Information Systems Analysis

Jan Magott

Choose yourself and new technologies

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Information Systems Analysis

Jan Magott

Introduction

Choose yourself and new techn

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!

Wrocław University of Technology

EUROPEAN
SOCIAL FUND

Information systems analysis

1. Introduction
2. Petri nets and their applications (JM)
3. Performance evaluation of systems (JM)
4. Temporal logic and its applications (PG)

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

PROGRAMME CONTENT

	Form of classes - lecture	Number of hours
Lec 1	Introduction into modelling of concurrent systems using Petri nets	1
Lec 2	Behavioral properties of Petri nets: boundness, safety, reachability, liveness, reversibility, existence of home marking, persistency	4
Lec 3	Synchronization distance, bounded fairness relation	1
Lec 4	Time Petri nets	1,5
Lec 5	Coverability tree	1
Lec 6	Matrices and net reductions in analysis of Petri nets properties	1,5
Lec 7	Introduction into performance evaluation of information systems	1
Lec 8	Performance evaluation of sequential programs	1
Lec 9	Performance evaluation using queueing models	2
Lec 10	Fundamental laws of operational analysis	4
Lec 11	Stochastic and generalized stochastic Petri nets	2

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Lec 12	Logic LTL	2
Lec 13	Logic CTL	1
Lec 14	Model verification of system	1
Lec 15	Model verification of system using UPPAL time state automata	2
Lec 16	Model verification of system using NuSMV state automata	3
Lec 17	Another kinds of temporal logics and temporal data bases	1

Total hours30

F11	PEK_W01 ÷ PEK_W02 PEK_U01 ÷ PEK_U02	Observation of: the preparation to the laboratory exercises, execution of the exercises, the results achieved, verbal responses.
F21	PEK_W03 ÷ PEK_W06 PEK_U03 ÷ PEK_U08	Observation of: the preparation to the laboratory exercises, execution of the exercises, the results achieved, verbal responses.
F31	PEK_W07 ÷ PEK_W9 PEK_U09 ÷ PEK_U11	Observation of: the preparation to the laboratory exercises, execution of the exercises, the results achieved, verbal responses.
F12		Exam
F22		Exam
F32		Exam

$F1=F11$ if $4{,}5 \leqslant F11$
$F1=F12$ if $3 \leqslant F11 < 4{,}5$
$F1=2$ if $F11=2$

$F2=F21$ if $4{,}5 \leqslant F21$
$F2=F22$ if $3 \leqslant F21 < 4{,}5$
$F2=2$ if $F21=2$

$F3=F31$ if $4{,}5 \leqslant F31$
$F3=F32$ if $3 \leqslant F31 < 4{,}5$
$F3=2$ if $F31=2$

$P=F1/3+F2/3+F3/3$ if ($3 \leqslant F1$ i $3 \leqslant F2$ i $3 \leqslant F3$), otherwise $P=2$

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wroclaw University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Information systems analysis

Literature for Petri nets and performance evaluation

- T. Murata, Petri nets: Properties, analysis and applications, Proceedings of the IEEE, 1989, Vol. 77, No. 4, 541-580.
- W. Reisig, Petri Nets – An Introduction, Springer, 1985.
- W. Reisig, Sieci Petriego, WNT, 1988.
- M. Szpyrka, Sieci Petriego w modelowaniu i analizie systemów współbieżnych, Inżynieria oprogramowania, WNT, 2008.
- E. D. Lazowska, J. Zahorjan, G. S. Graham, K. C. Sevcik, Quantitative System Performance, Computer System Analysis Using Queueing Network Models, Prentice-Hall, Englewood Cliffs, 1984.
- T. Czachórski, Modele kolejkowe w ocenie efektywności sieci i systemów komputerowych, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 1999.

Introduction

Life cycle phases of information systems where modelling and analysis are applied:

- Specification,
- Design,
- Verification,
- Testing,
- Implementation,
- Performance evaluation and engineering
- Reliability analysis and engineering
- Safety analysis and engineering

Introduction

Information systems models can be divided into:

- Analytic,
- Simulation.

Information systems models can be divided into:

- Deterministic,
- Non-deterministic,
- Probabilistic.

Introduction

Expressive power, decision power of a modelling technique

Expressive power is characterized by:

Classes of systems that can be expressed by a particular modelling technique,

Properties of systems that can be described.

Decision power is characterized by:

Classes of systems that solutions can be found by modelling technique for,

Properties of systems that solutions can be found for.

(Computational complexity limitations)

Information Systems Analysis

Jan Magott

Concurrent systems modelling using Petri nets

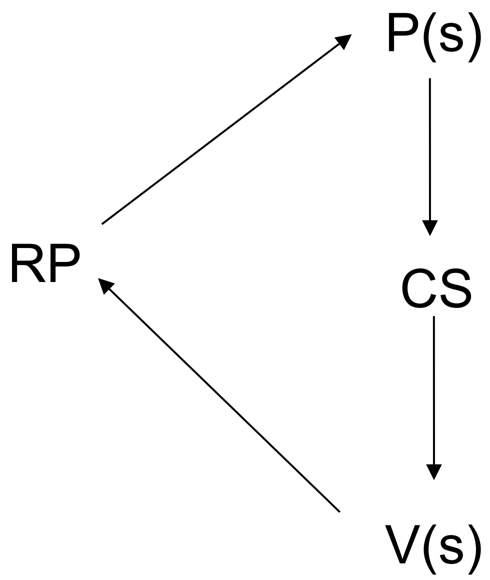
Choose yourself and new tech

Concurrent systems modelling using Petri nets

- Mutual exclusion of two cyclic sequential processes

Process activity stages:

CS – a critical section,
RP – remainder of the process,
P(*s*), *V*(*s*) – semaphore operations.

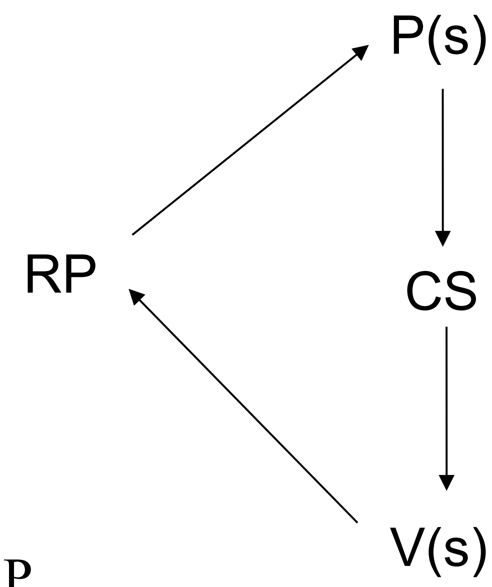


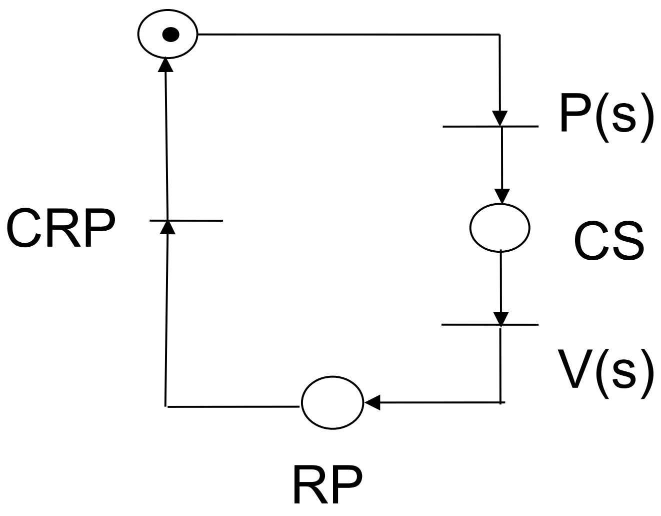
Concurrent systems modelling using Petri nets

Mutual exclusion of two cyclic sequential processes

Process activity stages:

CS – a critical section,
RP – remainder of the process,
P(*s*), *V*(*s*) – semaphore operations.
CRP – completion of execution of CRP

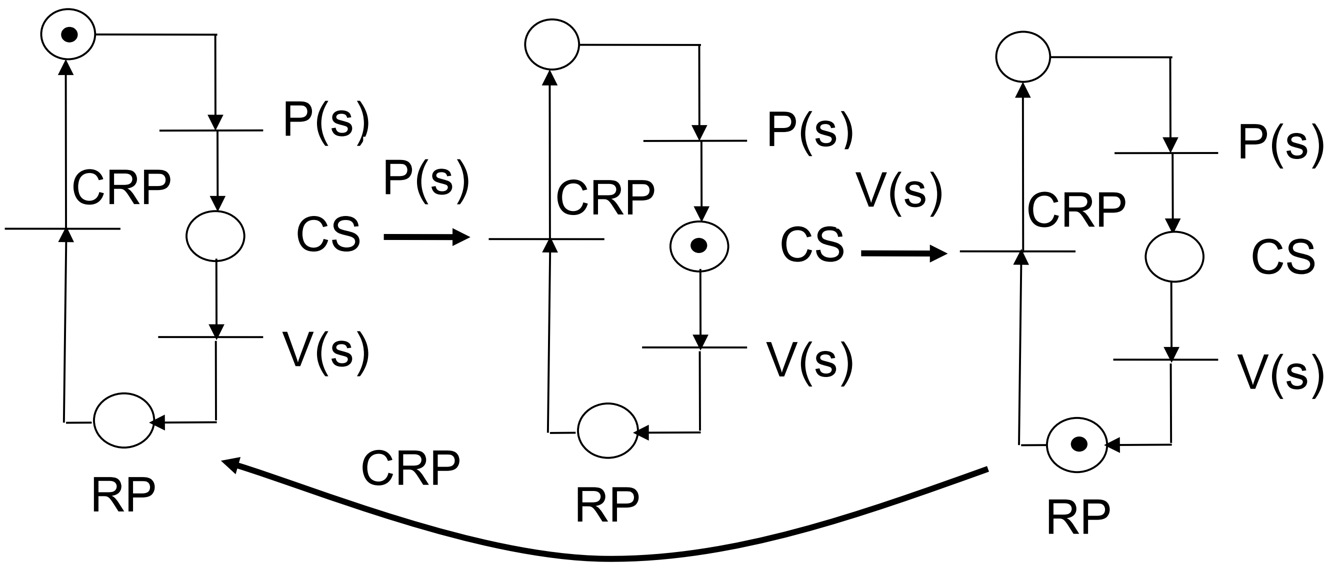




A Petri net (safe and live) model of the cyclic sequential process

Concurrent systems modelling using Petri nets

- Mutual exclusion of two cyclic sequential processes (cont.)

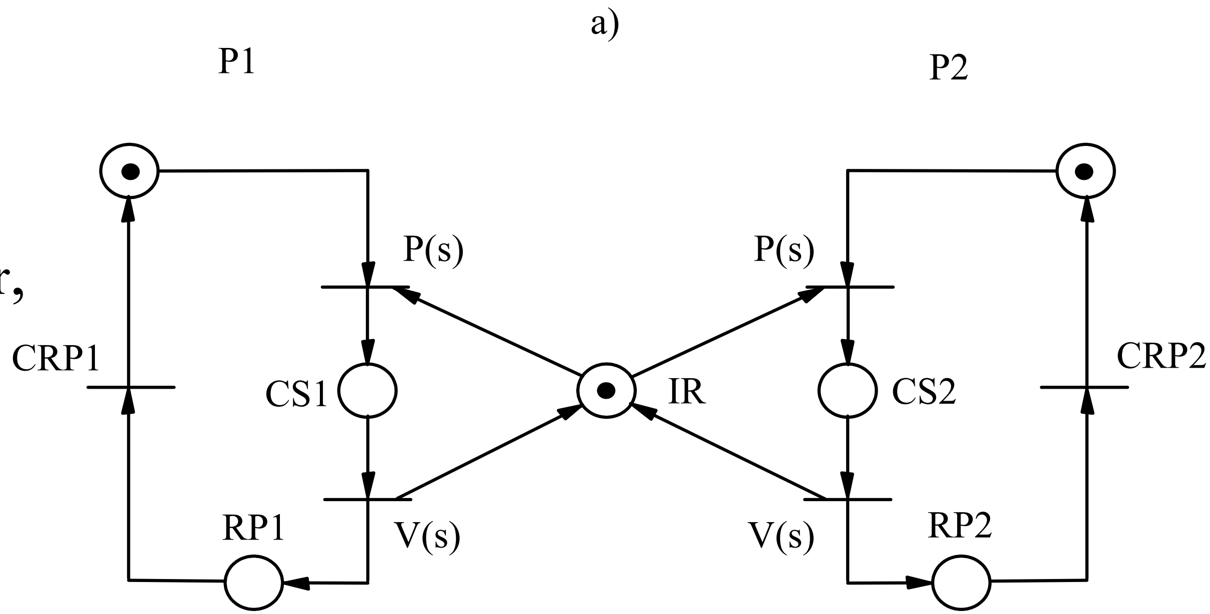


Transitions fired : P(s), V(s), RP

Concurrent systems modelling using Petri nets

- Mutual exclusion of two cyclic sequential processes (cont.)

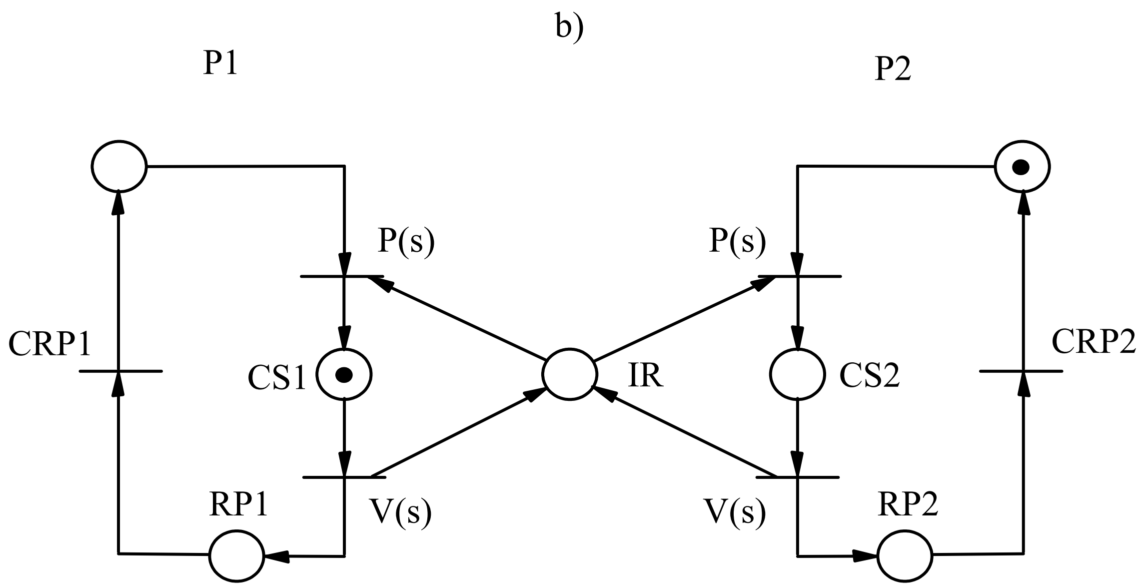
CS – the process is in critical section,
RP – the process is in its remainder,
CRP – completion of the process remainder,
P(*s*), *V*(*s*) – semaphore operations,
IR – critical resource is idle.



Transitions with labels *P*(*s*) are enabled.

Concurrent systems modelling using Petri nets

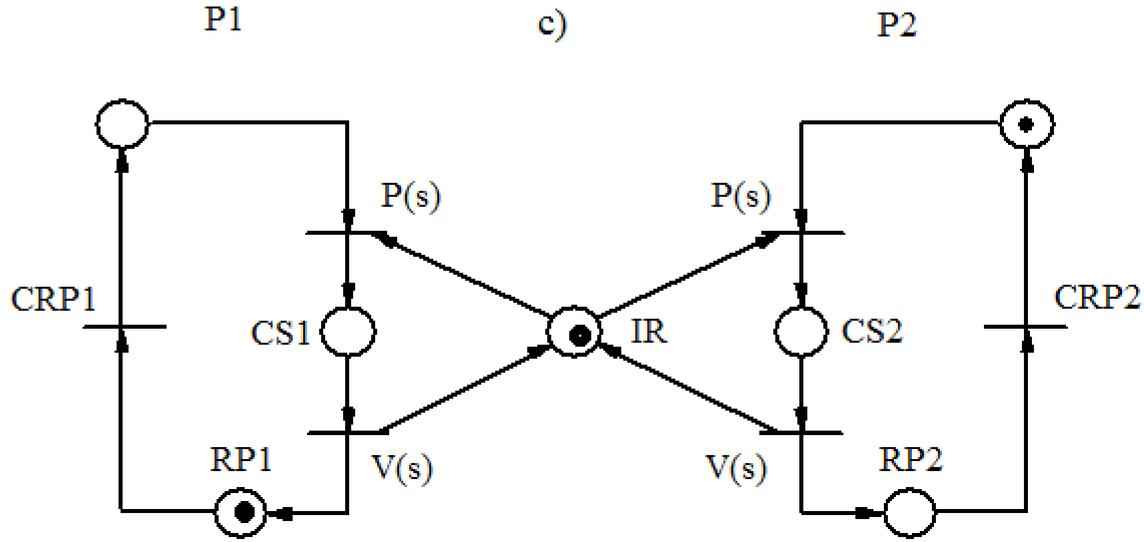
- Mutual exclusion of two cyclic sequential processes (cont.)



Transition $V(s)$ of the P1 process model is enabled
Petri net (safe and live) after firing of transition $P(s)$ of P1

Concurrent systems modelling using Petri nets

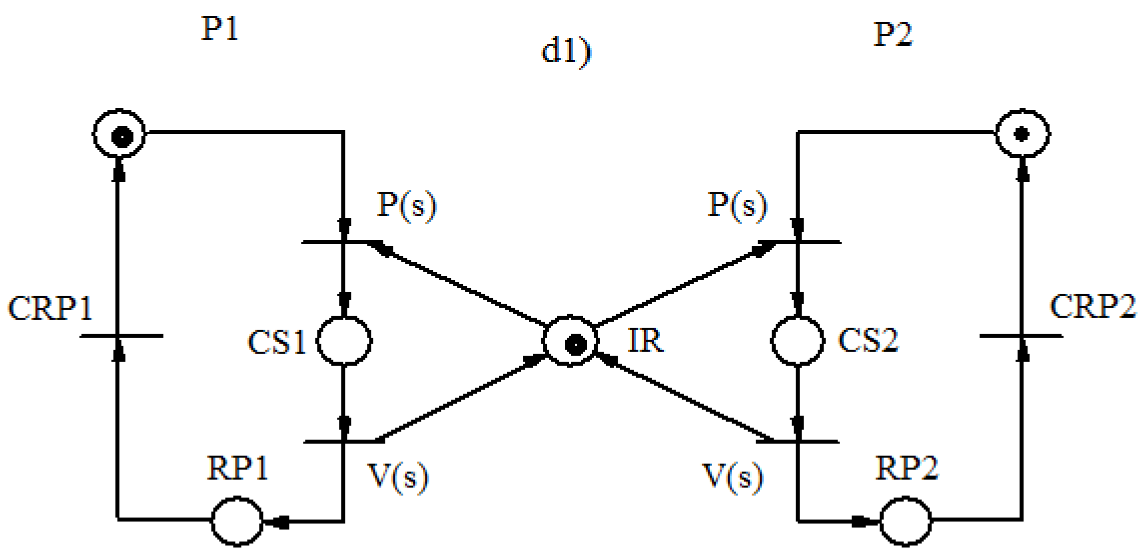
- Mutual exclusion of two cyclic sequential processes (cont.)



Transition $P(s)$ of the P2 process model is enabled
Petri net (safe and live) after firing of transition $V(s)$ of P1

Concurrent systems modelling using Petri nets

Mutual exclusion of two cyclic sequential processes (cont.)



Transitions *P*(*s*) of both process models are enabled
Petri net (safe and live) after firing of transition *CRP1*

Process 1

Process 2

R1 requested

R2 requested

Deadlock example:

