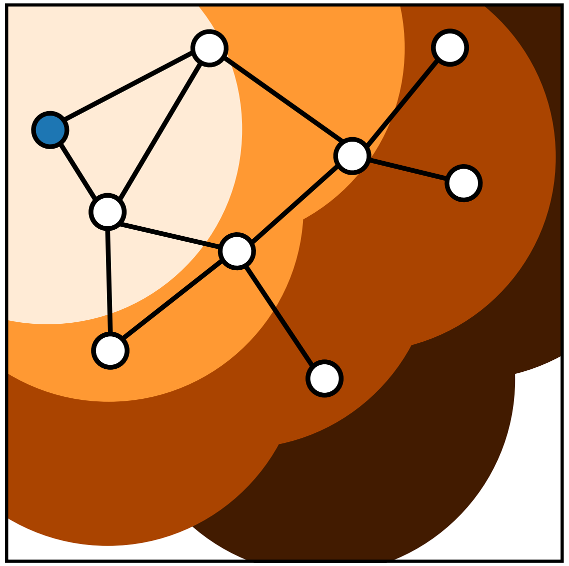
1. Model arithmetic operations with Petri nets
2. Use a simulator to explore the timed behavior of a simple Petri net model
3. Use a model-checker to adapt a system design

Quick recap
Discrete Event Systems



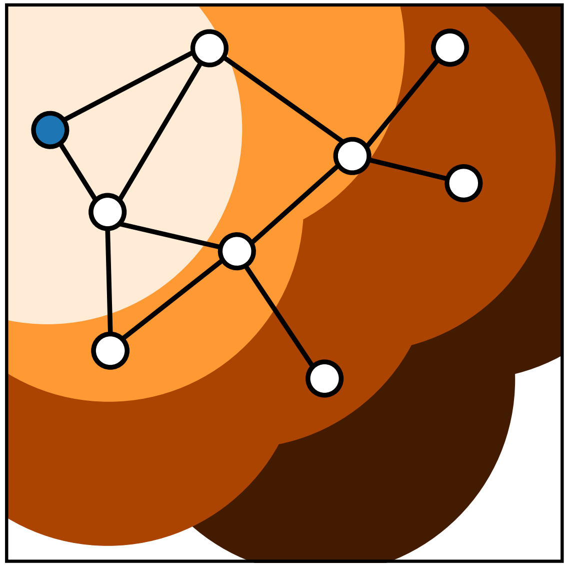
- How to efficiently explore the state space of DES models?
- How to formulate temporal properies of interest?
- How to formally verify such properties?
- How to efficiently model concurrency in DES?

Quick recap
Discrete Event Systems



How to efficiently explore the state space of DES models?	Set of states & BDDs
How to formulate temporal properies of interest?	CTL fomulas
How to formally verify such properties?	Reachability & model-checking
How to efficiently model concurrency in DES?	Petri nets w/ and w/o time

Thank you for following
Discrete Event Systems



Romain Jacob

www.romainjacob.net

ETH Zurich (D-ITET)

December 16, 2021

Most materials from Lothar Thiele