R2 requested

R1 requested

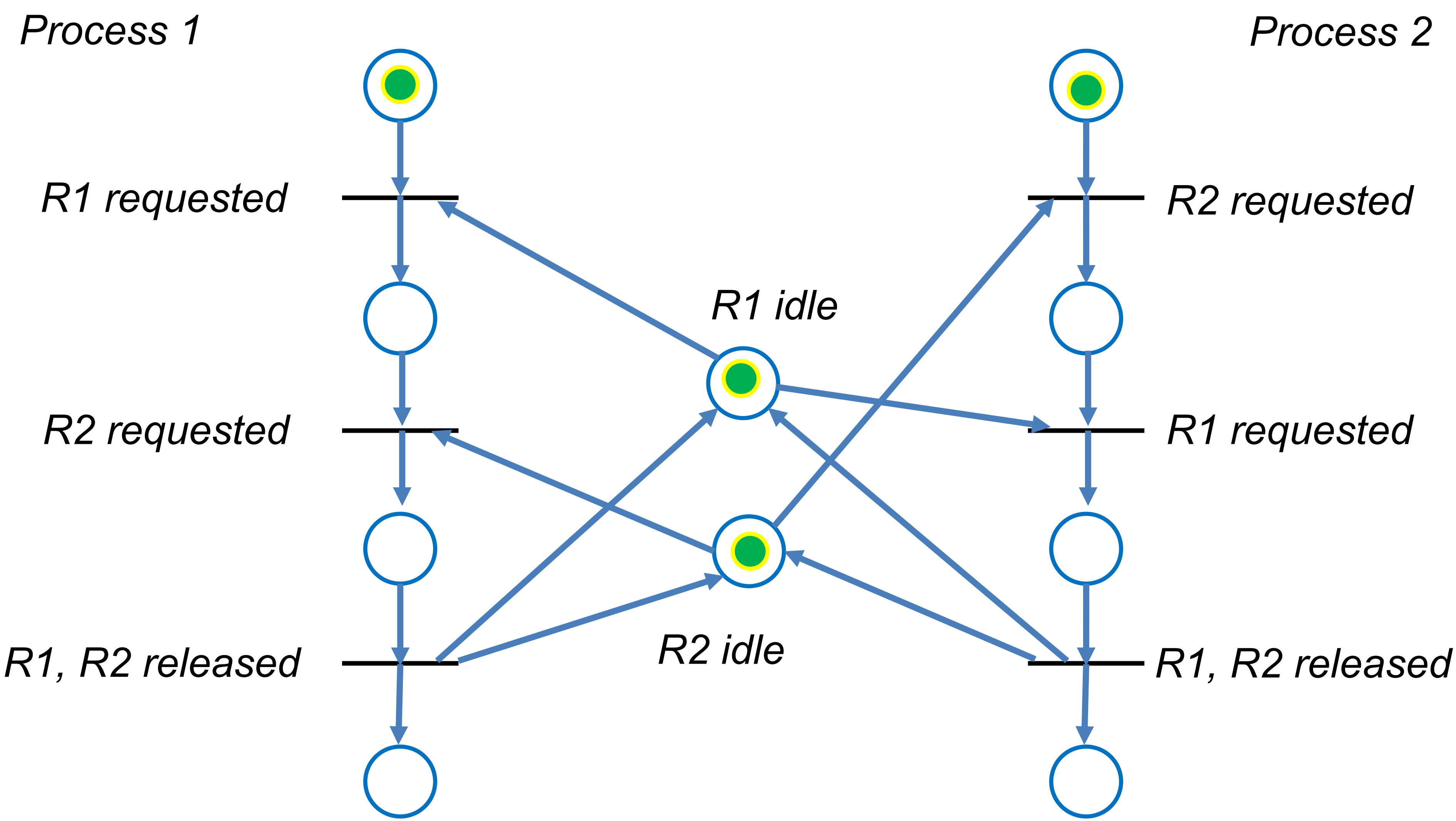
1. R1 allocated to Process 1

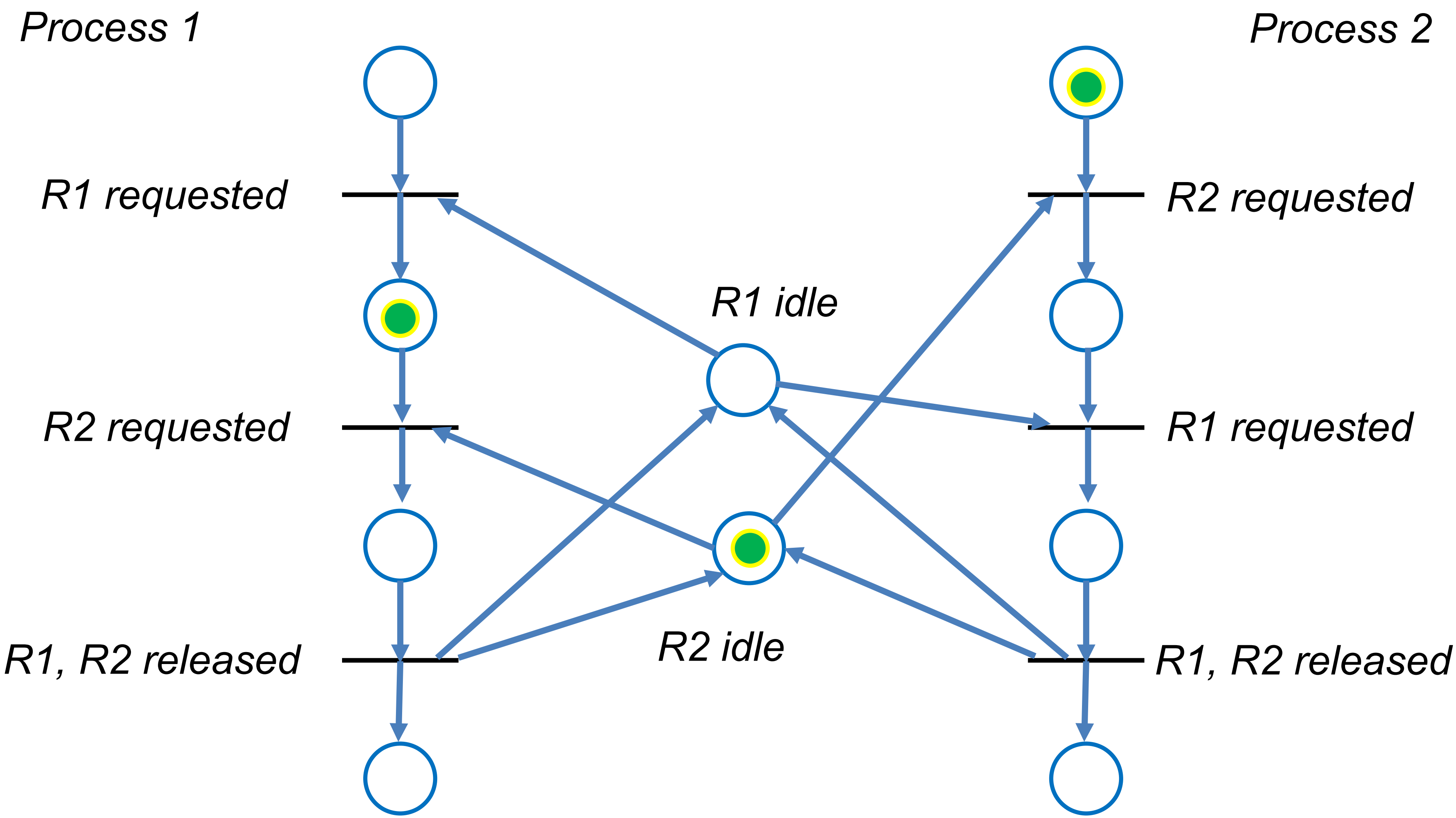
2. R2 allocated to Process 2

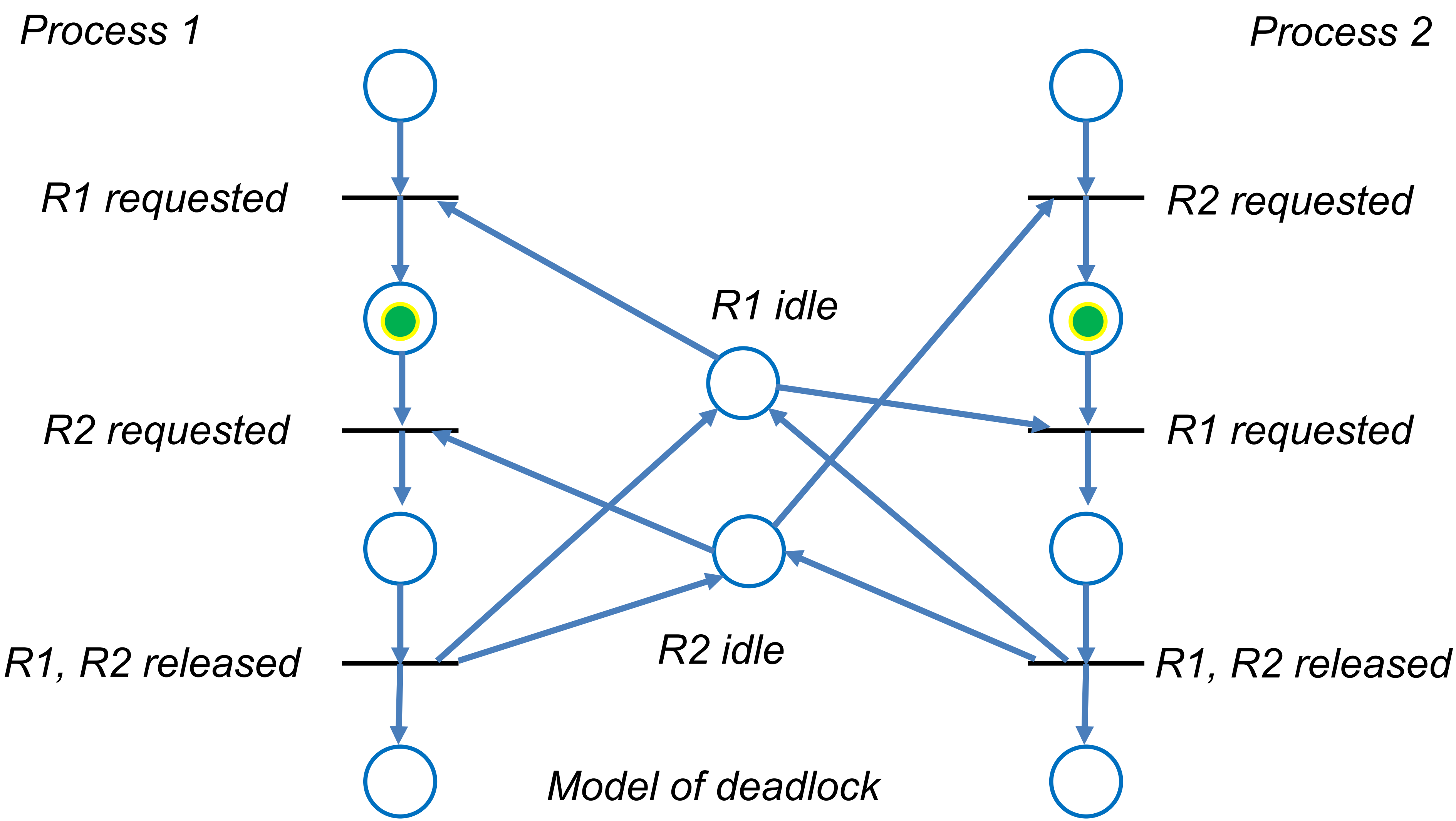
R1, R2 released

R1, R2 released

Deadlock

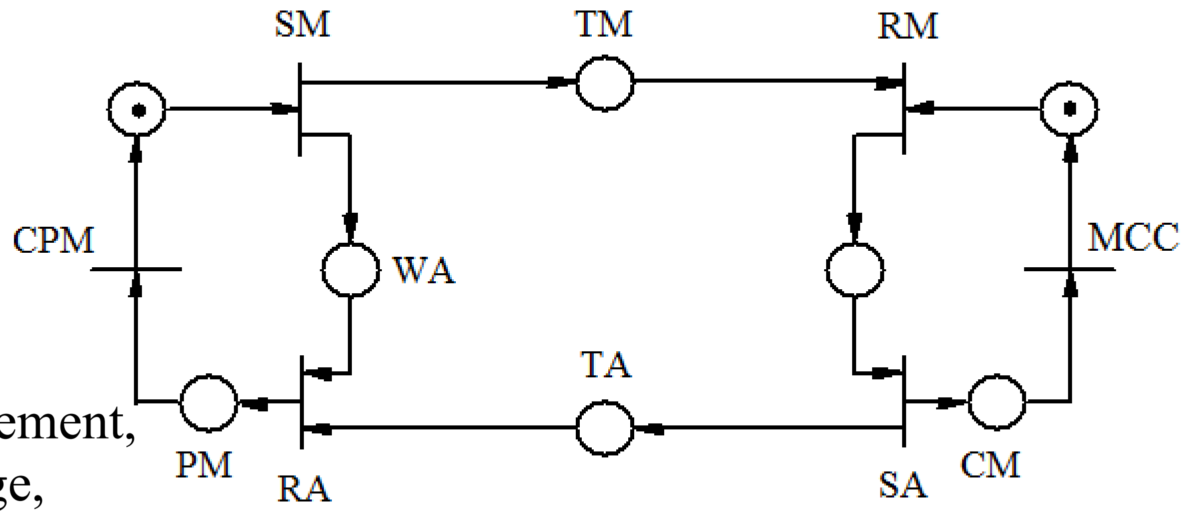

Process 1
Process 2
R1 requested
R2 requested
R1 idle
R2 requested
R1 requested
R1, R2 released
R2 idle
R1, R2 released


Process 1
Process 2
R1 requested
R2 requested
R1 idle
R2 requested
R1 requested
R1, R2 released
R2 idle
R1, R2 released



Model of deadlock

Concurrent systems modelling using Petri nets



Simple communication protocol

PM – production of a message,
CPM – completion of the message production,
SM – sending the message,
WA – waiting on an acknowledgement,
TM – transmission of the message,
RM – receiving the message,
SA – sending the acknowledgement,
CM – consumption of the message,
MCC – the message consumption completion,
TA – transmission of the acknowledgement,
RA – receiving the acknowledgement.

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Information Systems Analysis

Jan Magott

Behavioural properties of Petri nets

Choose yourself and new tech

Properties of Petri nets

- Definition

A *Petri net* is a 6-tuple:

$$N = \langle P, T, F, W, C, M_0 \rangle$$

P – a set of places,

T – a set of transitions,

$F \subseteq (P \times T) \cup (T \times P)$ – a set of arcs,

$W : F \rightarrow \{1,2,\ldots\}$ – an arc weight function,

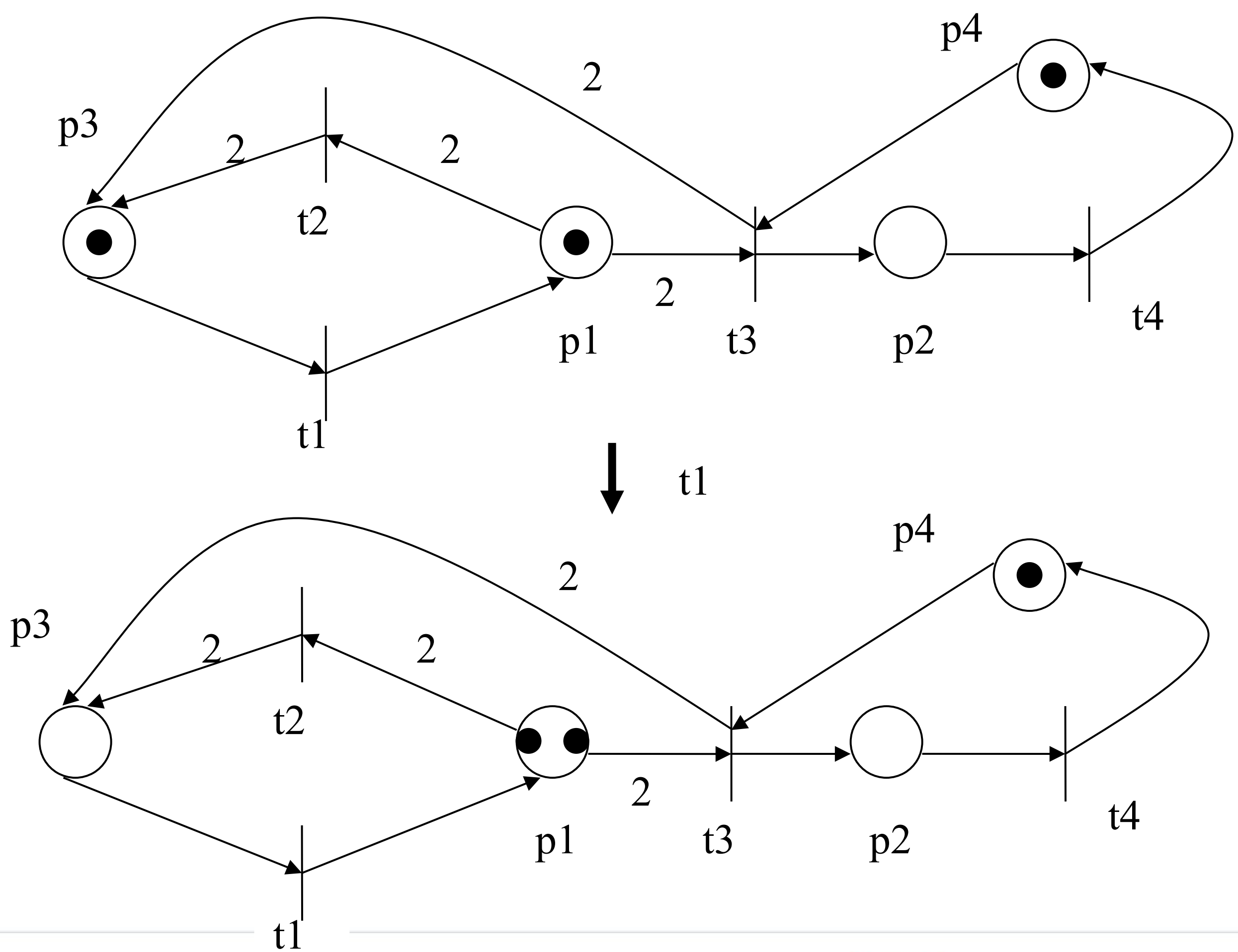
$C : P \rightarrow \{1,2,\ldots\}$ – a place capacity function,

$M_0 : P \rightarrow \{0,1,2,\ldots\}$ – an initial marking function.

Properties of Petri nets (Behavioural properties)

$(\forall p \in P)(C(p) = \infty)$

i.e., the place capacity function does not impose limitations on firing of transitions.



Properties of Petri nets (Behavioural properties)

- A *set of input* (*output*) *places* of a transition $t \in T$ is

  $^{\bullet}t = \{p \in P \mid W(p,t) > 0\}$ ($t^{\bullet} = \{p \in P \mid W(t,p) > 0\}$).
- A *set of input* (*output*) *transitions* of a place $p \in P$ is

  $^{\bullet}p = \{t \in T \mid W(t,p) > 0\}$ ($p^{\bullet} = \{t \in T \mid W(p,t) > 0\}$).

Assumption: $(\forall p \in P)(C(p) = \infty)$;A transition *t is enabled in marking M* if

$$(\forall p \in {}^{\bullet}t)(M(p) \geq W(p,t))$$

(each input place is marked with at least $W(p,t)$ tokens).

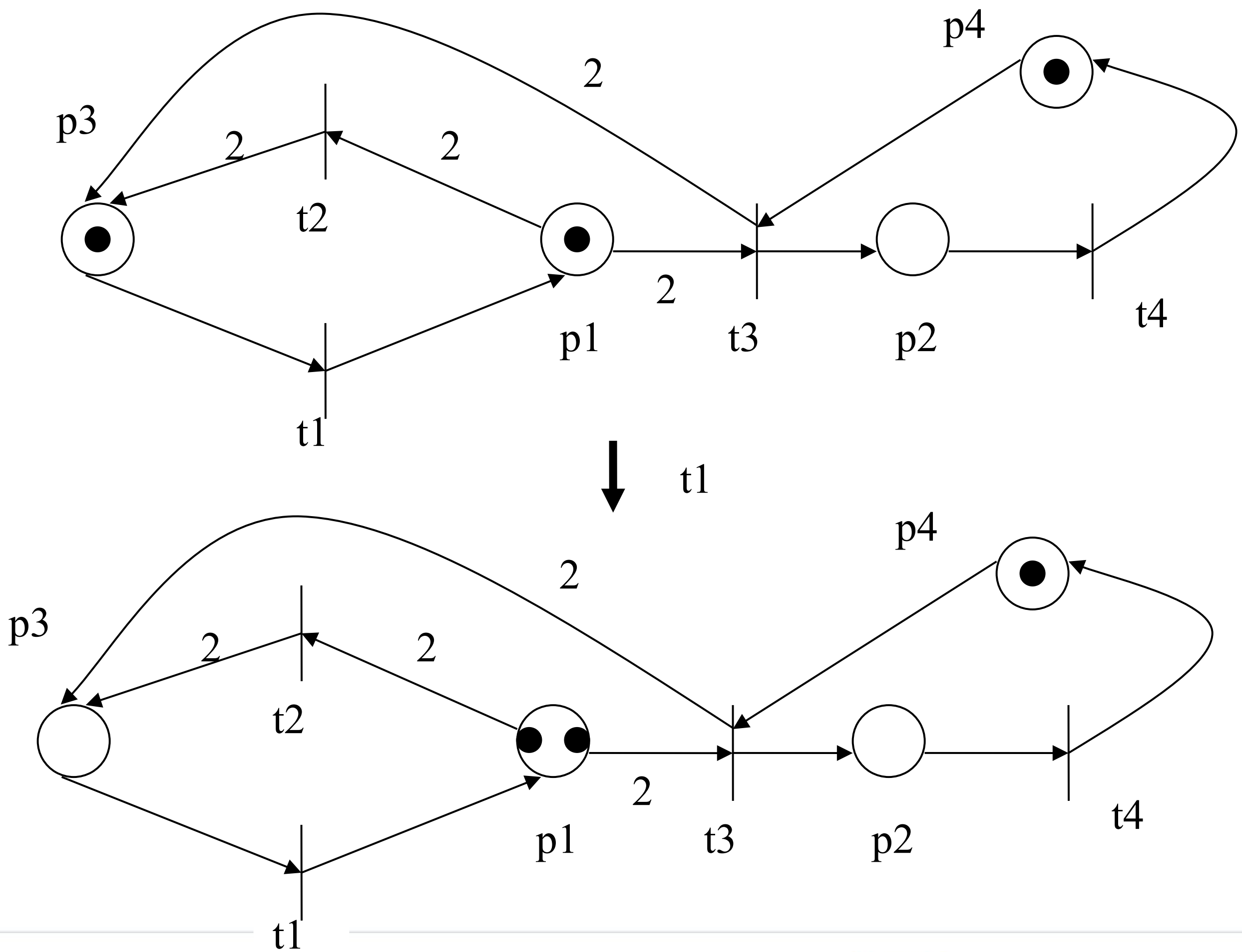
- *Firing of enabled transition t* causes the following change of marking:

  removes $W(p,t)$ tokens from each input place $p \in {}^{\bullet}t$,
  adds $W(t,p)$ tokens to each output place $p \in t^{\bullet}$.

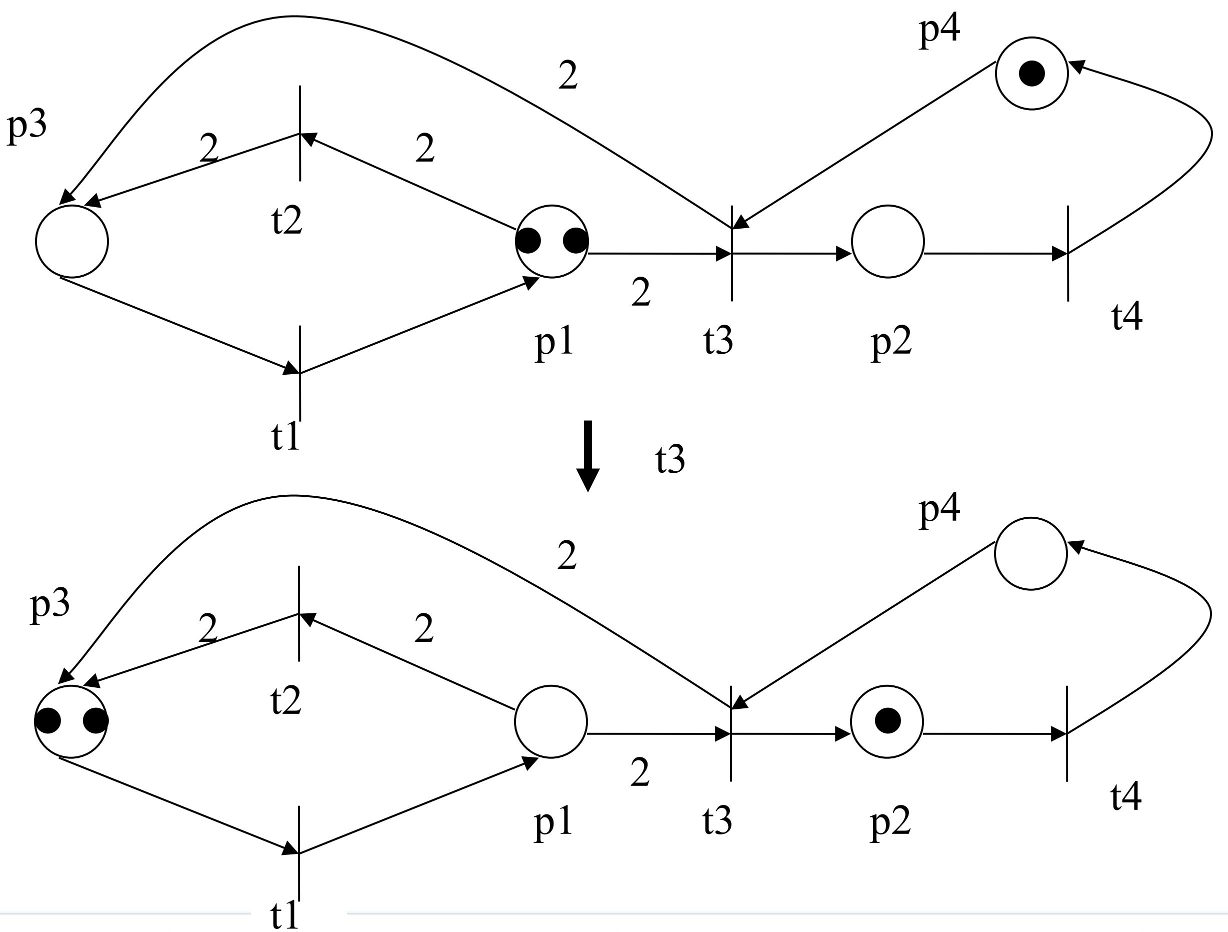
Properties of Petri nets (Behavioural properties)

$$(\forall p \in P)(C(p) = \infty)$$

i.e., the place capacity function does not impose limitations on firing of transitions.

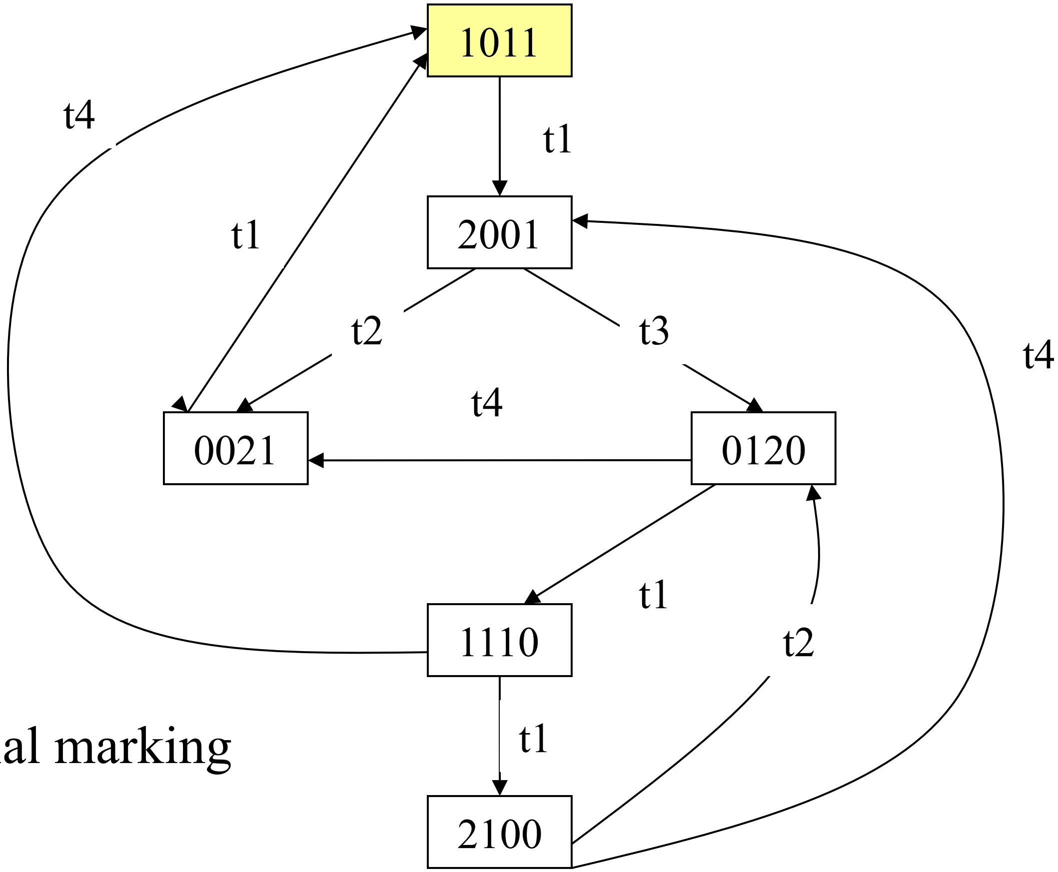


Properties of Petri nets (Behavioural properties)



Properties of Petri nets (Behavioural properties)

Reachability graph



- 1011 – initial marking

Properties of Petri nets (Behavioural properties)

Reachability graph

Directed graph $G<V,A>$ is strongly connested if for each ordered pair of vertices $< v_i, v_j >$ there exists a directed path from vertex v_i to v_j.

Properties of Petri nets (Behavioural properties)

- $M \xrightarrow{t} M'$ – transition *t* has been fired in marking *M*, and marking *M'* has been received

- Definition

  $\sigma = t_1 t_2 \ldots t_n$ is a *firing sequence* for marking *M*, if

  $$M = M_1 \xrightarrow{t_1} M_2 \xrightarrow{t_2} M_3 \ldots \xrightarrow{t_{n-1}} M_n \xrightarrow{t_n} M_{n+1} = M'$$

- Notation: $M \xrightarrow{\sigma} M'$ or $M[\sigma > M'$

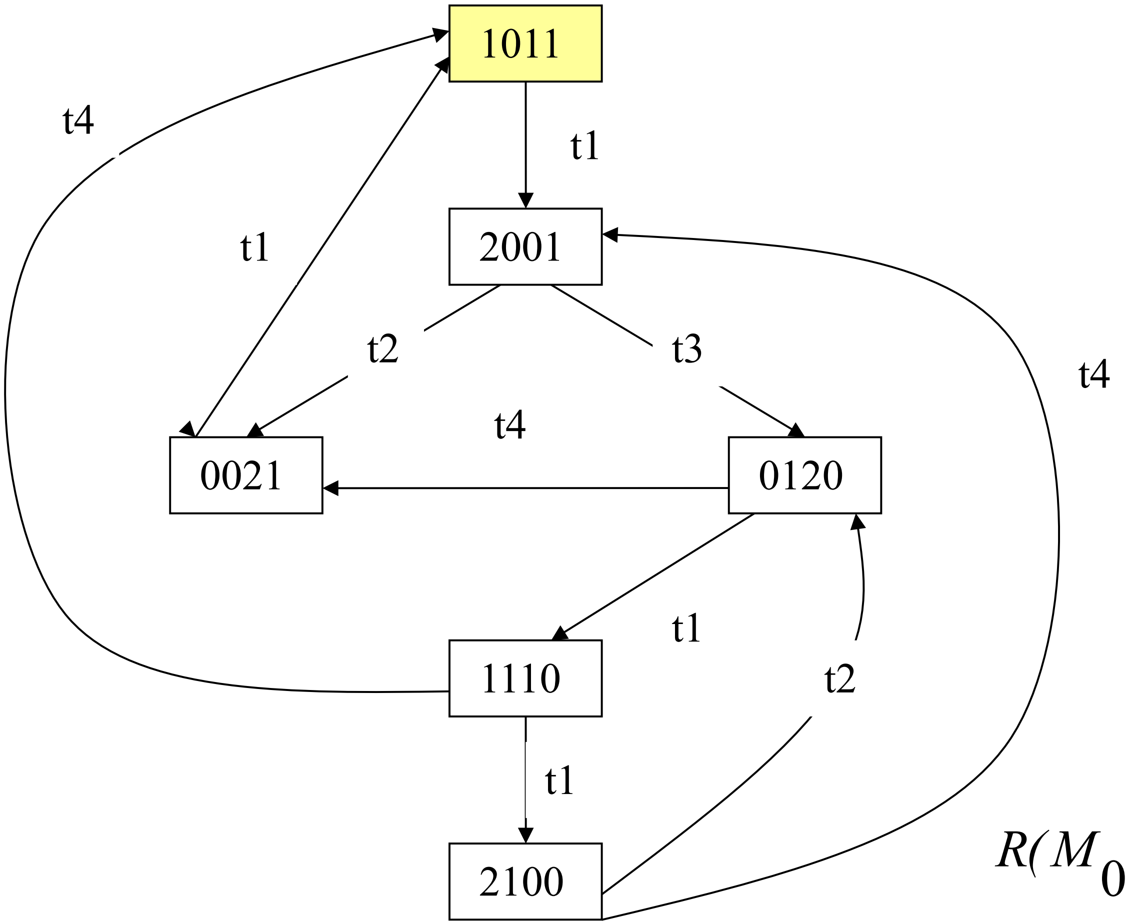
- Definition

  $R(M)$ – a *reachability set* for marking *M*
  (set of all markings that are reachable from marking *M*)

  $$R(M) = \{ M' \mid (\exists \sigma \in T^*)(M \xrightarrow{\sigma} M') \}$$

  where T^* – a set of finite words over an alphabet *T* with an empty word.

Properties of Petri nets (Behavioural properties)



- Firing sequences

$$1011 \xrightarrow{t1\ t3\ t4} 0021$$

$$2001 \xrightarrow{t3\ t1\ t1\ t2\ t4\ t1} 1011$$

- $t1\ t3\ t2$ is not any firing sequence

- Reachability set

$$R(M_0) = \{1011, 2001, 0021, 0120, 1110, 2100\}$$

$$(\forall M \in R(M_0))(R(M) = R(M_0))$$

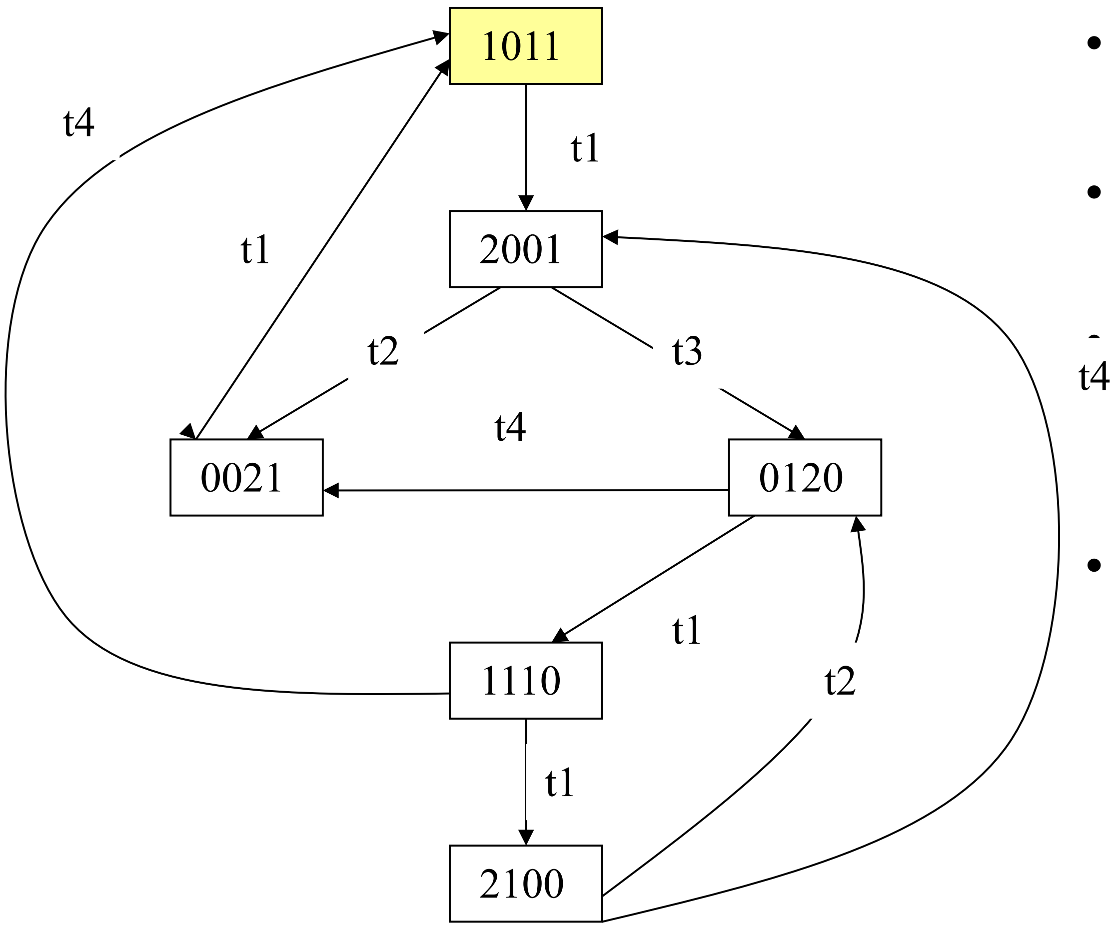
Properties of Petri nets (Behavioural properties)

- Definition
  The *Reachability problem*
  Input: Petri net $N = <P, T, F, W, C, M_0>$
  Output: $M \in R(M_0)$?

- Theorem (1981) (Foundations of Petri net theory created by C. A. Petri in 1962.)
  The Reachability problem for $N = <P, T, F, W, M_0>$ is decidable.
  Its computational complexity is exponential.

- Definition
  The *Submarking reachability problem*
  Input: Petri net $N = <P, T, F, W, C, M_0>$, $P' \subset P$, $M' : P' \to \{0,1,2,\ldots\}$
  Output: $(\exists M \in R(M_0))(M' = M|_{P'})$?
  where $M|_{P'}$ is restriction of marking function *M* to set *P'*.

Properties of Petri nets (Behavioural properties)



- Submarking reachability

- Submarking

  $M'_1(p2)=0, \quad M'_1(p3)=2$

  is reachable.

- Submarking

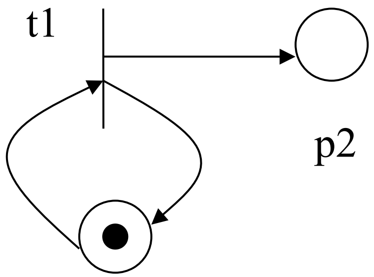
  $M'_1(p1)=1, \quad M'_1(p3)=2$

  is not reachable.

Properties of Petri nets (Behavioural properties)

- A Petri net $N = \langle P, T, F, W, C, M_0 \rangle$ is bounded iff

$$(\exists k \in \{1,2,\ldots\})(\forall M \in R(M_0))(\forall p \in P)(M(p) \leq k)$$



- Is the following definition:

$$(\forall M \in R(M_0))(\exists k \in \{1,2,\ldots\})(\forall p \in P)(M(p) \leq k)$$

equivalent to the above?

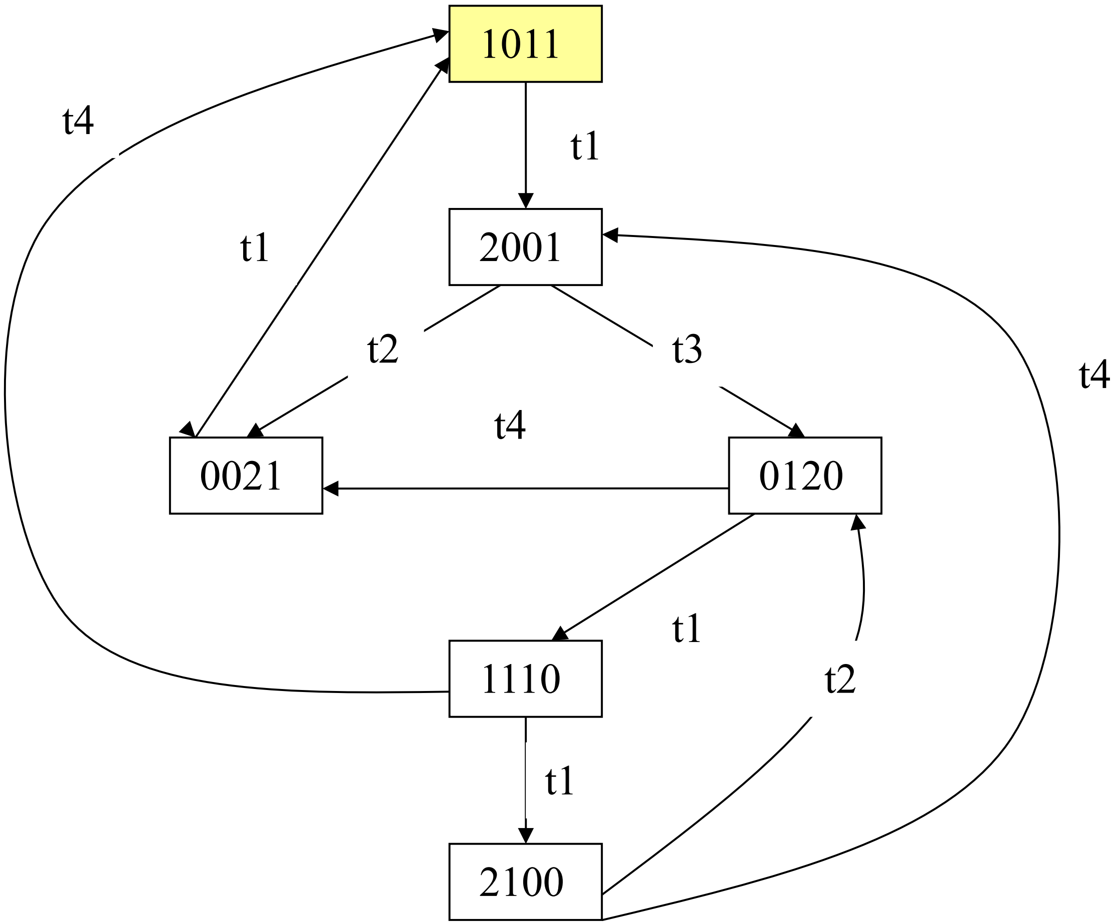
- A Petri net is *k*-bounded iff

$$(\forall M \in R(M_0))(\forall p \in P)(M(p) \leq k)$$

- A Petri net is safe if it is 1-bounded.

- Practical aspect. Problem: *Is Petri net safe?* can be used in verification problem: *Can buffer of capacity equal to* 1 *be overflowed*?

Properties of Petri nets (Behavioural properties)



- This net is:

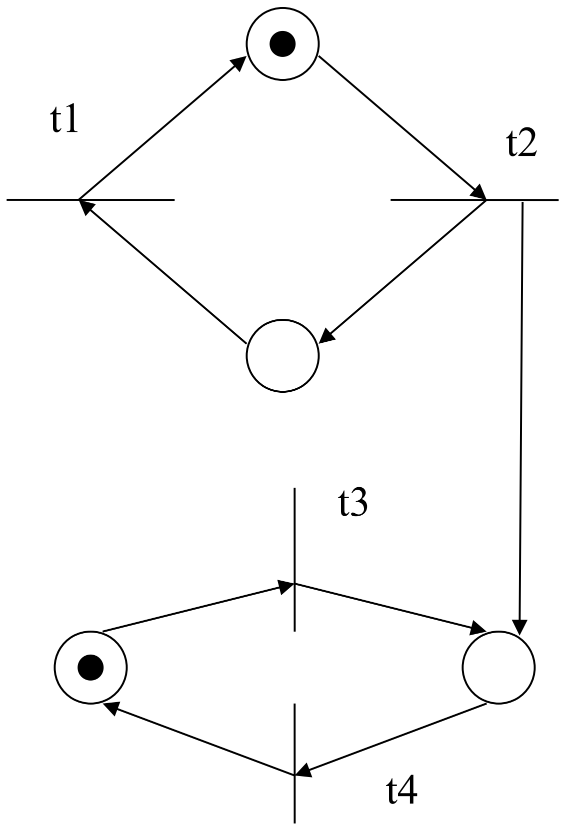
  bounded,

  4-bounded,

  2-bounded,

  is not safe.

Properties of Petri nets (Behavioural properties)



- This Petri net is not bounded.
- Why?

Properties of Petri nets (Behavioural properties)

- Definition

  A *transition t* is *live* iff

  $$(\forall M \in R(M_0))(\exists M' \in R(M))(M' \xrightarrow{t})$$

  where $M' \xrightarrow{t}$ - transition t can be fired for marking M'.

- Definition

  A *Petri net* is *live* if each of its transitions are live.

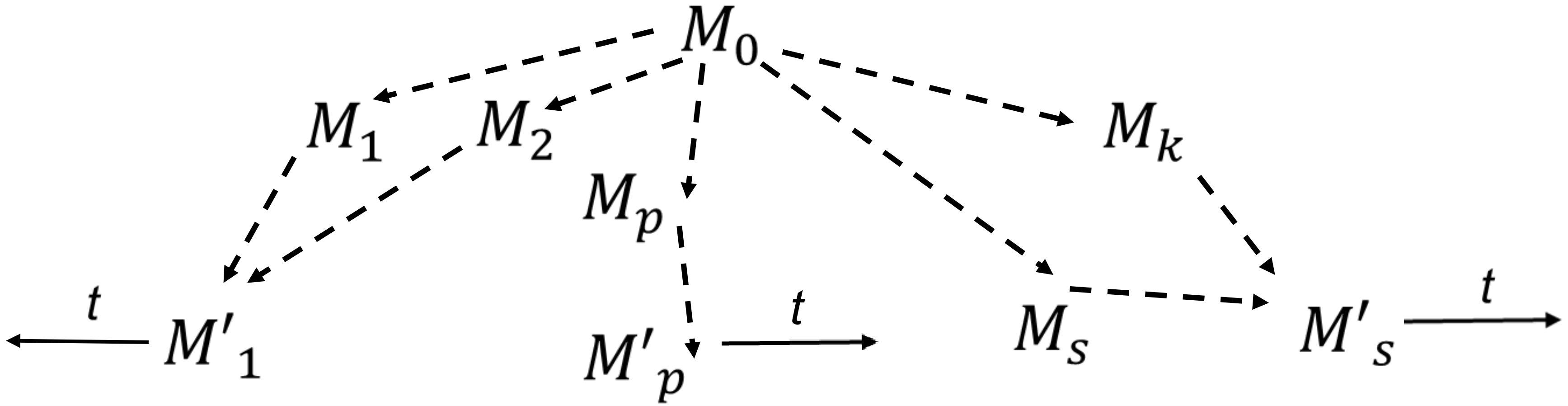
- Liveness means that there are no deadlocks in the net.

- Definition

A *transition* t is *live* iff

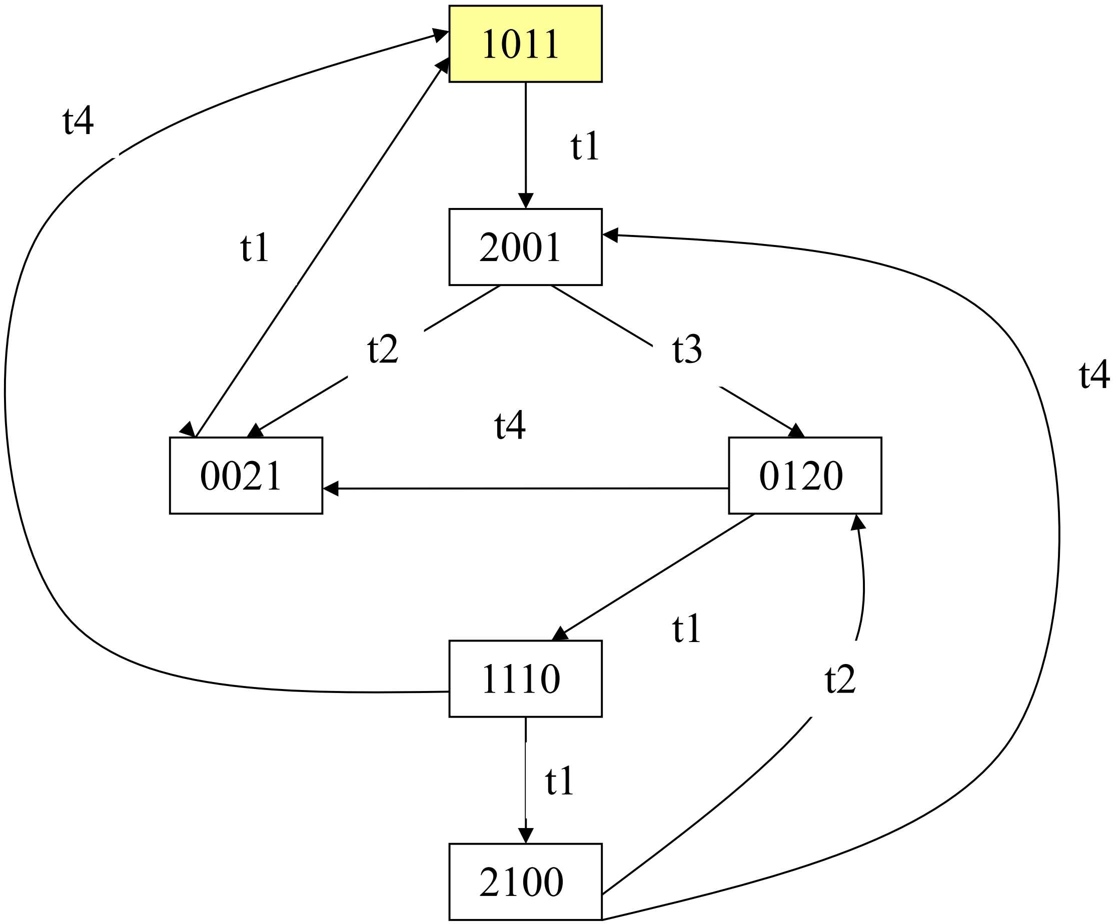
$$(\forall M \in R(M_0))(\exists M' \in R(M))(M' \xrightarrow{t})$$

where $M' \xrightarrow{t}$ - transition t can be fired for marking M'.



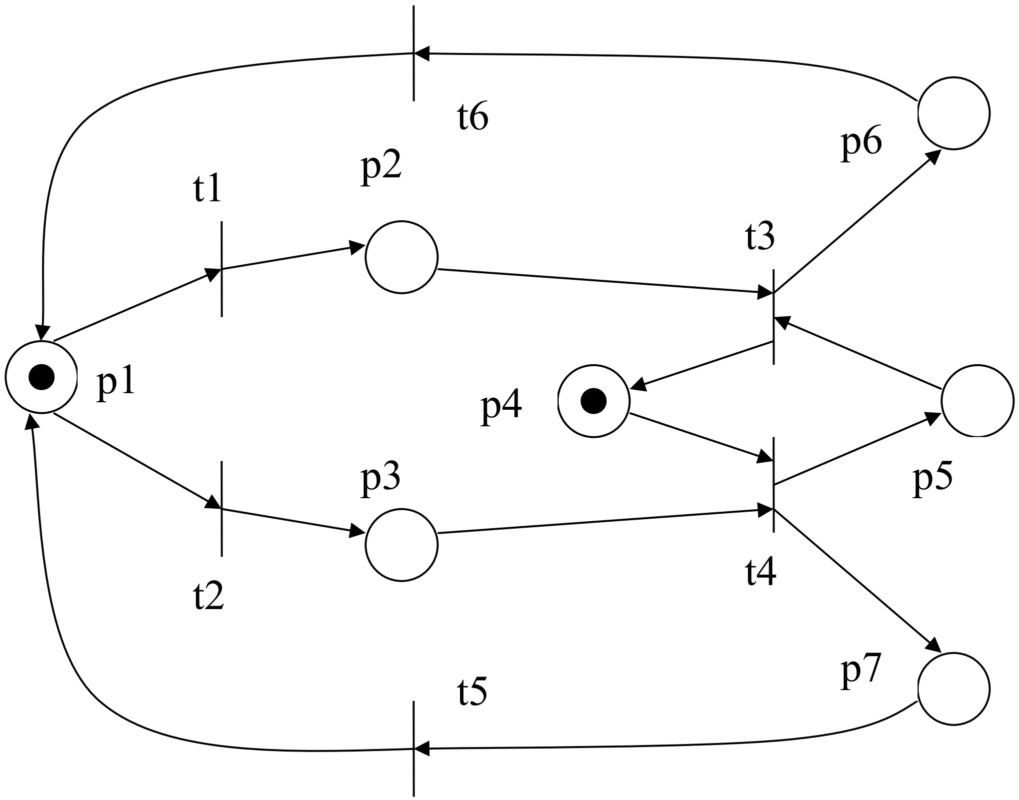
HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

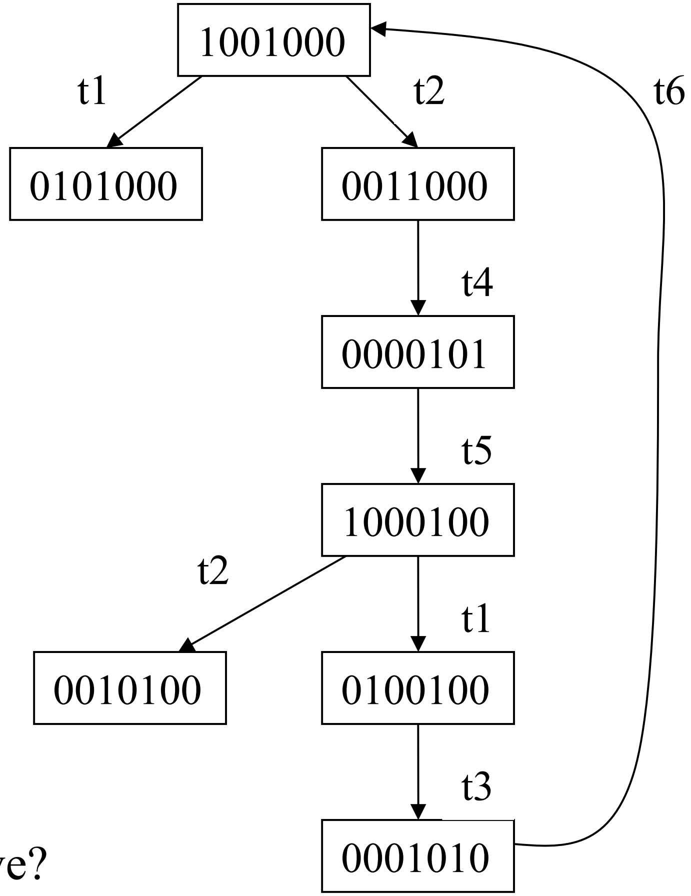
Properties of Petri nets (Behavioural properties)



- Is transition t3 live?
- Are all transitions live?
- This Petri net is live.

Properties of Petri nets (Behavioural properties)





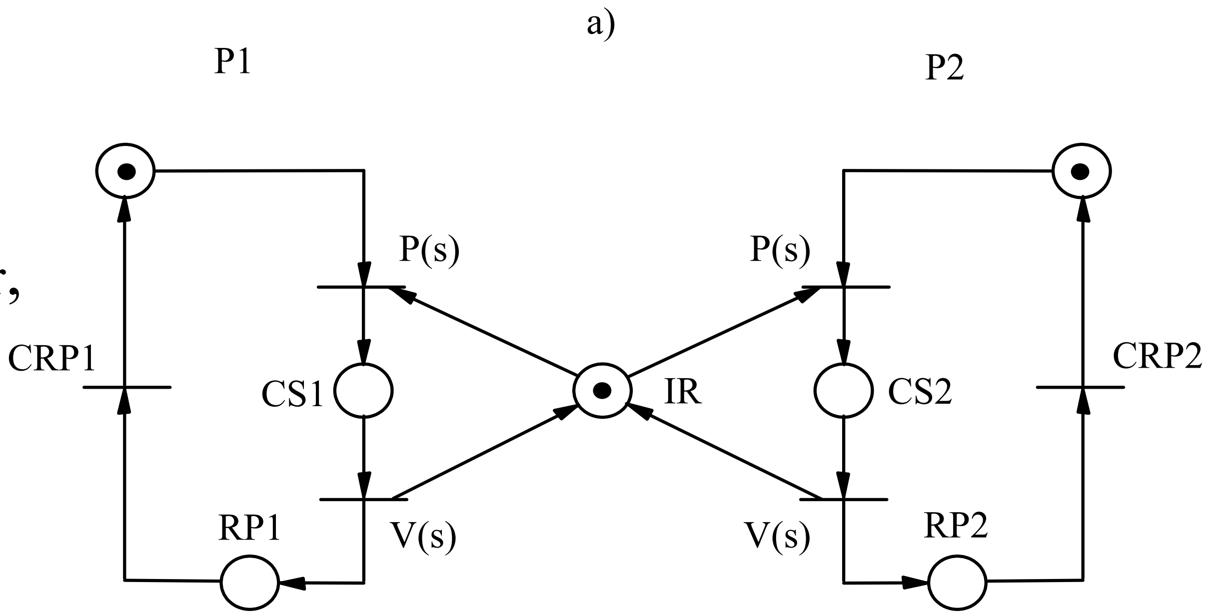
Which transition is live?

The Petri net is not live

Concurrent systems modelling using Petri nets

- Mutual exclusion of two cyclic sequential processes (cont.)

CS – the process is in critical section,
RP – the process is in its remainder,
CRP – completion of the process remainder,
P(*s*), *V*(*s*) – semaphore operations,
IR – critical resource is idle.



Transitions with labels *P*(*s*) are enabled.

Is this Petri net live ?

Properties of Petri nets (Behavioural properties)

- Definition
  The *Liveness problem*
  Input: Petri net N
  Output: Is N live?

- Theorem (1975)
  The Liveness problem for $N = <P,T,F,W,M_0>$ is equivalent to the reachability problem.

- Theorem (1981)
  The Reachability problem for $N = <P,T,F,W,M_0>$ is decidable.
  Its computational complexity is exponential.

- Conclusion
  The Liveness problem for $N = <P,T,F,W,M_0>$ is decidable.

Properties of Petri nets (Behavioural properties)

- Definition

  A Petri net is *reversible* if

$$(\forall M \in R(M_0))(M_0 \in R(M)).$$

- Reversibility is a strong property. Hence, a weaker one has been introduced.

- Definition

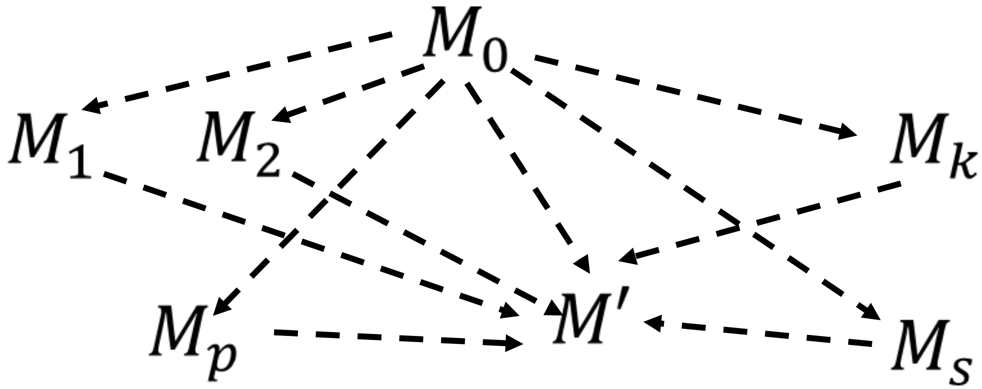
  Marking *M'* is a so-called *home marking* if

$$(\forall M \in R(M_0))(M' \in R(M)).$$

- Definition

  Marking *M'* is a so-called *home marking* if

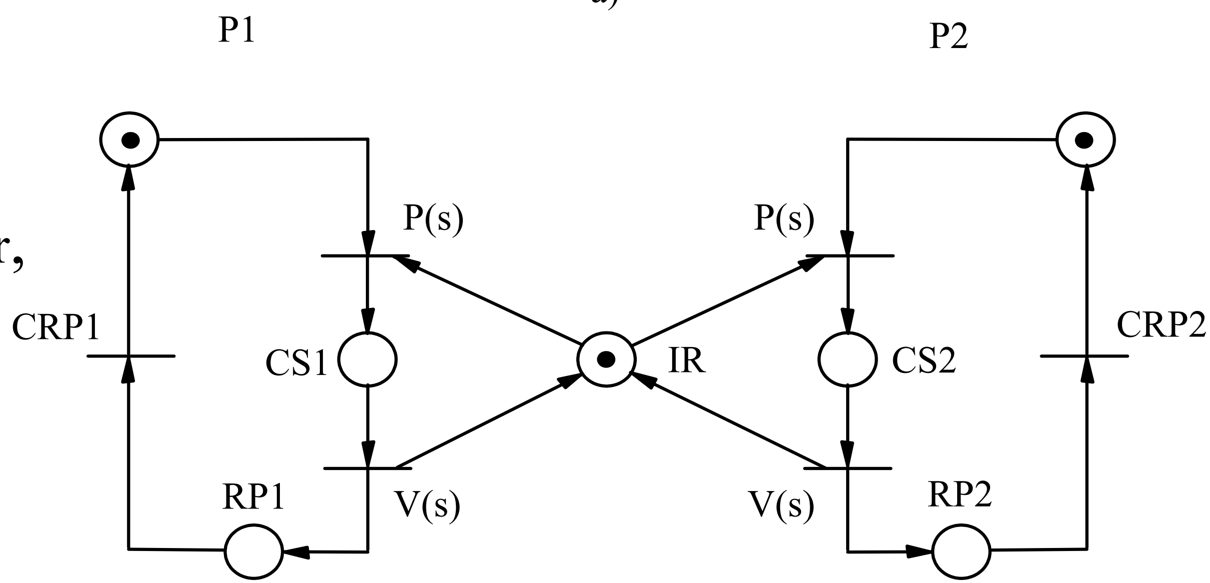
$$(\forall M \in R(M_0))(M' \in R(M)).$$



Concurrent systems modelling using Petri nets

- Mutual exclusion of two cyclic sequential processes (cont.)

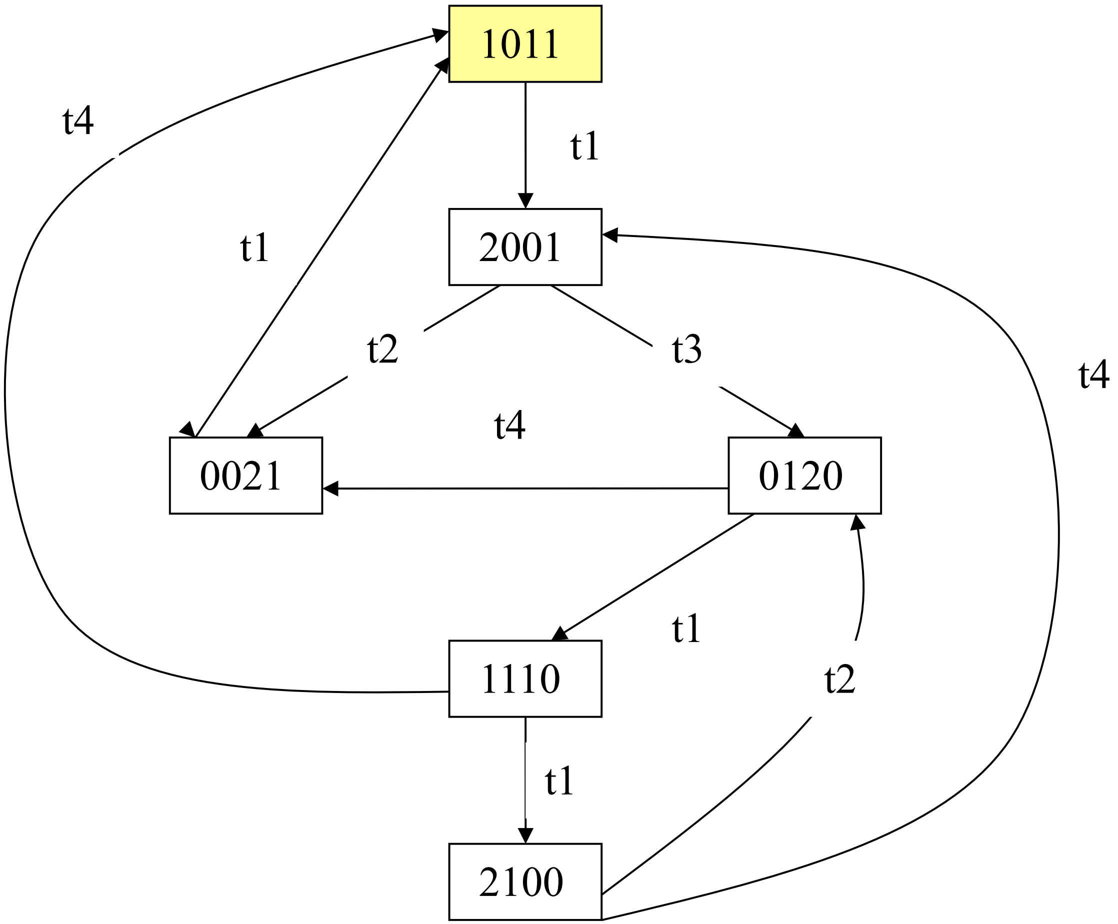
a)



CS – the process is in critical section,
RP – the process is in its remainder,
CRP – completion of the process remainder,
P(*s*), *V*(*s*) – semaphore operations,
IR – critical resource is idle. Transitions with labels *P*(*s*) are enabled.

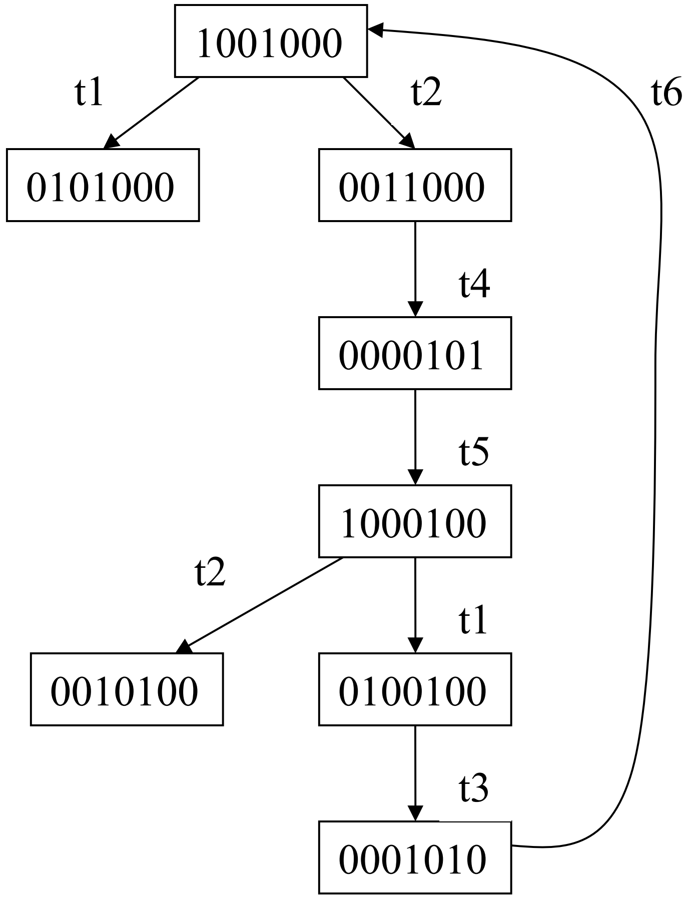
Is this Petri net reversible?

Properties of Petri nets (Behavioural properties)



- Is this reachability graph of a reversible net?
- Each reachable marking is the home one.

Properties of Petri nets (Behavioural properties)



- Is this reachability graph of a reversible net?
- There is no home marking.

Properties of Petri nets (Behavioural properties)

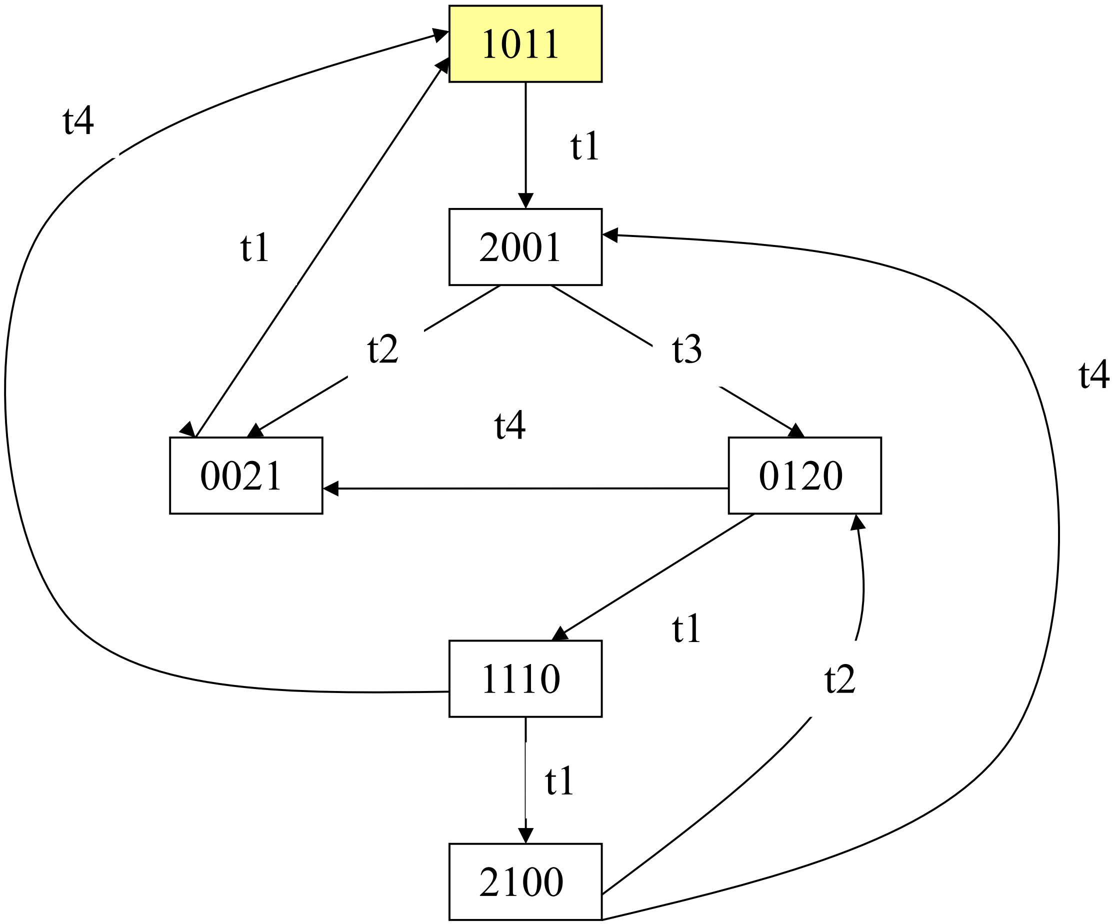
- Definition

  A *marking* M is *coverable* if

$$(\exists M' \in R(M_0))(\forall p \in P)(M(p) \leq M'(p))$$

- For Petri net without capacity function, if transition t can be fired in marking M, then t can be fired in M'.

Properties of Petri nets (Behavioural properties)



- Which of these markings are coverable?

- 1101,
- 0110,
- 0022,
- 0011,
- 1001.

Properties of Petri nets (Behavioural properties)

- Definition

  A Petri net is *persistent* if

$$(\forall M \in R(M_0))(\forall t1, t2 \in T)((M \xrightarrow{t1} \wedge M \xrightarrow{t2}) \Rightarrow M \xrightarrow{t1t2}).$$

- For persistent Petri nets, once enabled transition can become disabled by its firing only.

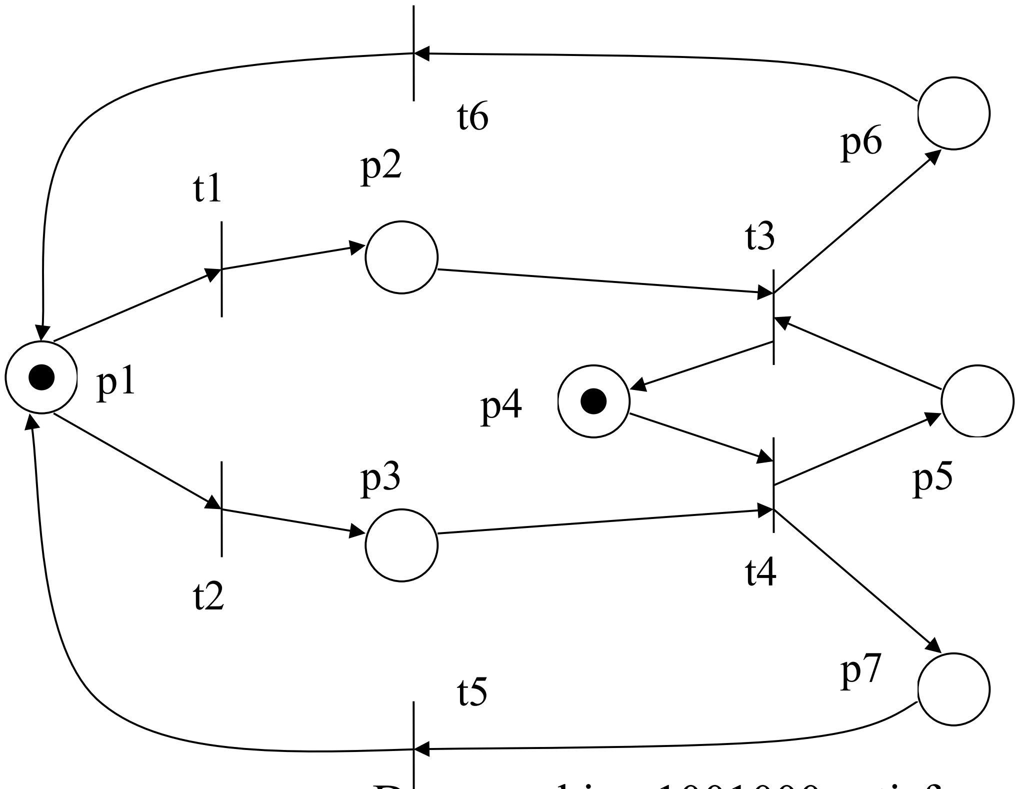
Properties of Petri nets (Behavioural properties)

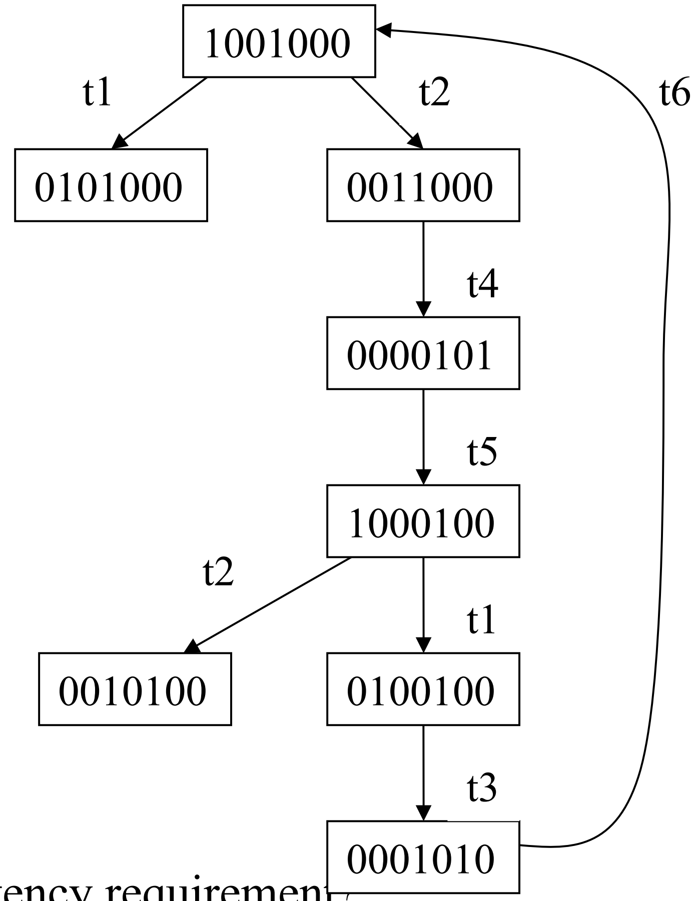
$$Condition \Rightarrow Conclusion$$

Condition (Premise)	Conclusion	Implication
true	true	true
false	false	true
false	true	true
true	false	false

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Properties of Petri nets (Behavioural properties)



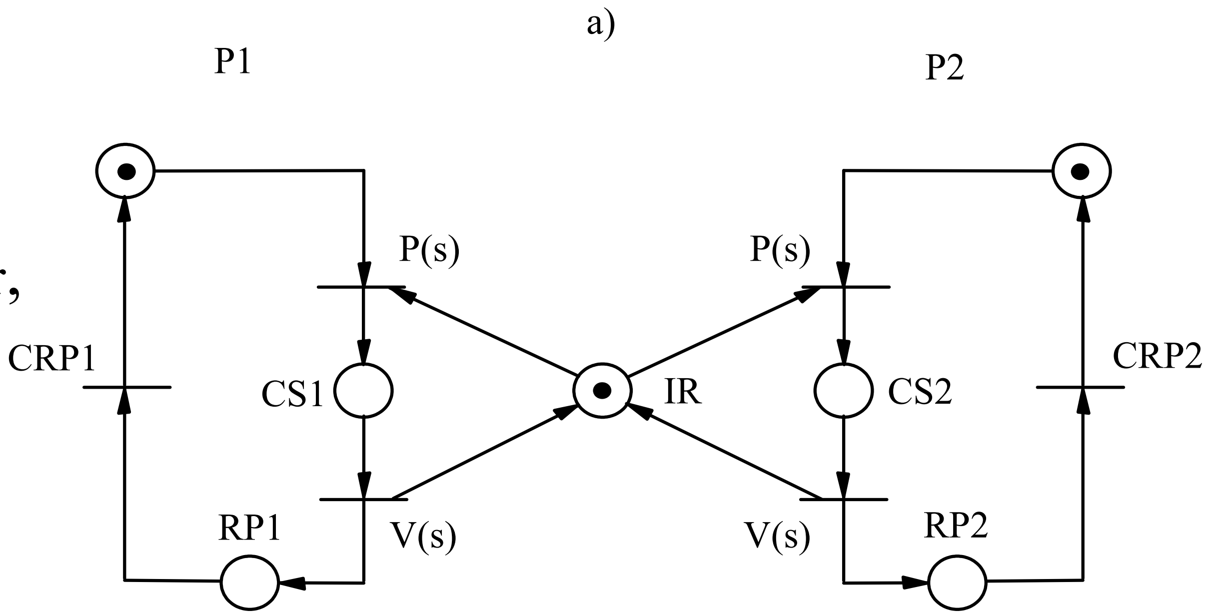


Does marking 1001000 satisfy persistency requirement?
Do the other markings satisfy persistency requirement?
Is the Petri net persistent?

Concurrent systems modelling using Petri nets

- Mutual exclusion of two cyclic sequential processes (cont.)

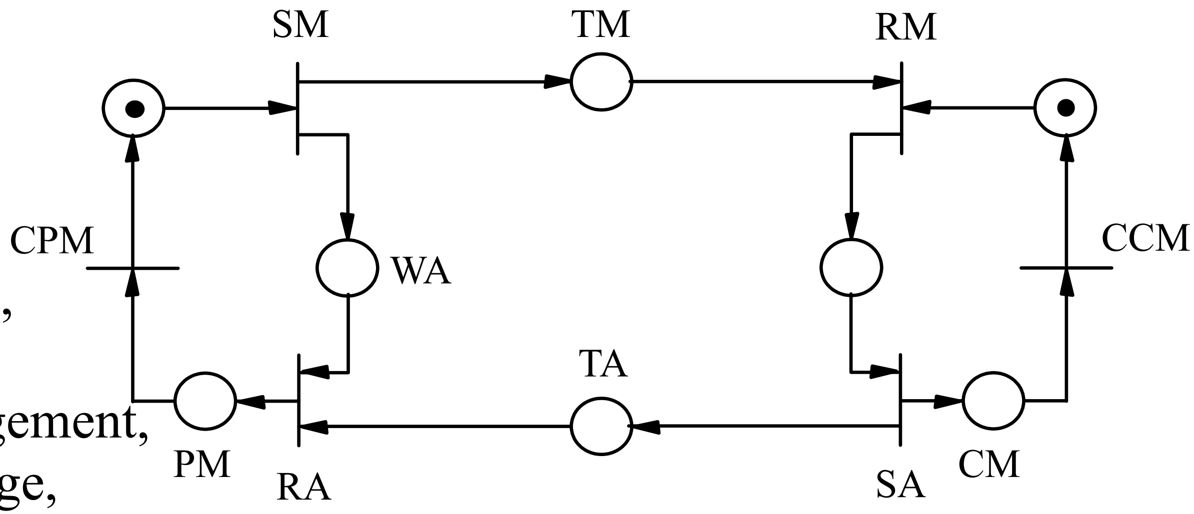
CS – the process is in critical section,
RP – the process is in its remainder,
CRP – completion of the process remainder,
P(*s*), *V*(*s*) – semaphore operations,
IR – critical resource is idle.



Transitions with labels *P*(*s*) are enabled.

Is this Petri net persistent?

Concurrent systems modelling using Petri nets



Simple communication protocol

PM – production of a message,
CPM – completion of the message production,
SM – sending the message,
WA – waiting on an acknowledgement,
TM – transmission of the message,
RM – receiving the message,
SA – sending the acknowledgement,
CM – consumption of the message,
CCM – the message consumption completion,
TA – transmission of the acknowledgement,
RA – receiving the acknowledgement.

Is this Petri net persistent?

Properties of Petri nets (Behavioural properties)

- Definition

  A *synchronization distance between transitions t*1, *t*2 of Petri net *N* is:

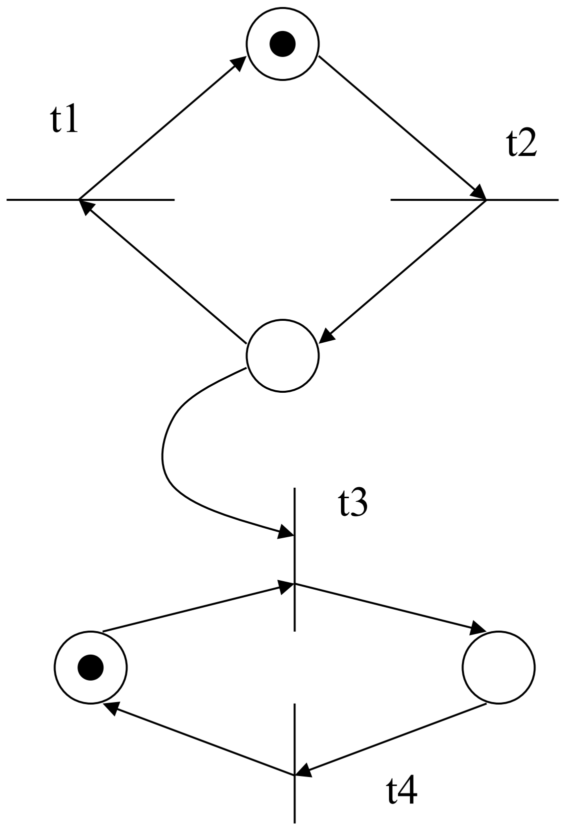
$$d_{12} = max_{\sigma \in \Sigma(N)} |\overline{\sigma}(t1) - \overline{\sigma}(t2)|$$

$\Sigma(N)$ – a set of all firing sequences for all reachable markings,

$\overline{\sigma}(t)$ – a number of firings of transition *t* in firing sequence *σ*.

.

Properties of Petri nets (Behavioural properties)



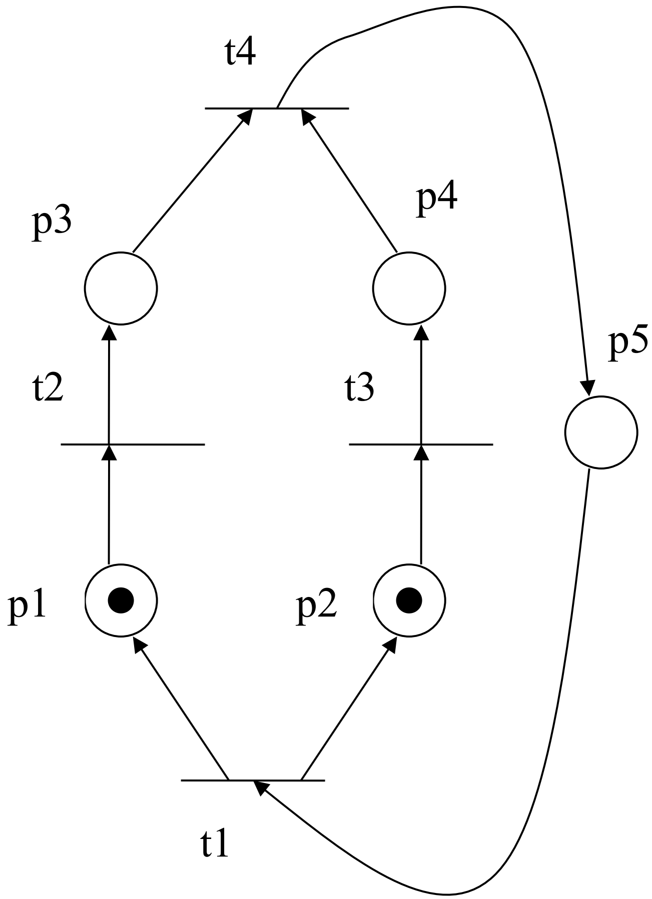
- Examples of synchronization distances between transitions

$$d_{12} = 1$$

$$d_{34} = 1$$

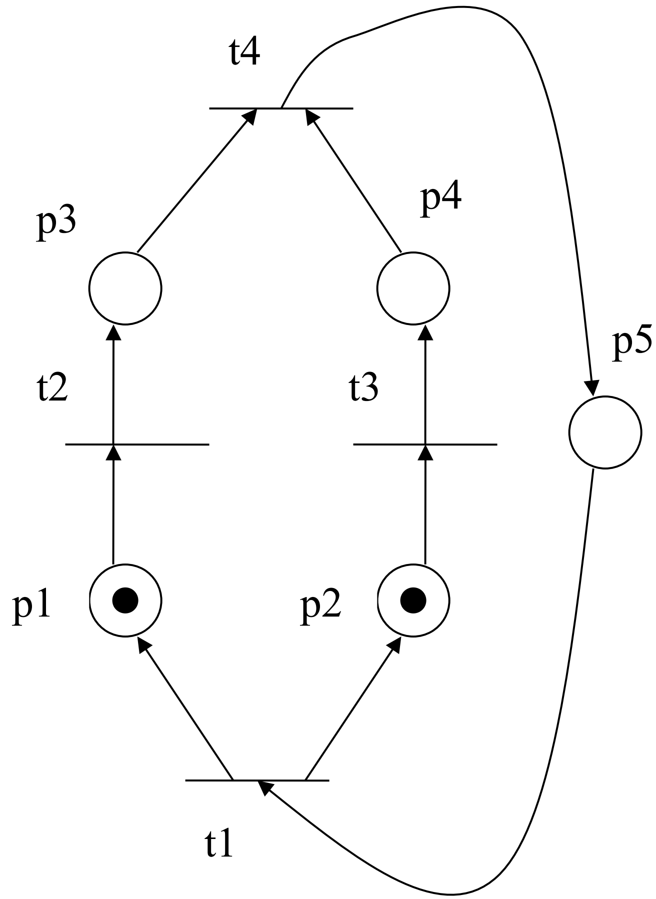
$$d_{13} = \infty$$

Properties of Petri nets (Behavioural properties)



- Is it true that $(\forall i, j \in \{\overline{1,4}\})(d_{ij} = 1)$?

Properties of Petri nets (Behavioural properties)



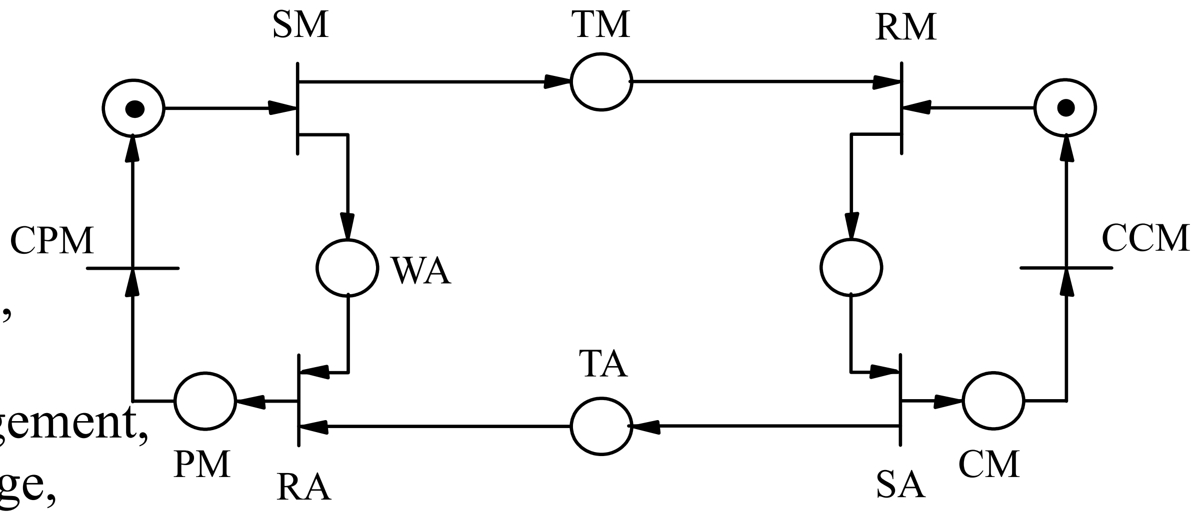
- Is it true that $(\forall i, j \in \overline{\{1,4\}})(d_{ij} = 1)$?
- Let us consider marking 10010.

$$(\exists \sigma = t_2 t_4 t_1 t_2)(\overline{\sigma}(t_2) = 2 \wedge \overline{\sigma}(t_3) = 0)$$

$$d_{23} = 2$$

Concurrent systems modelling using Petri nets

Simple communication protocol



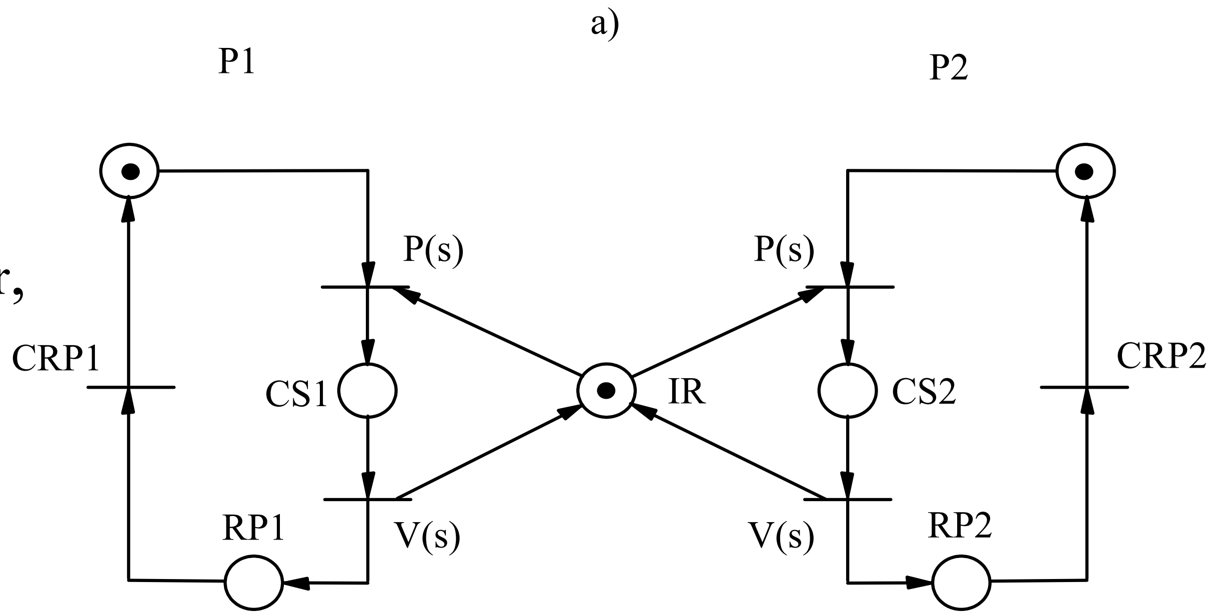
PM – production of a message,
CPM – completion of the message production,
SM – sending the message,
WA – waiting on an acknowledgement,
TM – transmission of the message,
RM – receiving the message,
SA – sending the acknowledgement,
CM – consumption of the message,
CCM – the message consumption completion,
TA – transmission of the acknowledgement,
RA – receiving the acknowledgement.

What is synchronization distance between SM and CCM ? 1 *or* 2*?*

Concurrent systems modelling using Petri nets

- Mutual exclusion of two cyclic sequential processes (cont.)

CS – the process is in critical section,
RP – the process is in its remainder,
CRP – completion of the process remainder,
P(*s*), *V*(*s*) – semaphore operations,
IR – critical resource is idle.



Transitions with labels *P*(*s*) are enabled.

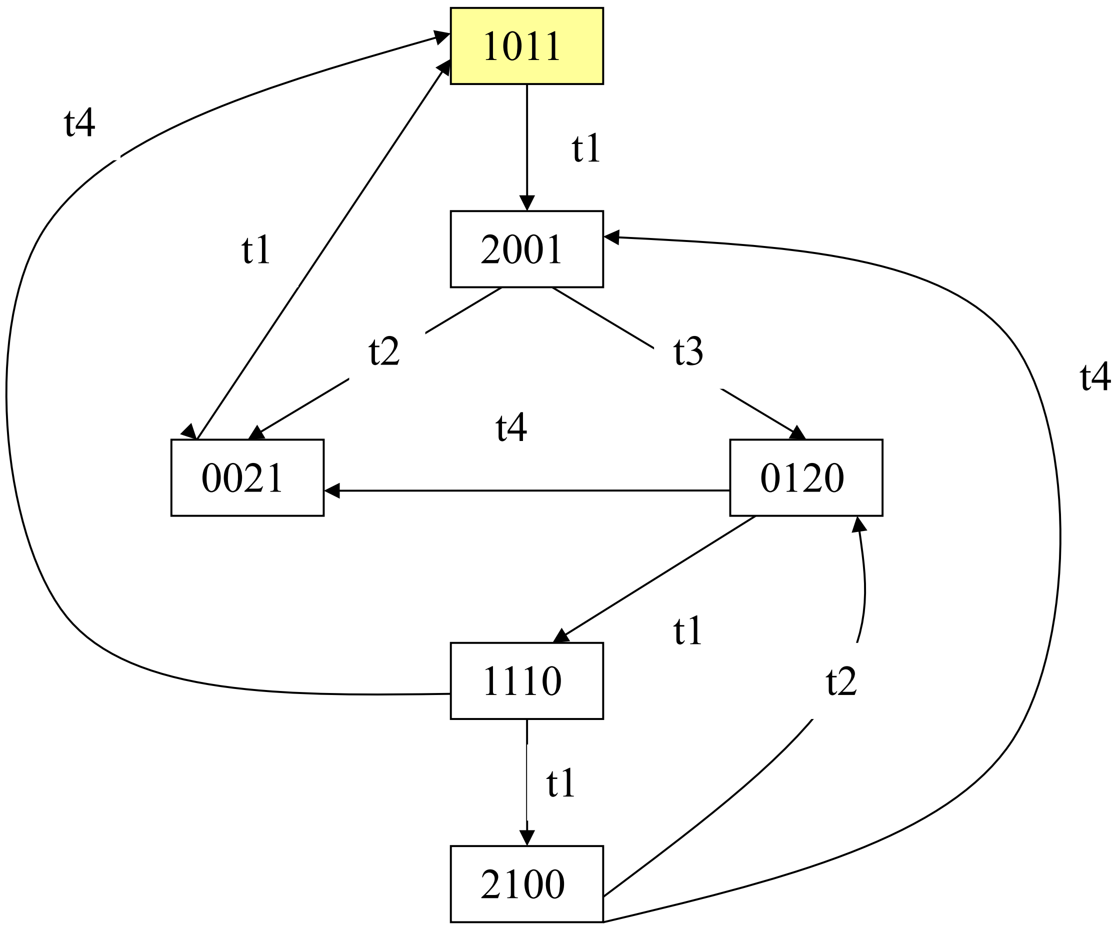
What is synchronization distance between CRP1 and CRP2 ?

Properties of Petri nets (Behavioural properties)

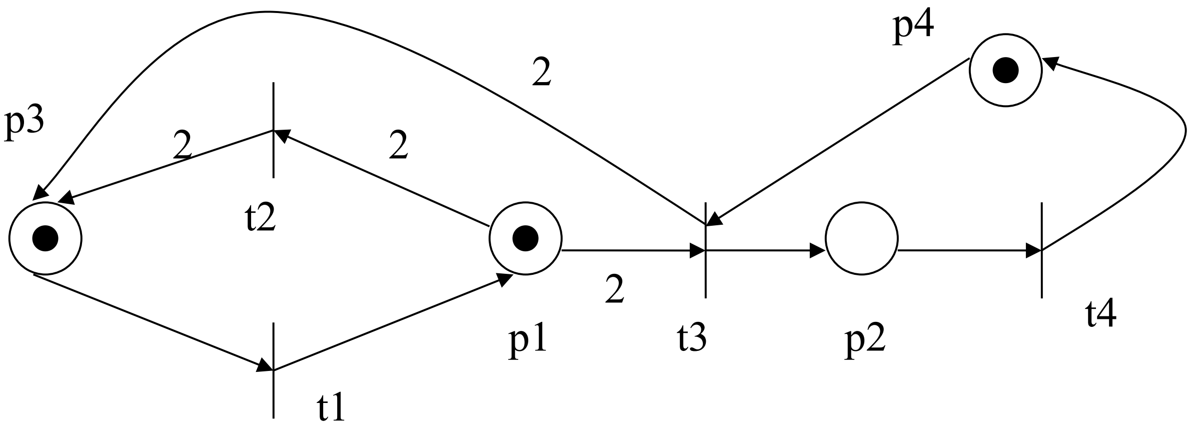
- Definition
  Transitions t1, t2 are in
  a *bounded fairness relation* if the maximal number of times that either one can fire while the other is not firing is bounded.

- Petri net is a *bounded fair net* if every possible pair of transitions is in bounded fairness relation.

- Are the pairs of transitions:
  t1, t2
  t3,t4
  in bounded fairness relation?



Properties of Petri nets (Behavioural properties)



- The pair t1, t2 is not in a bounded fairness relation (see σ=(t1 t3 t1 t4)*).

$$d_{12} = \infty$$

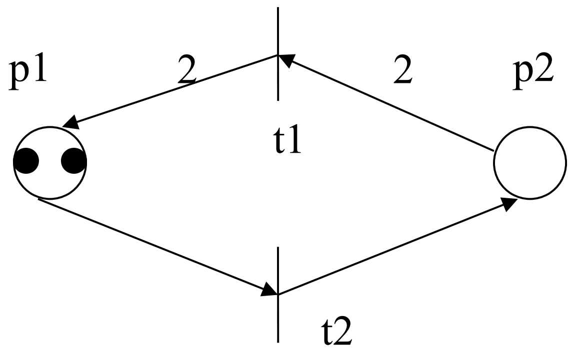
- The pair t3,t4 is in a bounded fairness relation.

$$d_{34} = 1$$

Properties of Petri nets (Behavioural properties)

- Is it true that:
  A pair of transitions t1, t2 is in a bounded fairness relation iff $d_{12} < \infty$?

- Example



$d_{12} = \infty$ because of σ=(t2 t2 t1)* .

t1, t2 are in a bounded fairness relation.

Information Systems Analysis

Jan Magott

Petri nets with time factor

Choose yourself and new techn

Alternatives of introducing a time factor into Petri nets

Nature of time specification:

- Deterministic,
- Non-deterministic,
- Probabilistic.

Petri net elements that the time factor is assigned to:

- Place,
- Transition,
- Arc.

Models for applications

Timed and time Petri nets for verification whether the functional requirements with quantitative time are satisfied by the design of the system.

Stochastic and generalized stochastic Petri nets for performance and reliability evaluation and engineering of systems.

Time Petri nets with time interval assigned to transition

Timed Petri nets is 6-tuple:

P – a set of places,

T – a set of transitions,

$F \subseteq (P \times T) \cup (T \times P)$ – a set of arcs,

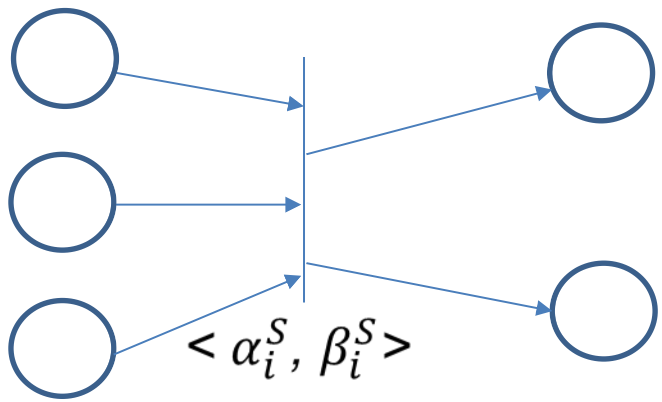
$I \subseteq P \times T$ – a set of inhibitor arcs,

$M_0 : P \rightarrow \{0,1,2,\ldots\}$ – an initial marking function,

$SI : T \rightarrow Q_+ \times (Q_+ \cup \{\infty\})$ – static firing time interval, where Q_+ is the set of non-negative rational numbers,

and $\alpha_i^S \in Q_+$, $\beta_i^S \in (Q_+ \cup \{\infty\})$, respectively, are the earliest and the latest firing times of transition t_i.

Time Petri nets with time interval assigned to transition



τ - time instant when transition t_i became enabled,
transition t_i can be fired not earlier than in time instant $\tau + \alpha_i^S$ and not later than in $\tau + \beta_i^S$

Time Petri nets with time interval assigned to transition

States of the net are represented by:

$$S = < M, I >$$

where

M is the marking of the net,

I – dynamic firing time intervals of transitions.

For each enabled transition t_i, its dynamic firing time interval is assigned:

$DFTI(t_i) = < DETI(t_i), DLTI(t_i) > \in Q_+ \times (Q_+ \cup \{\infty\})$,

$DETI(t_i)$, $DLTI(t_i)$, respectively, are the earliest and the latest firing time instants of transition t_i relative to present time instant.

For Petri nets without time factor, states of the net are expressed by markings.

For time Petri nets, state class can represent infinite number of states. The class is defined by system of linear inequalities with at most two variables per inequality.

Time Petri nets with time interval assigned to transition

Mathematical tool:

System of linear inequalities with at most two variables per inequality.

Software tool:

TINA (Time Petri Net Analyzer) LAAS CNRS

Time Petri nets with time interval assigned to arcs

Timed Petri nets is 6-tuple:

P – a set of places,

T – a set of transitions,

$F \subseteq (P \times T) \cup (T \times P)$ – a set of arcs,

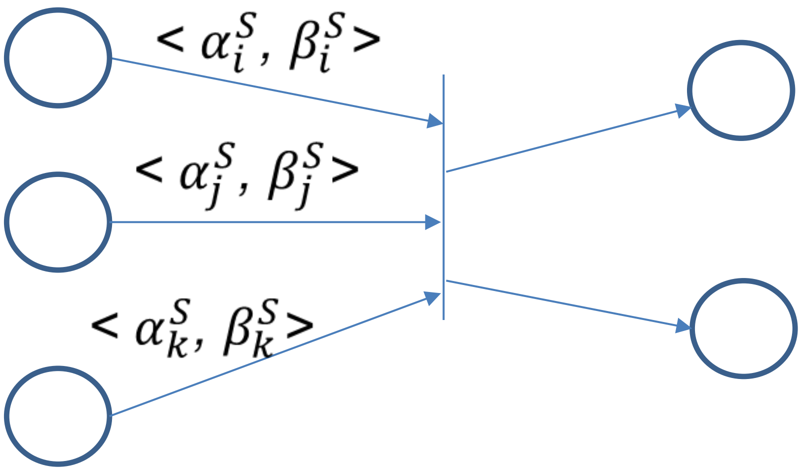
$I \subseteq P \times T$ – a set of inhibitor arcs,

$M_0 : P \rightarrow \{0,1,2,\ldots\}$ – an initial marking function,

$SI : F \times (P \times T) \rightarrow Q_+ \times (Q_+ \cup \{\infty\})$ – static enabling time interval, where Q_+ is the set of non-negative rational numbers,

and $\alpha_i^S \in Q_+$, $\beta_i^S \in (Q_+ \cup \{\infty\})$, respectively, are the earliest and the latest enabling time instants when the token in input place of transition t_i can enable this transition.

Time Petri nets with time interval assigned to arcs



Information Systems Analysis

Jan Magott

Behavioural properties of Petri nets (continued)

Properties of Petri nets

- Definition

A *Petri net* is a 6-tuple:

$$N = < P, T, F, W, M_0 >$$

P – a set of places,

T – a set of transitions,

$F \subseteq (P \times T) \cup (T \times P)$ – a set of arcs,

$W : F \to \{1, 2, \ldots\}$ – an arc weight function,

$M_0 : P \to \{0, 1, 2, \ldots\}$ – an initial marking function.

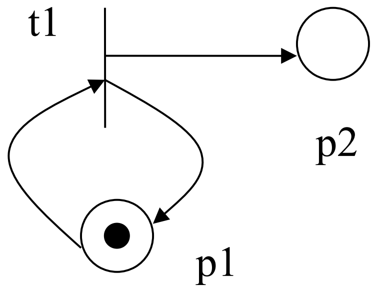
Properties of Petri nets (Behavioural properties)

- The reachability graph (b) for the net (a) is not finite.
- A problem is decidable if there is an algorithm with finite number of steps.
- Symbol „ω" in *coverability tree* (c) represents the fact that an unbounded number of tokens can be contained in a place.

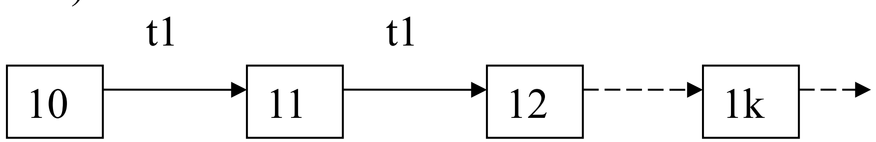
$$(\forall n \in N)(n < \omega \wedge \omega + n = \omega \wedge \omega - n = \omega)$$

- In place p2 an infinite number of tokens might be contained.

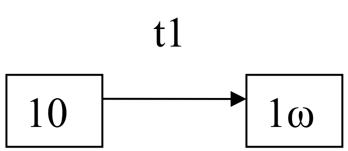
a)



b)

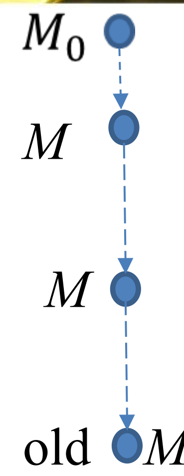


c)



Properties of Petri nets (Behavioural properties)

A coverability tree construction algorithm



1. Label the root of the tree as initial marking and symbol „new”.
2. Until there are „new” markings, execute:

   2.1 Choose „new” marking M

   2.2 If M is identical with a marking on the path from the root to the M, then label M as „old”.

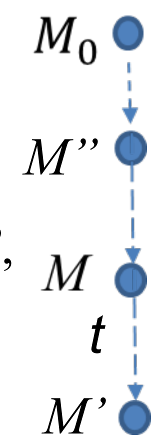
   2.3 If there are no transitions that are enabled in marking M, then label M as „dead”.

   2.4 Until there are enabled transitions in M, for each of them execute:

      2.4.1 Compute marking M' obtained after firing transition t

      2.4.2 If there is a marking M'' on the path from the root to the M that $(\forall p \in P)(M''(p) \le M'(p)) \wedge (M'' \ne M')$, i.e., M'' is covered by M', then for each p that $M''(p) < M'(p)$ replace $M'(p)$ by ω.

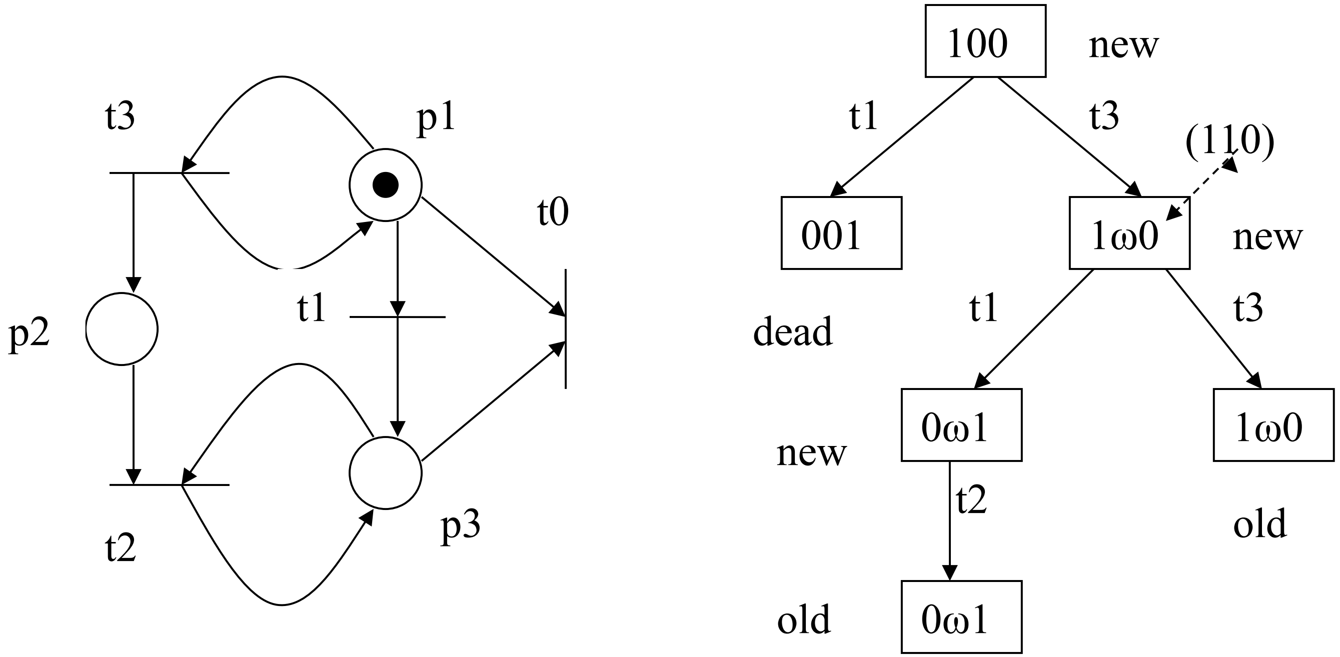
      2.4.3 Add M' as „new”, and draw the arc directed from vertex with M to vertex M', and label the arc as transition t.



Theorem

Coverability tree is finite.

Properties of Petri nets (Behavioural properties)



A Petri net and its coverability tree

Properties of Petri nets (Behavioural properties)

Properties that can be verified using a coverability tree

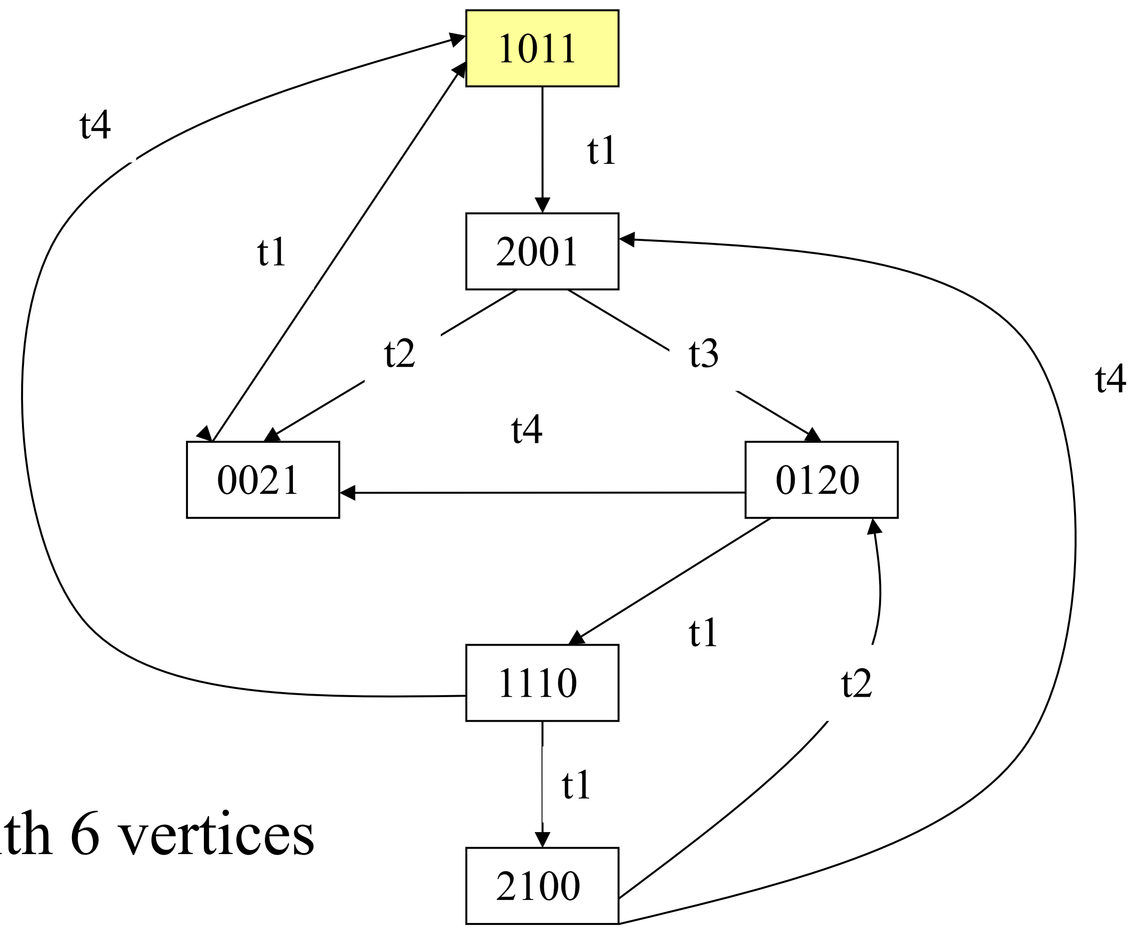
- A Petri net *N* is bounded ($R(M_0)$ is finite) iff there are no symbol ω in the tree.
- For a bounded Petri net, the reachability problem can be solved because each $M \in R(M_0)$ occurs in the tree.
- If $M \in R(M_0)$ then there exists a vertex with label *M'* in the tree that $M \leq M'$.
- A Petri net *N* is safe iff „0" and „1" only occur in rectangles of the tree.
- A transition *t* can never be fired iff *t* does not occur as label of arc in the tree.

Properties of Petri nets (Behavioural properties)

Properties that cannot be verified using coverability tree

- If a Petri net is not bouded then coverability tree (with symbol „ω”) is sufficient to solve neither reachability nor liveness problem.

Properties of Petri nets (Behavioural properties)

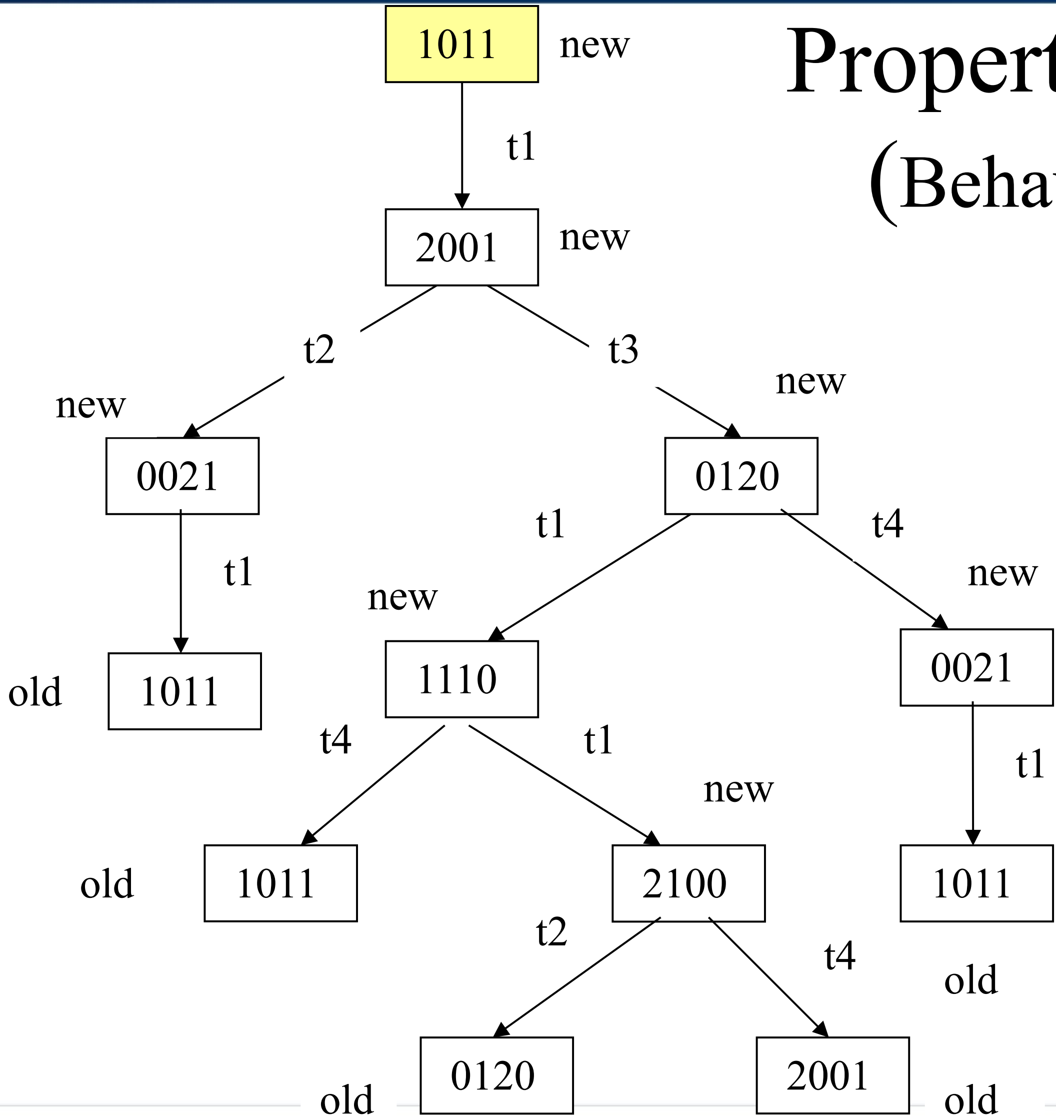


A rechability graph with 6 vertices

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Properties of Petri nets

(Behavioural properties)



A coverability tree with 12 vertices

← Why this vertex is not the „old" one?

Properties of Petri nets (Behavioural properties)

- A Petri net $N = \langle P, T, F, W, C, M_0 \rangle$ is bounded if

  $$(\exists k \in \{1,2,\dots\})(\forall M \in R(M_0))(\forall p \in P)(M(p) \le k)$$

- Is the following definition:

  $$(\forall M \in R(M_0))(\exists k \in \{1,2,\dots\})(\forall p \in P)(M(p) \le k)$$

  equivalent to the above?

- A Petri net is *k*-bounded if

  $$(\forall M \in R(M_0))(\forall p \in P)(M(p) \le k)$$

- A Petri net is safe if it is 1-bounded.

- Practical aspect. Problem: *Is Petri net safe?* can be used in verification problem: *Can buffer of capacity equal to* 1 *be overflowed*?

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Properties of Petri nets (Behavioural properties)

- Definition

  A *transition t* is *live* iff

  $$(\forall M \in R(M_0))(\exists M' \in R(M))(M' \xrightarrow{t})$$

  where $M' \xrightarrow{t}$ - transition *t* can be fired for marking *M'*.

- Definition

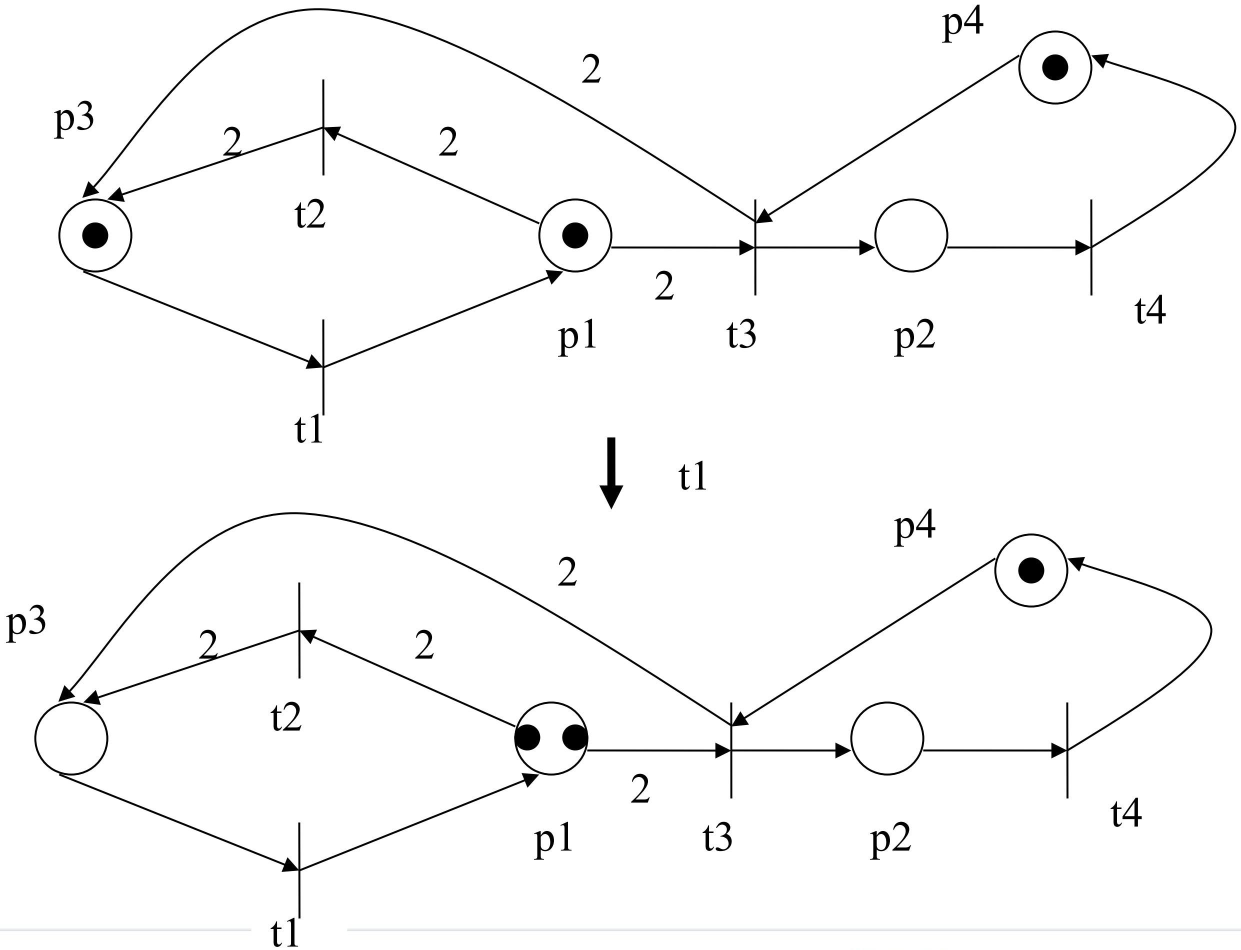
  A *Petri net* is *live* if each of its transitions are live.

- Liveness means that there are no deadlocks in the net.

Properties of Petri nets (Behavioural properties)

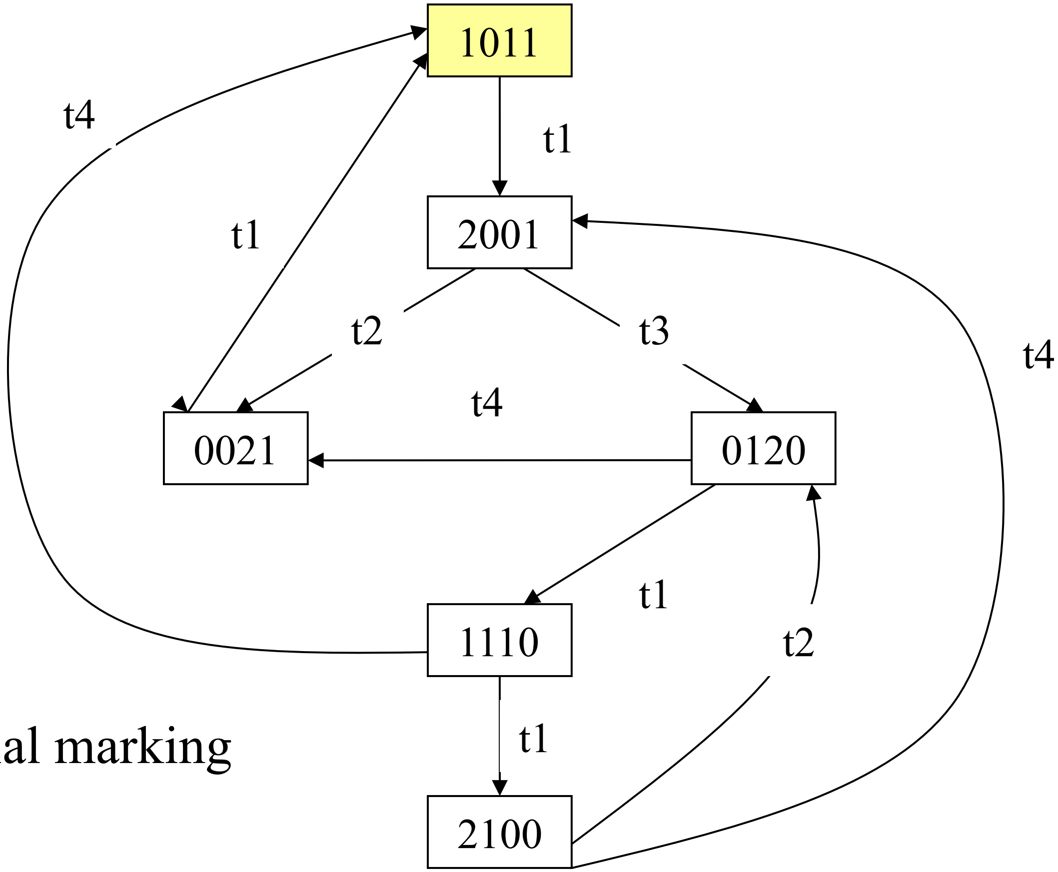
$(\forall p \in P)(C(p) = \infty)$

i.e., the place capacity function does not impose limitations on firing of transitions.



Properties of Petri nets (Behavioural properties)

Reachability graph



- 1011 – initial marking

Properties of Petri nets (Behavioural properties)

Decision power of reachability graph when liveness problem has to be solved

- If for every vertex of the finite reachability graph there exists a path directed from this vertex which contains arc labelled as *t*, then transition *t* is live.
- If finite reachability graphs are strongly connected then transition *t* is live iff there exists an arc labelled as this transition.

Properties of Petri nets (Behavioural properties)

- Definition
  The *Reachability problem*
  Input: Petri net $N = \langle P, T, F, W, M_0 \rangle$
  Output: $M \in R(M_0)$?

- Theorem (1981)
  The Reachability problem is decidable.
  Computational complexity of an algorithm is exponential.

Properties of Petri nets (Behavioural properties)

- Definition
  The *Liveness problem*
  Input: Petri net N
  Output: Is N live?

- Theorem (1975)
  The Liveness problem is equivalent to the reachability problem.

- Conclusion
  Computational complexity of Liveness problem is exponential.

Information Systems Analysis

Jan Magott

Behavioural properties of Petri nets
The matrix equation method

Choose yourself and new tech

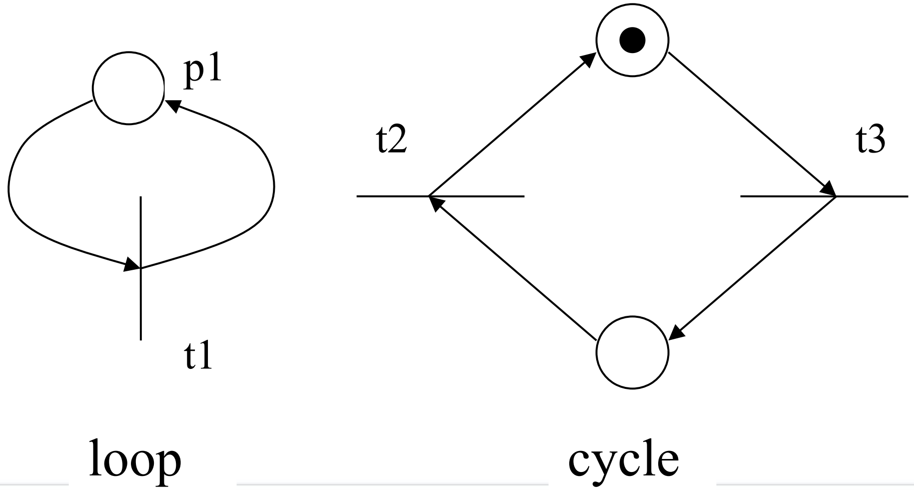
Properties of Petri nets (Behavioural properties)
The matrix equation method

- Definition

  A Petri net is pure if

$$\neg(\exists p \in P)(\exists t \in T)(p \in {}^{\bullet}t \wedge t \in {}^{\bullet}p)$$

- A graph of a pure Petri net does not contain loops.



Properties of Petri nets (Behavioural properties)

The matrix equation method

- For a pure Petri net with $|P|=n$ and $|T|=m$, incidence matrix C is defined as follows:

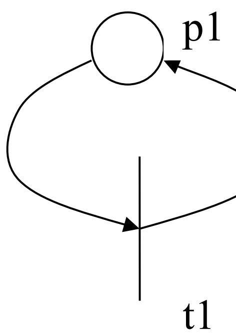
$$C = [c_{ij}]_{n \times m}$$

$$c_{ij} = \begin{cases} -\mathrm{W}(<p_i, t_j>) & \text{if } (<p_i, t_j>) \in F \\ \mathrm{W}(<t_j, p_i>) & \text{if } (<t_j, p_i>) \in F \\ 0 & \text{if otherwise} \end{cases}$$

Properties of Petri nets (Behavioural properties)

The matrix equation method

- Why a pure Petri net cannot be represented by any incidence matrix?



$$c_{11} = \begin{cases} -1 & \text{because} \quad (\langle p_1, t_1 \rangle) \in F \\ 1 & \text{because} \quad (\langle t_1, p_1 \rangle) \in F \end{cases}$$

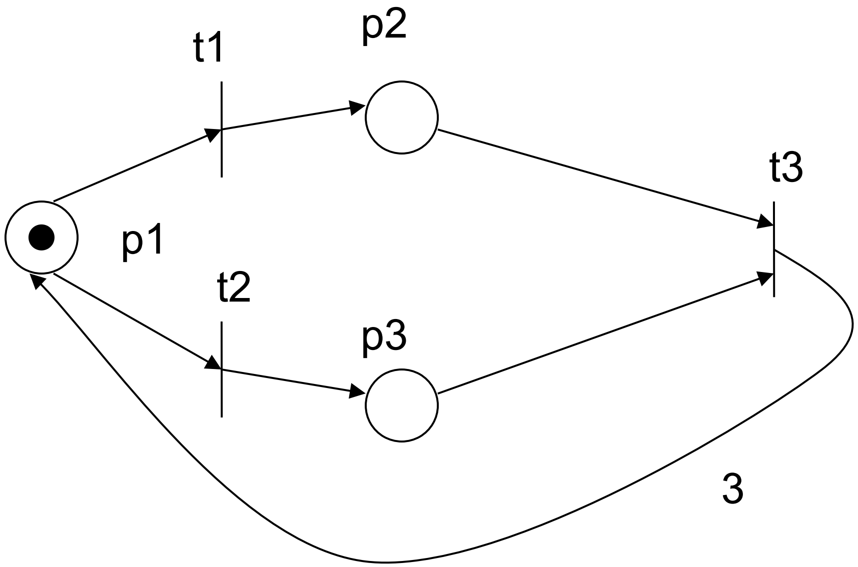
This is a contradiction.

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Properties of Petri nets (Behavioural properties)

The matrix equation method

- Example



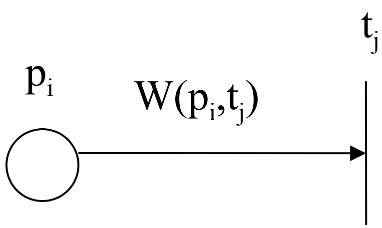
$$C = \begin{bmatrix} -1 & -1 & 3 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$$

Properties of Petri nets (Behavioural properties)

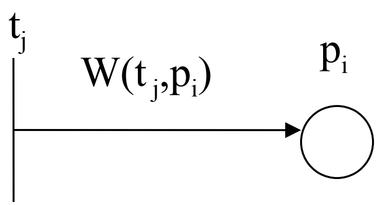
The matrix equation method

- *j*-th column of matrix C expresses influence of transition t_j firing

$$\begin{matrix} c_{1j} \\ \cdot \\ \cdot \\ c_{ij} \\ \cdot \\ \cdot \\ c_{nj} \end{matrix}$$



← Firing of the transition decreases the number of tokens in the place by $W(p_i,t_j)$



← Firing of the transition increases the number of tokens in the place by $W(t_j,p_i)$

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Properties of Petri nets (Behavioural properties)

The matrix equation method

- Let

  M_0 - an initial marking

  σ - a firing sequence that starts in marking M_0

  $\overline{\sigma}$ - a firing vector of a sequence σ which contains *m*=|*T*| entries and i-th entry $\overline{\sigma(t_i)}$ is equal the occurrence number of the transition t_i in the sequence σ

$$\overline{\sigma}_{|T|\times 1}$$

$$C \cdot \overline{\sigma} = \begin{bmatrix} c_{11} & \cdot & c_{1j} & \cdot & c_{1m} \\ \cdot & & \cdot & & \cdot \\ c_{i1} & \cdot & c_{ij} & \cdot & c_{im} \\ \cdot & & \cdot & & \cdot \\ c_{n1} & \cdot & c_{nj} & \cdot & c_{nm} \end{bmatrix} \bullet \begin{bmatrix} \overline{\sigma(t_1)} \\ \cdot \\ \overline{\sigma(t_j)} \\ \cdot \\ \overline{\sigma(t_m)} \end{bmatrix}$$

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Properties of Petri nets (Behavioural properties)

The matrix equation method

- A token number change in place pi as a result of $\overline{\sigma(t_j)}$ firings of transition tj

$$c_{ij} \cdot \overline{\sigma(t_j)}$$

- A token number change in place pi as a result of firing sequence σ

  C_i - i-th row of matrix C

$$C_i \cdot \overline{\sigma} = \begin{bmatrix} c_{i1} & . & c_{ij} & . & c_{im} \end{bmatrix} \bullet \begin{bmatrix} \overline{\sigma(t_1)} \\ . \\ \overline{\sigma(t_j)} \\ . \\ \overline{\sigma(t_m)} \end{bmatrix}$$

Properties of Petri nets (Behavioural properties)

The matrix equation method

- A matrix representation of marking M

$$M_{|P|\times 1}$$

- If $M_0 \xrightarrow{\sigma} M$ then $M = M_0 + C\,\overline{\sigma}$

- Hence, $(\forall M \in R(M_0))(\exists X \in \{0,1,2,\ldots\}^{|P|})(M = M_0 + C \cdot X)$

or $(\forall M \in R(M_0))(\exists X \in \{0,1,2,\ldots\}^{|P|})(C \cdot X = M - M_0)$

Properties of Petri nets (Behavioural properties)
The matrix equation method

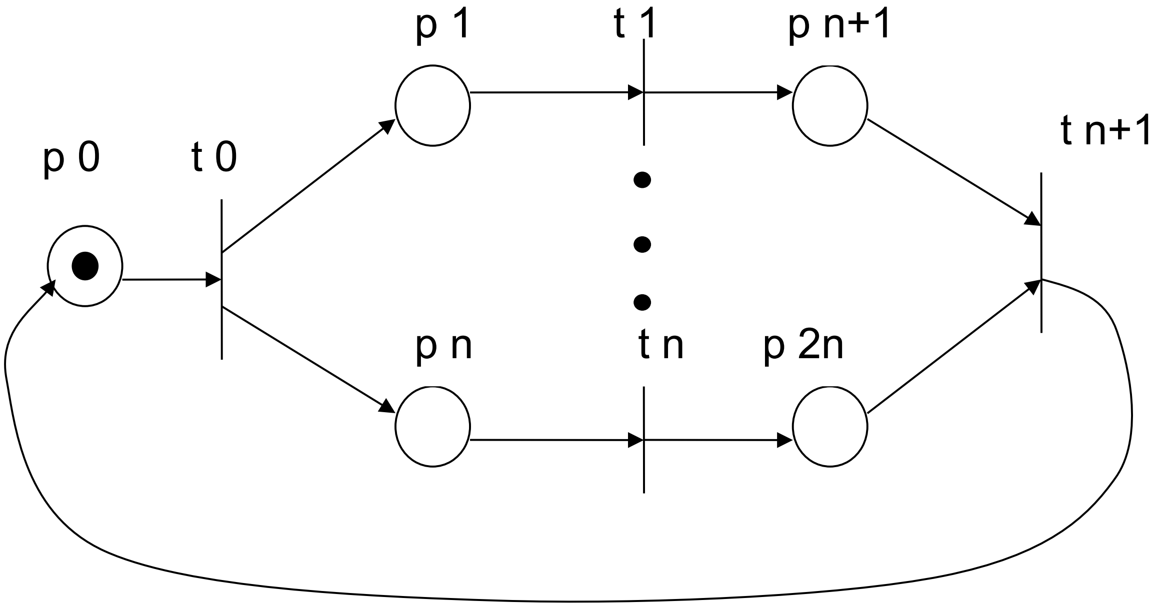
- Linear algebra methods are more efficient computationally than combinatorial enumeration of reachability graphs.

- Computational complexity of an algorithm solving linear equations $A_{m\times n}\cdot X_n = b_{m\times 1}$ is $O(n^{2.71})$.

- Computational complexity of an algorithm solving linear equations $A_{m\times n}\cdot X_n = b_{m\times 1}$ with integer entries is ???

Properties of Petri nets (Behavioural properties)

The matrix equation method



A number of reachable markings

$$2^n + 1$$

Conclusion

The number of reachable markings can be an exponential function of a Petri net size.

Properties of Petri nets (Behavioural properties)

The matrix equation method

- Problem

  Does the condition

$$(\exists X \in \{0,1,2,\ldots\}^{|P|})(C \cdot X = M - M_0)$$

is necessary and sufficient condition that $M \in R(M_0)$?

It is the necessary condition indeed because

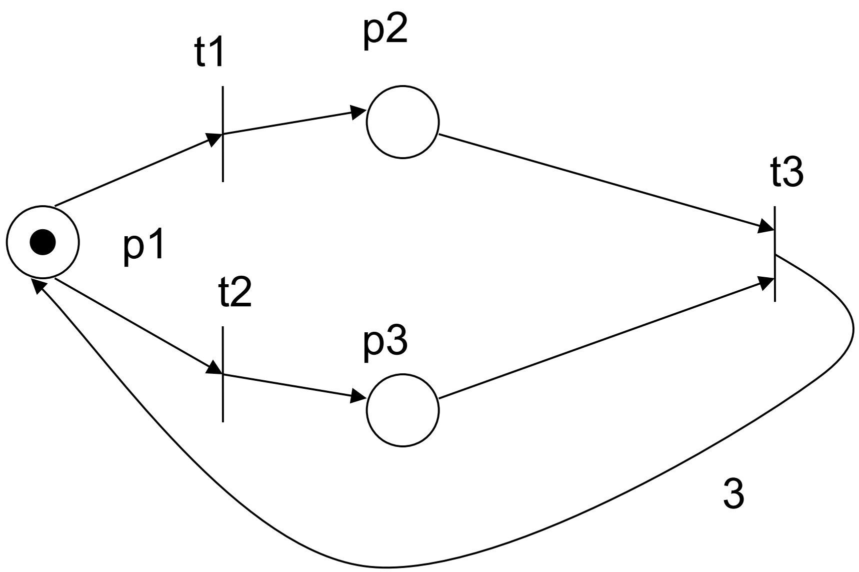
$$M \in R(M_0) \Rightarrow (\exists \sigma \in T^*)(M_0 \xrightarrow{\sigma} M)\} \Rightarrow M = M_0 + C \cdot \overline{\sigma} \Rightarrow X = \overline{\sigma}$$

However, it is not the sufficient condition.

Properties of Petri nets (Behavioural properties)

The matrix equation method

- Example (that it is not the sufficient condition)



$$C = \begin{bmatrix} -1 & -1 & 3 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$$

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad M = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

Equation:

$$M = M_0 + C \cdot X$$

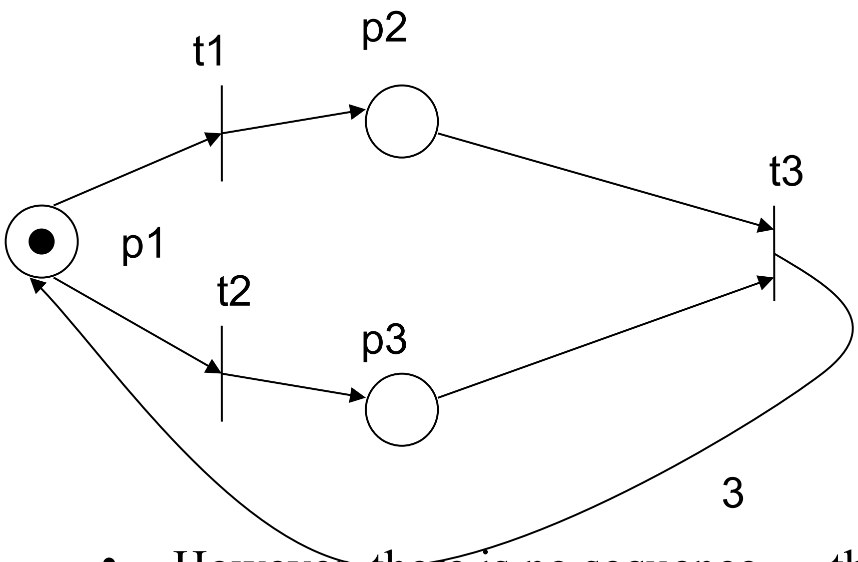
Solution:

$$X = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Properties of Petri nets (Behavioural properties)

The matrix equation method

- Example (that it is not sufficient condition)



$$C = \begin{bmatrix} -1 & -1 & 3 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \qquad M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad M = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

$$M_0 + C \cdot X = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & -1 & 3 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} = M$$

- However, there is no sequence σ that

$$M_0 \xrightarrow{\sigma} M \wedge \overline{\sigma(t1)} = 1 \wedge \overline{\sigma(t2)} = 1 \wedge \overline{\sigma(t3)} = 1$$

Information Systems Analysis

Jan Magott

Behavioural properties of Petri nets
The net reduction method

Properties of Petri nets (Behavioural properties)

The net reduction method

- In order to reduce computational complexity of algorithms for Petri net analysis problems, reductions are used.

- Reductions that preserve liveness, boundness, and safety of Petri nets are such that:

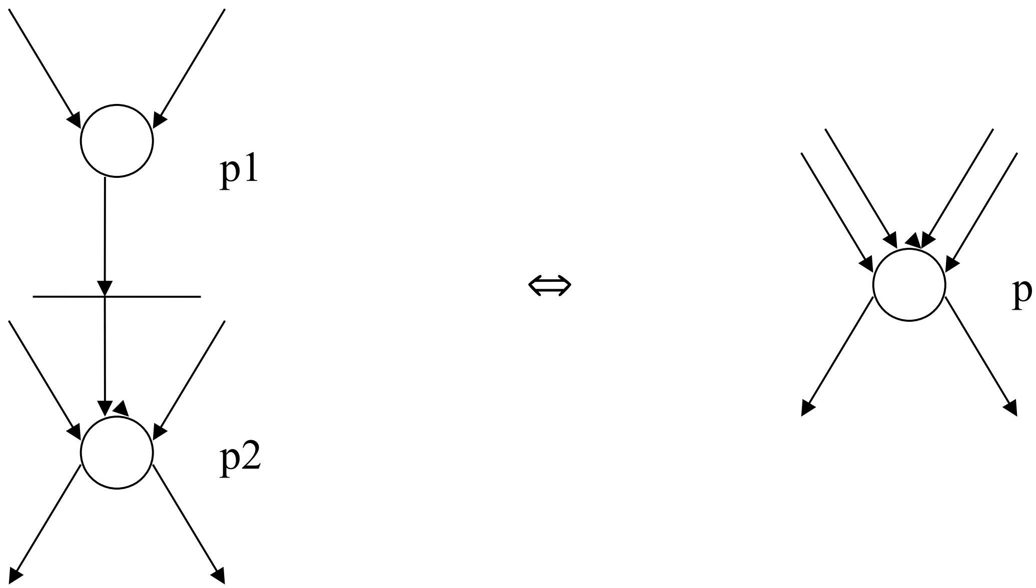
  Let *N* and *N'*, respectively, be nets before and after reduction.

  Net *N'* is live, bounded, and safe iff net *N* is live, bounded, and safe.

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Properties of Petri nets (Behavioural properties)
The net reduction method

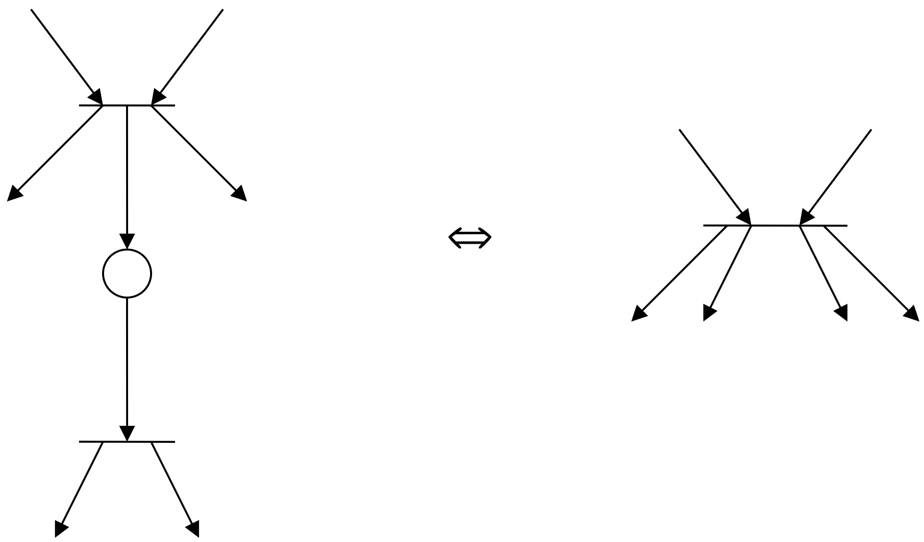
- Assumption: $M(p)=M(p1)+M(p2)$



A sequential connection of places

Properties of Petri nets (Behavioural properties)

The net reduction method

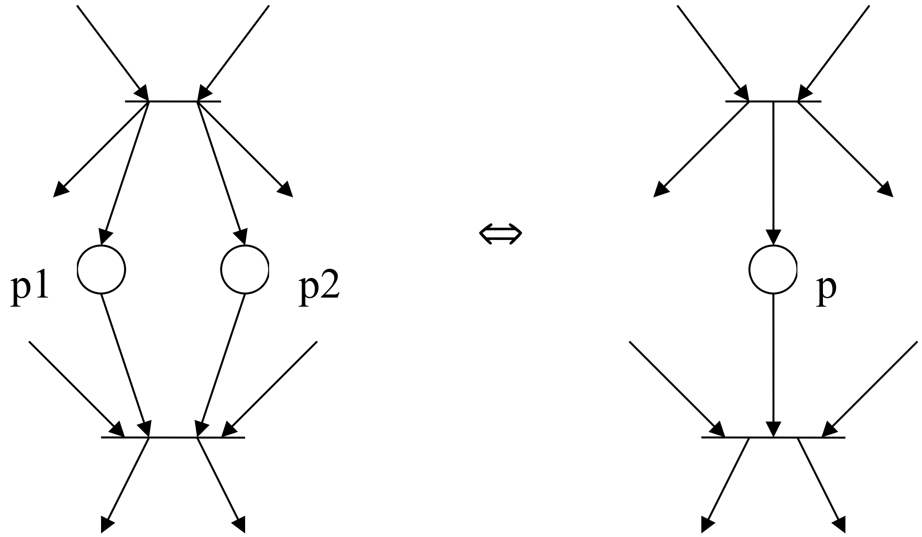


A sequential connection of transitions

Properties of Petri nets (Behavioural properties)

The net reduction method

- Assumption: $M(p1)=M(p2)=M(p)$

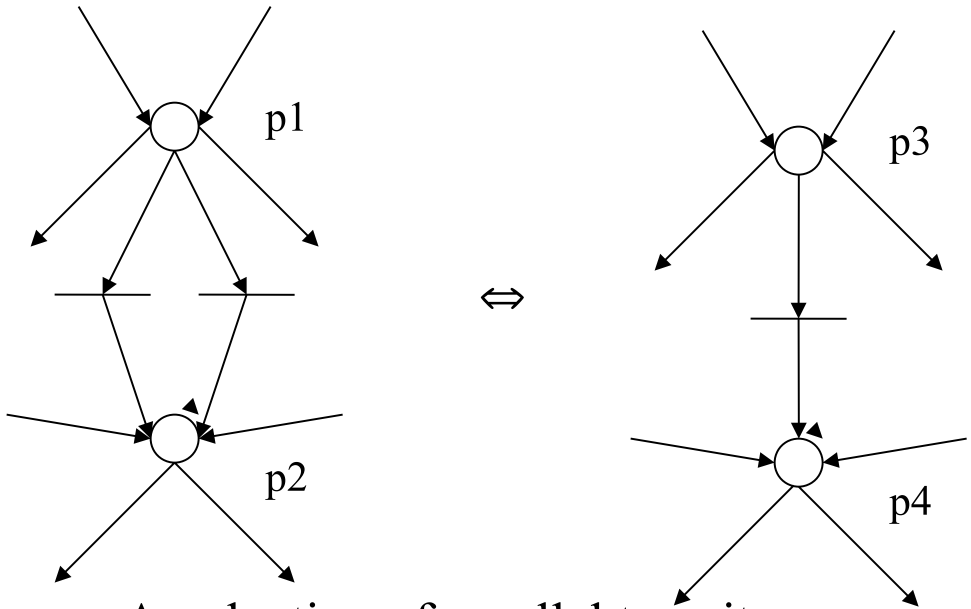


A parallel connection of places

Properties of Petri nets (Behavioural properties)

The net reduction method

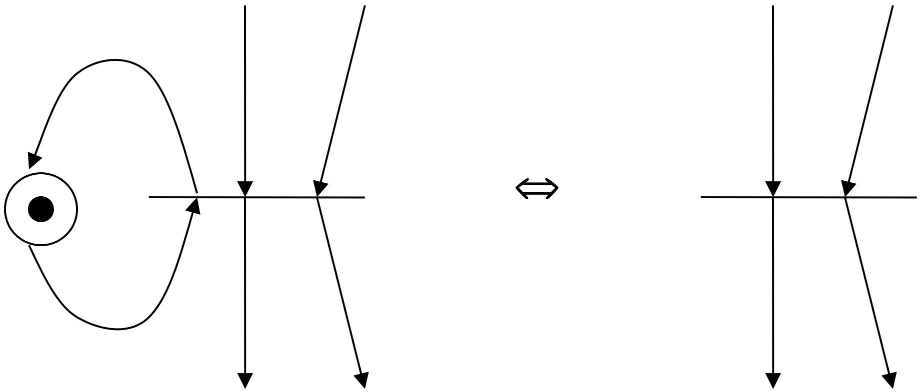
- Assumption: $M(p1)=M(p3)$, $M(p2)=M(p4)$



A reduction of parallel transitons

Properties of Petri nets (Behavioural properties)

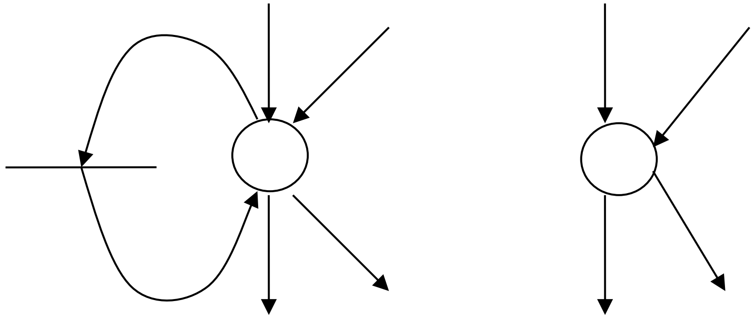
The net reduction method



A loop-place elimination

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Properties of Petri nets (Behavioural properties)
The net reduction method



A loop-transition elimination

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Information Systems Analysis

Jan Magott

Performance evaluation of systems
Introduction

Choose yourself and new tech

Performance evaluation of systems

- Performance evaluation can be done at the following *abstraction levels*:
  hardware configuration,
  program,
  computer system.

- Performance measures for hardware configuration level:
  memory cycle time,
  disc access time,
  instruction type execution time,
  chanel transmission throughput.

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Performance evaluation of systems

- Performance measures for program level:
  program mean execution time,
  mean transition time between two selected points of program,
  maximal number of executions of a loop in real time system program.

- Performance measures for computer system level:
  mean response time for system with terminals,
  mean access time to data in data base,
  mean packet transmission time between pair of selected network nodes.

Performance evaluation of systems

- Approaches in computer systems performance evaluation:
  intuition and trends extrapolation,
  experimental evaluation of alternative solutions,
  modelling.

In experimental evaluation, measurements are executed using hardware and software monitors.

Model based estimations can be obtained using simulation or analysis.

Performance evaluation of systems

- Measurements of … provided …:
  Real system in real workload,
  Real system in artificial workload,
  Prototype system in real workload,
  Prototype system in artificial workload.

- Modelling using:
  Simulation models,
  Analytic models.

Performance evaluation of systems

- Modelling

  Aspects that are important from modelling goal should be expressed.

  Abstraction level (How many details are represented?)

  Simulation models are usually more detailed and more adequate than analytic ones.

  Simulation models are usually solved in a simpler way.

  Computation time of analytic models is usually shorter than of simulation models.

Information Systems Analysis

Jan Magott

Performance evaluation of systems
Programs with sequential control structures

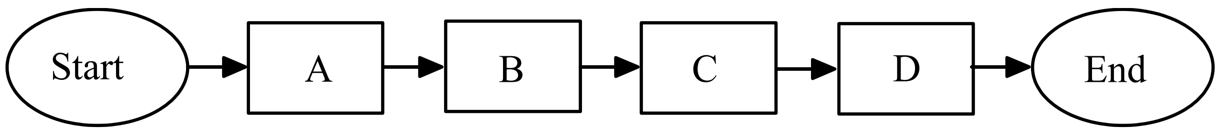
Choose yourself and new tech

$\tau(\mathrm{Pr}) = \tau(A) + \tau(B) + \tau(C) + \tau(D)$

Performance evaluation of systems

Programs with sequential control structures

- Assumption
  Execution times of operations, actions or statements are given by a real number.



Linear program Pr where *A, B, C, D* are operations, actions or statements

$\boldsymbol{\tau}(A) \in R_+$ - execution time of *A*

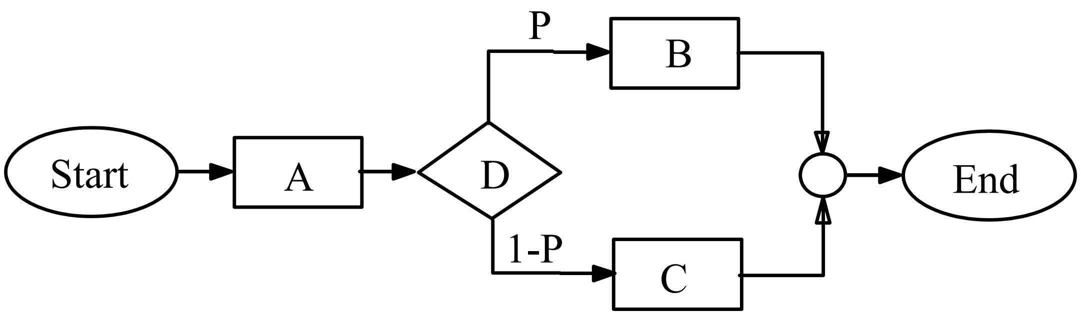
- Execution time of the program Pr

$$\tau(\mathrm{Pr}) = \tau(A) + \tau(B) + \tau(C) + \tau(D)$$

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Performance evaluation of systems

Programs with sequential control structures



A program with decision

$\tau(D)$ – execution time of decision D

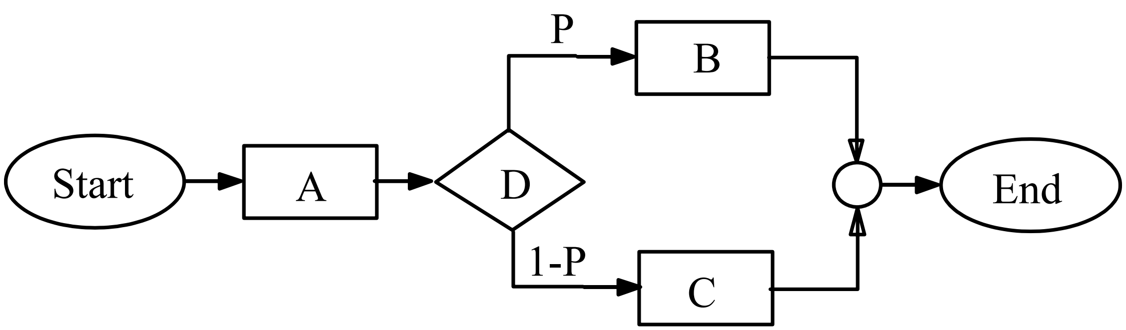
P – probability of an event that after the decision D, the B action is executed

If $\tau(B) \neq \tau(C)$ then execution time of the program depends on the decision.

Performance evaluation of systems

Programs with sequential control structures

- Execution time estimation methods



1. *The worst case method*

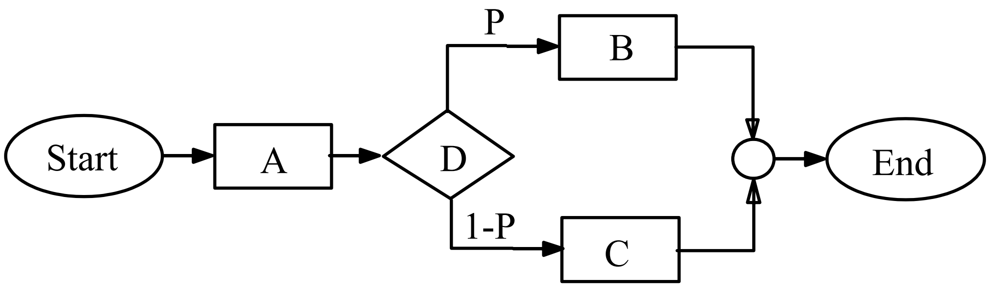
$$\tau^{w}(\mathrm{Pr}) = \tau(A) + \tau(D) + \max\{\tau(B), \tau(C)\}$$

$\tau^{w}(\mathrm{Pr})$ - execution time estimation of the worst case

Application: real-time systems.

Performance evaluation of systems
Programs with sequential control structures

2. *The most probable path method*



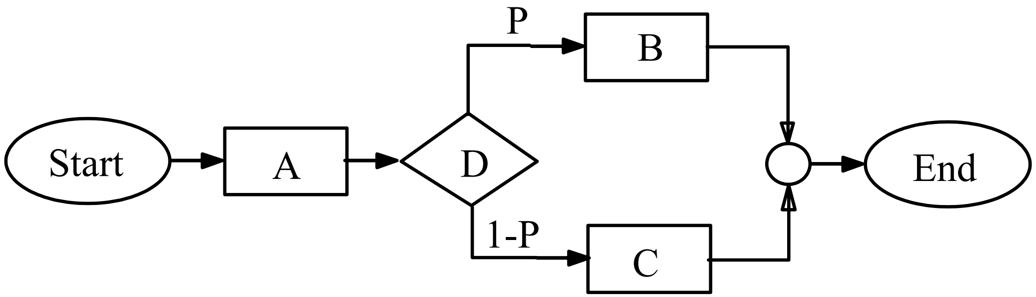
If P>1-P then $\tau^{g}(\Pr) = \tau(A) + \tau(D) + \tau(B)$

If P<1-P then $\tau^{g}(\Pr) = \tau(A) + \tau(D) + \tau(C)$

Performance evaluation of systems

Programs with sequential control structures

3. *The arithmetic mean method*



$$\tau^{a}(\mathrm{Pr}) = \tau(A) + \tau(D) + \frac{\tau(B) + \tau(C)}{2}$$

4. *The random variable mean method*

$$\tau^{m}(\mathrm{Pr}) = \tau(A) + \tau(D) + P \cdot \tau(B) + (1 - P) \cdot \tau(C)$$

Application: general purpose systems.

Performance evaluation of systems

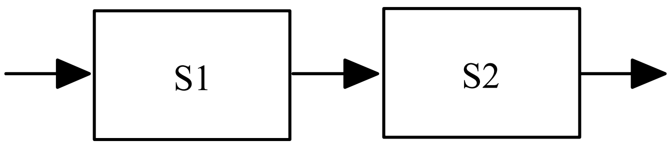
Programs with sequential control structures

- Execution time estimation for program control structures
- Notation:
  B – a Boolean expression,
  S, Si – operations, actions, statements,
  Ci – value of the same type as expression E.
- Assumptions
  Random variables of execution times of control structures components are independent.
  Execution times of decisions are equal to zero.

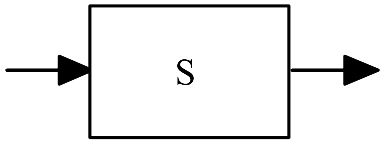
Performance evaluation of systems

Programs with sequential control structures

- *Sequential composition* {S1;S2}



μ_{S1}, μ_{S2} - mean value of random variables of exection times of S1, S2



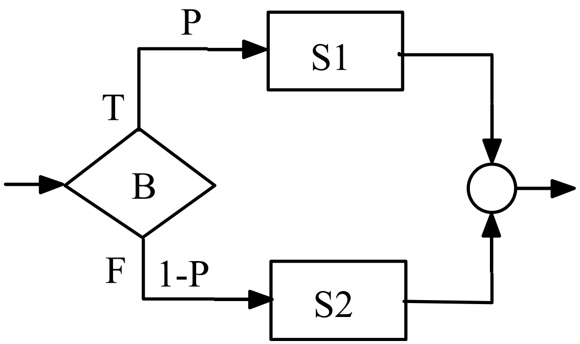
Result of sequential composition of S1 and S2

$\mu_S = \mu_{S1} + \mu_{S2}$

$\sigma_S^{\ 2} = \sigma^2{}_{S1} + \sigma^2{}_{S2}$

Performance evaluation of systems
Programs with sequential control structures

- *Alternative* if (B) S1 else S2



μ_S, σ_S^2 - mean value, variance of execution time of the alternative

$$\mu_S = P \cdot \mu_{S1} + (1 - P) \cdot \mu_{S2}$$

$$\sigma_S^2 = P \cdot \sigma_{S1}^2 + (1 - P) \cdot \sigma_{S2}^2 + P \cdot \mu_{S1}^2 + (1 - P) \cdot \mu_{S2}^2 - (P \cdot \mu_{S1} + (1 - P) \cdot \mu_{S2})^2$$

Performance evaluation of systems
Programs with sequential control structures

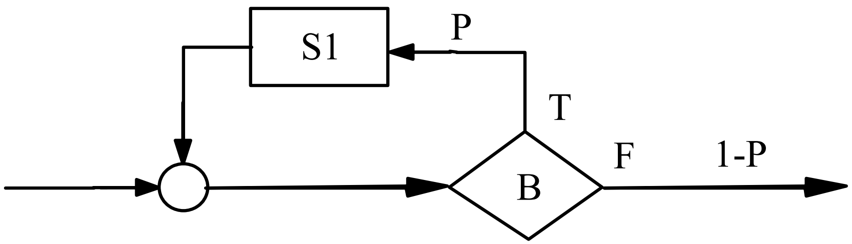
- *Iteration* for (i=0; i<n; i=i+1) S1

$$\mu_S = n \cdot \mu_{S1}$$

$$\sigma_S^2 = n \cdot \sigma_{S1}^2$$

Performance evaluation of systems
Programs with sequential control structures

- *Iteration* while (B) S1



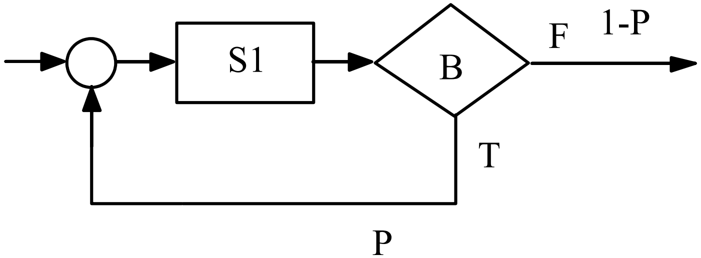
$$\mu_S = \frac{P}{1-P}\mu_{S1}$$

$$\sigma_S^2 = \frac{P}{(1-P)^2}\mu_{S1}^2 + \frac{P}{1-P}\sigma_{S1}^2$$

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Performance evaluation of systems
Programs with sequential control structures

Iteration do S1 while (B)



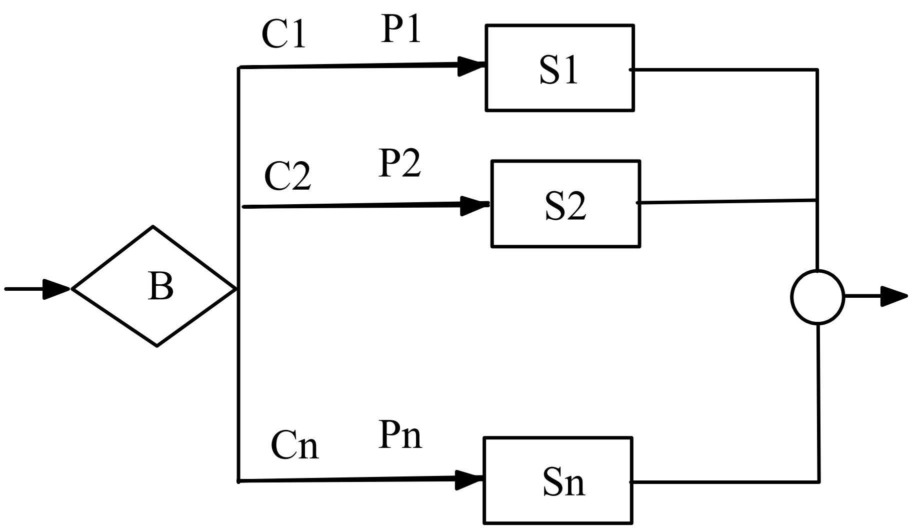
$$\mu_S = \frac{1}{1-P}\mu_{S1}$$

$$\sigma_S^2 = \frac{P}{(1-P)^2}\mu_{S1}^2 + \frac{1}{1-P}\sigma_{S1}^2$$

Performance evaluation of systems

Programs with sequential control structures

Choice switch (E) { case C1:S1; case C2:S2; ... , case Cn:Sn; }



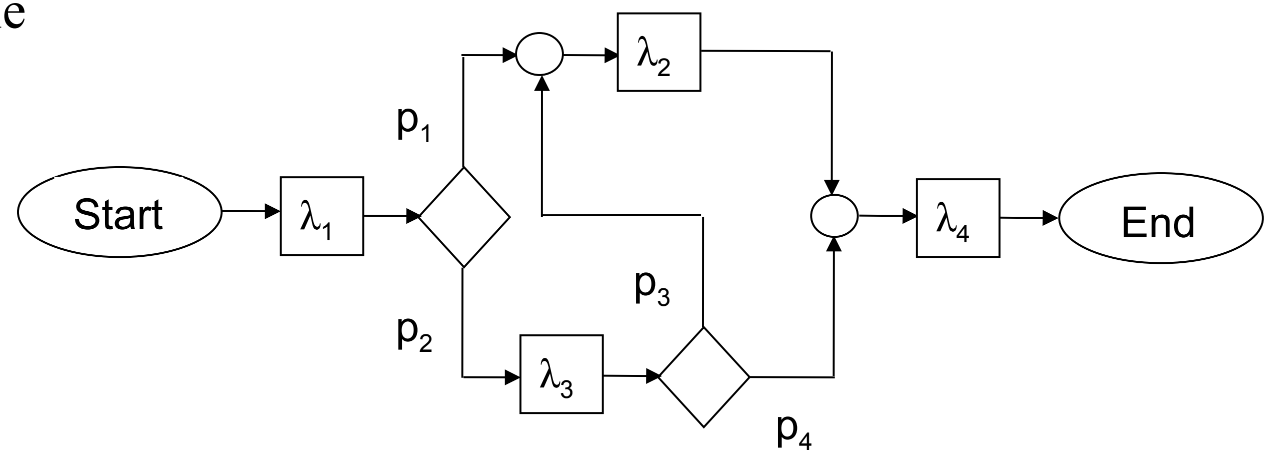
$$\mu_S = \sum_{i=1}^{n} Pi \cdot \mu_{Si}$$

$$\sigma_S^2 = \sum_{i=1}^{n} Pi \cdot \sigma_{Si}^2 + \sum_{i=1}^{n} Pi \cdot \mu_{Si}^2 - (\sum_{i=1}^{n} Pi \cdot \mu_{Si})^2$$

Performance evaluation of systems
Programs with sequential control structures

- *Non-structural program without loops*

Example



Performance evaluation of systems

Programs with sequential control structures

Non-structural program without loops (cont.)

- Assumptions:

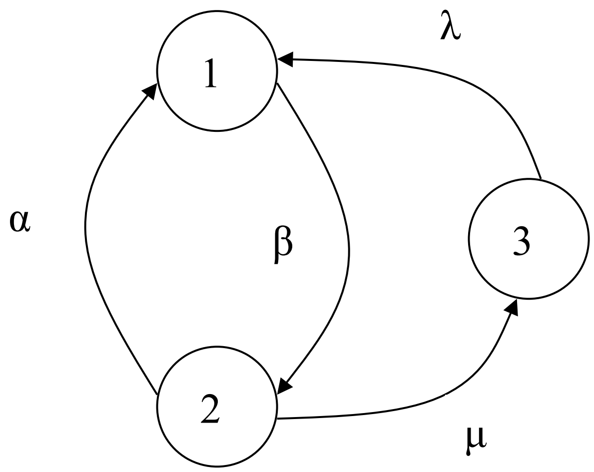
1. Decision execution time is equal to zero.

2. Random variables of operation executions times are independent.

3. Operation execution time of i-th operation is expressed by an exponential random variable with parameter λ_i

Performance evaluation of systems

Markov chains with continuous time



1, 2, 3 – Markov chain states,

α, β, λ, μ – transition intensities (parameters of exponential distributions of transition times between states)

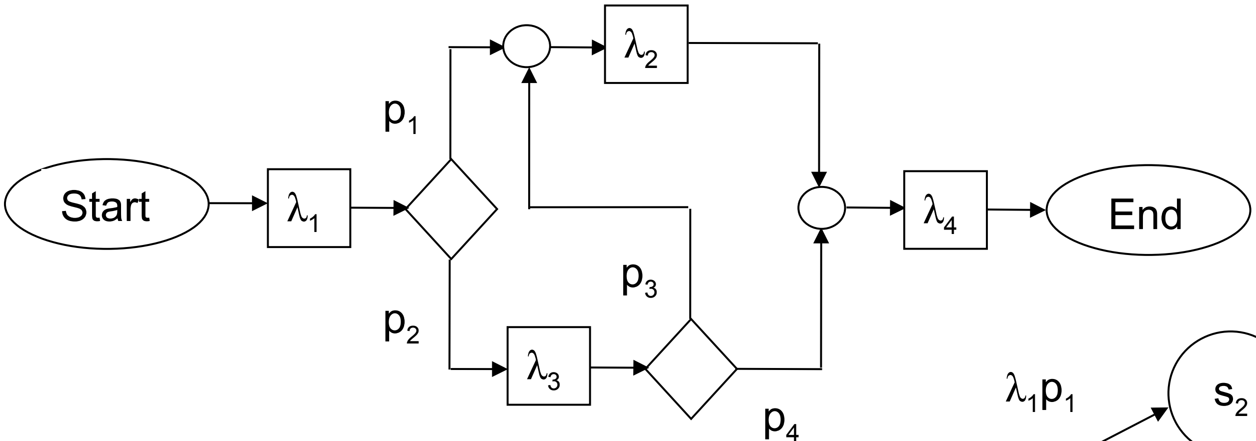
m_X - mean value of random variable X

$$m_X = \frac{1}{\lambda_X}$$

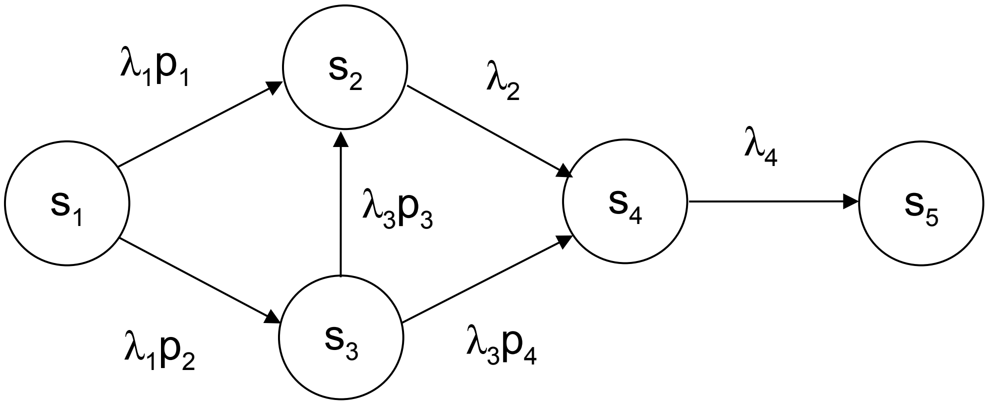
$$Q = \begin{bmatrix} -\beta & \beta & 0 \\ \alpha & -(\alpha+\mu) & \mu \\ \lambda & 0 & -\lambda \end{bmatrix}$$

Performance evaluation of systems
Programs with sequential control structures

Non-structural program without loops (cont.)

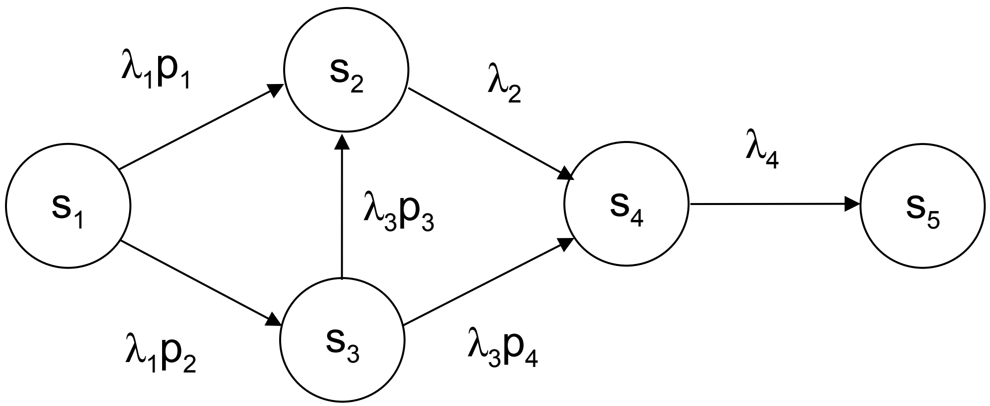


A Markov chain transition diagram for the program



Performance evaluation of systems
Programs with sequential control structures

Non-structural program without loops (cont.)



s_1, s_2, s_3, s_4 - transient states,

s_5 – the absorbing state (final state).

Performance evaluation of systems
Programs with sequential control structures

Non-structural program without loops (cont.)

Mean sojourn time in a transient state s_i of Markov chain:

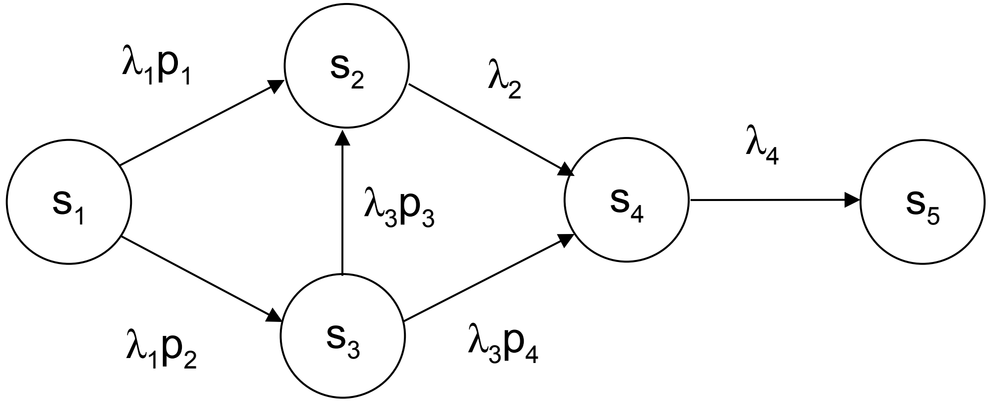
$$E(s_i) = \frac{1}{\sum\limits_{k \in O(s_i)} \lambda_{ik}}$$

$O(s_i)$ - a set of states that transitions from state s_i are directed to,

λ_{ik} - transition intensity from state s_i to state s_k .

Performance evaluation of systems
Programs with sequential control structures

Non-structural program without loops (cont.)



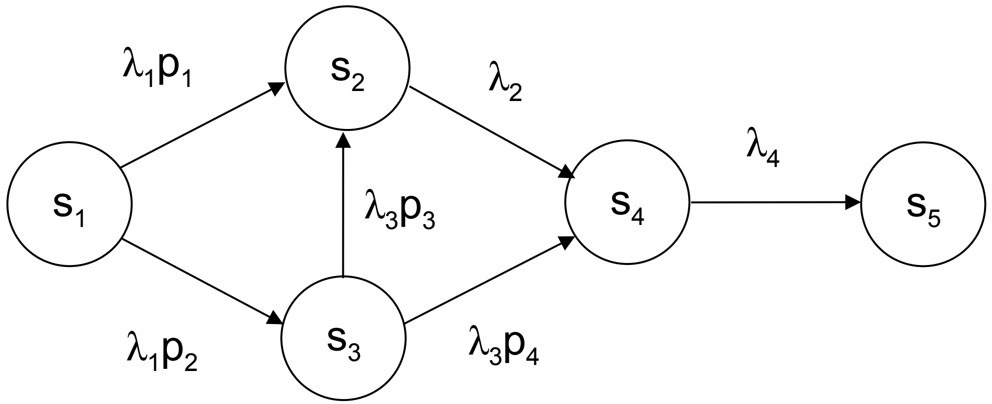
$$O(s_1) = \{s_2, s_3\}$$
$$O(s_2) = \{s_4\}$$
$$O(s_3) = \{s_2, s_4\}$$
$$O(s_4) = \{s_5\}.$$

$$E(s_1) = \frac{1}{\lambda_1 \cdot p_1 + \lambda_1 \cdot p_2} = \frac{1}{\lambda_1} \quad E(s_2) = \frac{1}{\lambda_2} \quad E(s_3) = \frac{1}{\lambda_3 \cdot p_3 + \lambda_3 \cdot p_4} = \frac{1}{\lambda_3} \quad E(s_4) = \frac{1}{\lambda_4}.$$

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Performance evaluation of systems
Programs with sequential control structures

Non-structural program without loops (cont.)



$I(s_1) = \varnothing \quad I(s_2) = \{s_1, s_3\} \quad I(s_3) = \{s_1\} \quad I(s_4) = \{s_2, s_3\} \quad I(s_5) = \{s_4\}$

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Performance evaluation of systems
Programs with sequential control structures

Non-structural program without loops (cont.)

Mean number of transitions $E(t_i)$ through state s_i of the Markov chain is:

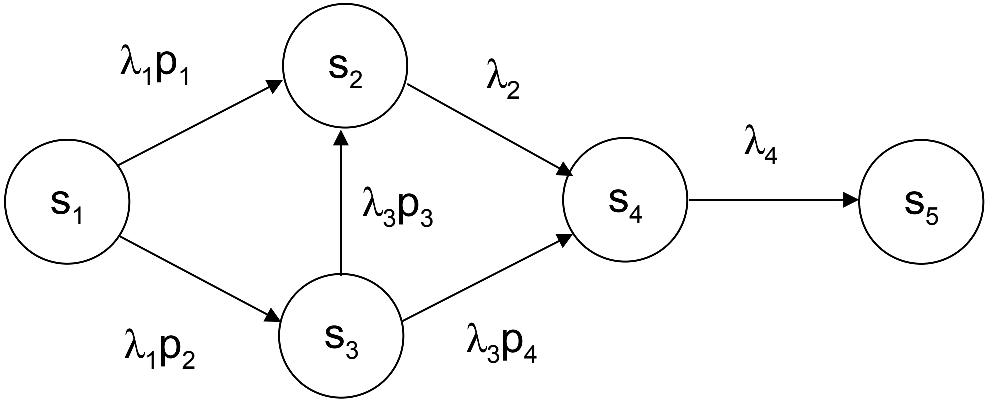
$$E(t_i) = \sum_{k \in I(s_i)} E(t_k) \cdot p_{ki}$$

$I(s_i)$ - a set of states that transitions to state s_i are directed from,

p_{ki} - a probability that the Markov chain from state s_k transits to state s_i.

Performance evaluation of systems
Programs with sequential control structures

Non-structural program without loops (cont.)



$$p_{ki} = \frac{\lambda_{ki}}{\sum_{j \in O(s_k)} \lambda_{kj}}$$

$$p_{12} = \frac{\lambda_1 \cdot p_1}{\lambda_1 \cdot p_1 + \lambda_1 \cdot p_2} = p_1 \qquad p_{13} = \frac{\lambda_1 \cdot p_2}{\lambda_1 \cdot p_1 + \lambda_1 \cdot p_2} = p_2 \qquad p_{32} = \frac{\lambda_3 \cdot p_3}{\lambda_3 \cdot p_3 + \lambda_3 \cdot p_4} = p_3$$

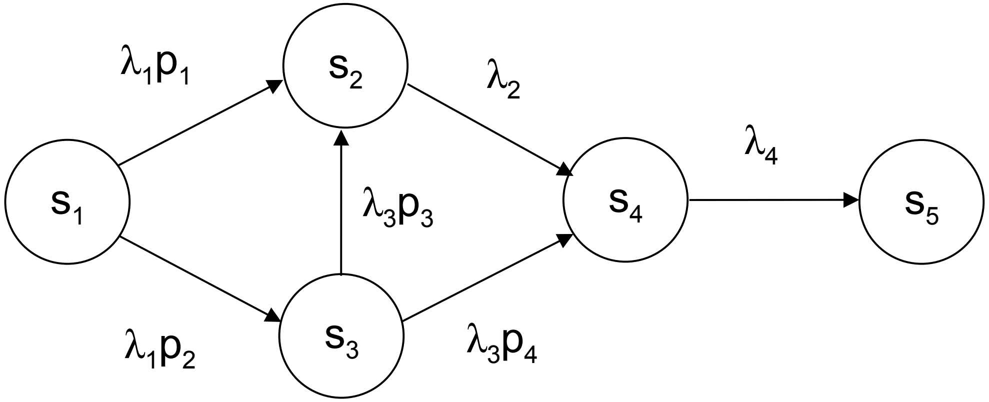
$$p_{24} = \frac{\lambda_2}{\lambda_2} = 1 \qquad p_{34} = \frac{\lambda_3 \cdot p_4}{\lambda_3 \cdot p_3 + \lambda_3 \cdot p_4} = p_4 \qquad p_{45} = \frac{\lambda_4}{\lambda_4} = 1$$

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Performance evaluation of systems

Programs with sequential control structures

Non-structural program without loops (cont.)



Mean number of transitions $E(t_i)$ through state of s_i the Markov chain is:

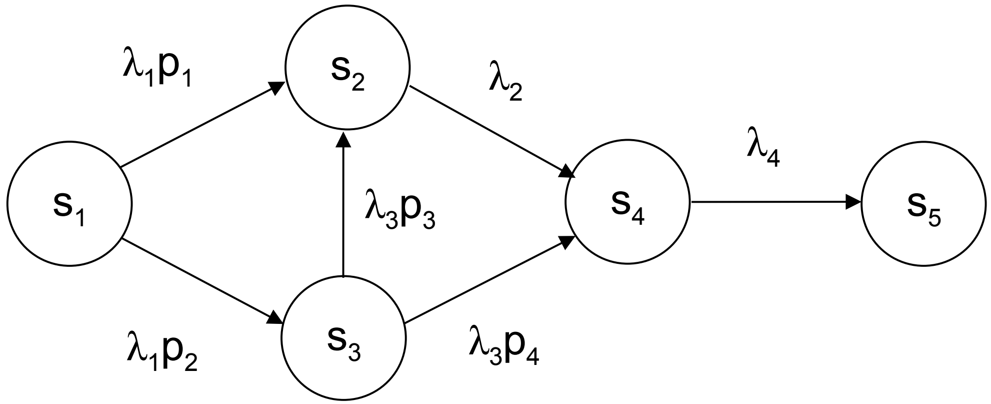
$$E(t_i) = \sum_{k \in I(s_i)} E(t_k) \cdot p_{ki}$$

Hence

$$E(t_1) = 1 \qquad E(t_2) = p_1 + p_2 \cdot p_3 \qquad E(t_3) = p_2 \qquad E(t_4) = 1$$

Performance evaluation of systems
Programs with sequential control structures

Non-structural program without loops (cont.)



Mean execution time of the program:

$$E(t) = \sum_{i=1}^{4} E(t_i) \cdot E(s_i)$$

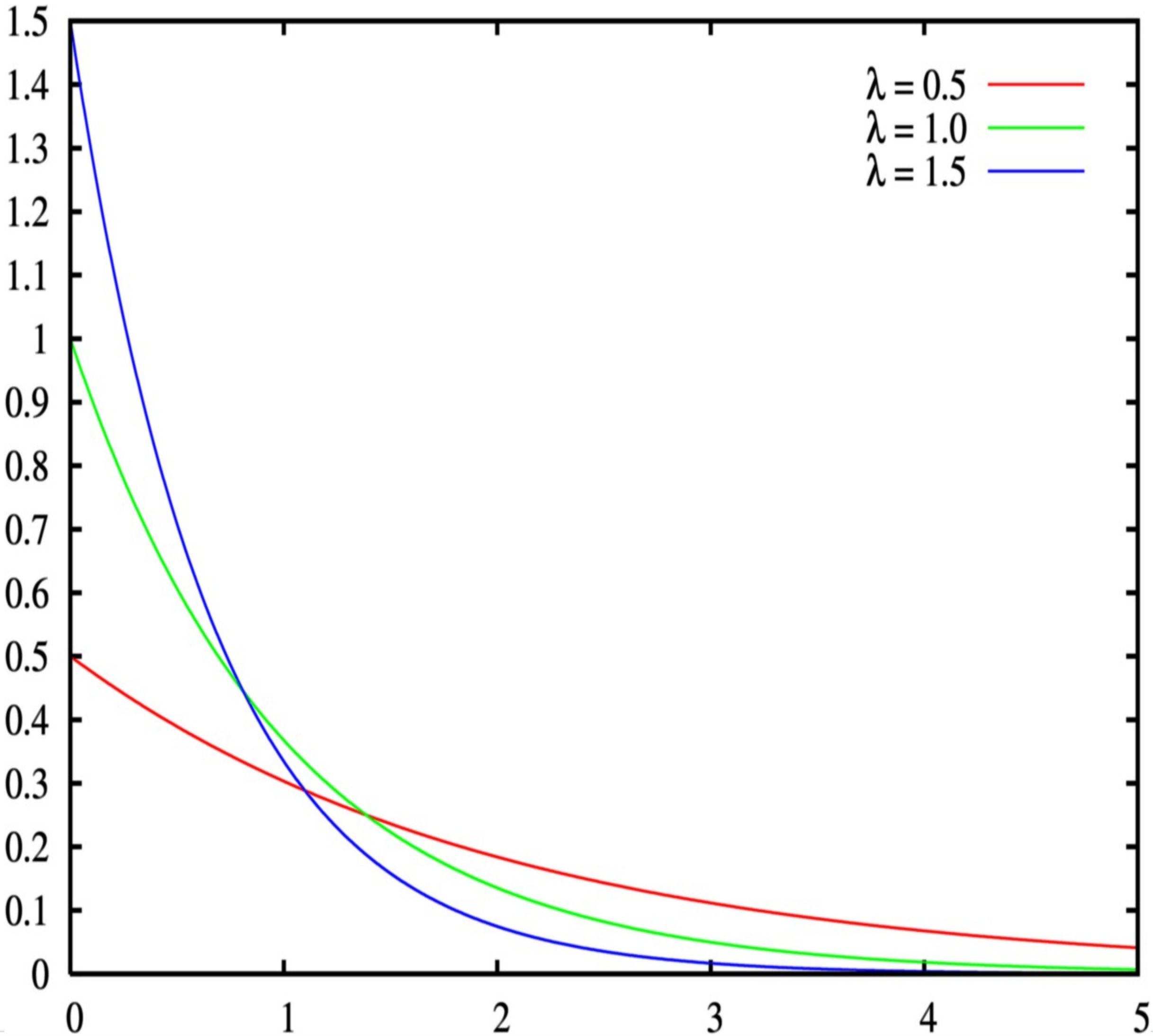
$E(t_i)$ - mean number of transitions through state s_i of the Markov chain,
$E(s_i)$ - mean sojourn time in transient state s_i of the Markov chain.

Information Systems Analysis

Jan Magott

Performance evaluation of systems
Markov chains with continuous time

Choose yourself and new tech



Probability density function of exponential random variable X

$$f_X(x) = \lambda e^{-\lambda x}$$

Mean value of X

$$m_X = \frac{1}{\lambda_X}$$

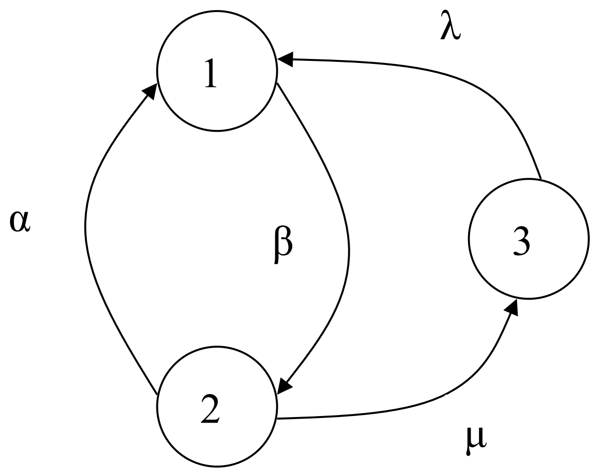
Variance of X

$$\sigma^2 = \frac{1}{{\lambda_X}^2}$$

Figure from Wikipedia

Performance evaluation of systems

Markov chains with continuous time



1, 2, 3 – Markov chain states,

α, β, λ, μ – transition intensities

(parameters of exponential distributions of transition times between states)

Probability density function $f_X(x) = \lambda \cdot e^{-\lambda x}$

m_X - mean value of random variable X

$$m_X = \frac{1}{\lambda_X}$$

$$Q = \begin{bmatrix} -\beta & \beta & 0 \\ \alpha & -(\alpha+\mu) & \mu \\ \lambda & 0 & -\lambda \end{bmatrix}$$

A transition intensities matrix

A diagram of transition intensities between states

Performance evaluation of systems

Markov chains with continuous time

$q_{ij},\ i \neq j,$ - intensity of transition from state i into state j

$$q_{ii} = -\sum_{j \neq i} q_{ij}$$

$\pi_i(t)$ - probability that in time instant t the process is in state i

$q_{ij} \cdot h,\ i \neq j,\ h \rightarrow 0$ - probability that the process will transit from state i into state j in time interval of length h provided $h \rightarrow 0$

$$\pi_1(t+h) = \pi_1(t) \cdot (1 - \beta \cdot h) + \pi_2(t) \cdot \alpha \cdot h + \pi_3(t) \cdot \lambda \cdot h + o(h)$$

$o(h)$ is such that $\lim_{h \rightarrow 0} \frac{o(h)}{h} = 0$

Performance evaluation of systems

Markov chains with continuous time

$$\pi_1(t+h) = \pi_1(t)\cdot(1-\beta\cdot h) + \pi_2(t)\cdot\alpha\cdot h + \pi_3(t)\cdot\lambda\cdot h + o(h)$$

$$\lim_{h\to\infty}\frac{\pi_1(t+h)-\pi_1(t)}{h} =$$

$$\lim_{h\to\infty}\left(\frac{-\pi_1(t)\cdot\beta\cdot h}{h} + \frac{\pi_2(t)\cdot\alpha\cdot h}{h} + \frac{\pi_3(t)\cdot\lambda\cdot h}{h} + \frac{o(h)}{h}\right)$$

$$\pi_1'(t) = -\pi_1(t)\cdot\beta + \pi_2(t)\cdot\alpha + \pi_3(t)\cdot\lambda$$

- In a stationary state:

$$\lim_{t\to\infty}\pi_i'(t) = 0 \quad\Rightarrow\quad \lim_{t\to\infty}\pi_i(t) = \pi_i$$

$$-\pi_1\cdot\beta + \pi_2\cdot\alpha + \pi_3\cdot\lambda = 0$$

Performance evaluation of systems

Markov chains with continuous time

- What is the relation between coefficients of obtained equation:

$$-\pi_1 \cdot\beta + \pi_2 \cdot\alpha + \pi_3 \cdot\lambda = 0$$

and the transition intensities matrix:

$$Q = \begin{bmatrix} -\beta & \beta & 0 \\ \alpha & -(\alpha+\mu) & \mu \\ \lambda & 0 & -\lambda \end{bmatrix}$$

?

Recommendation: Read about Chapman-Kolmogorov equations, please.

Performance evaluation of systems

Markov chains with continuous time

$$-\pi_1 \cdot \beta + \pi_2 \cdot \alpha + \pi_3 \cdot \lambda = 0$$
$$\pi_1 \cdot \beta - \pi_2 \cdot (\alpha + \mu) + \pi_3 \cdot 0 = 0$$
$$\pi_1 \cdot 0 + \pi_2 \cdot \mu - \pi_3 \cdot \lambda = 0$$

The above equations are dependent.

Hence, an additional equation is required.

$$\pi_1 + \pi_2 + \pi_3 = 1$$

Solution:

$$\Pi = \frac{1}{\lambda(\alpha + \beta) + \mu(\lambda + \beta)} \begin{bmatrix} \lambda(\alpha + \mu) & \lambda\beta & \beta\mu \end{bmatrix}$$

Performance evaluation of systems

Markov chains with continuous time

- The ergodic (stationary) solution is obtained by solving the following linear equation system

$$\Pi \cdot Q = 0$$

where

$$\Pi = [\pi_1, \ldots, \pi_i, \ldots, \pi_n]$$

n - a number of states

$$\sum_{i=1}^{i=n} \pi_i = 1$$

$X(t)$ - process state at a time instant t

$\pi_i = \lim_{t \to \infty} P\{X(t) = i\}$ - probability that the process in a stationary state is in state i

Performance evaluation of systems

Markov chains with continuous time

The equations obtained :

$$-\pi_1 \cdot \beta + \pi_2 \cdot \alpha + \pi_3 \cdot \lambda = 0$$

$$\pi_1 \cdot \beta - \pi_2 \cdot (\alpha + \mu) + \pi_3 \cdot 0 = 0$$

$$\pi_1 \cdot 0 + \pi_2 \cdot \mu - \pi_3 \cdot \lambda = 0$$

can be transformed into the following equation system:

$$\pi_1 \cdot \beta = \pi_2 \cdot \alpha + \pi_3 \cdot \lambda$$

$$\pi_2 \cdot (\alpha + \mu) = \pi_1 \cdot \beta$$

$$\pi_3 \cdot \lambda = \pi_2 \cdot \mu$$

Let us analyse the first equation: $\pi_1 \cdot \beta = \pi_2 \cdot \alpha + \pi_3 \cdot \lambda$

Performance evaluation of systems

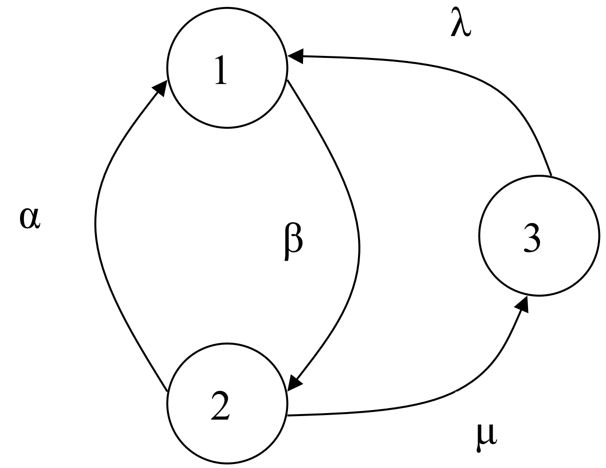
Markov chains processes with continuous time

Let us analyse the first equation: $\pi_1 \cdot\beta = \pi_2 \cdot\alpha + \pi_3 \cdot\lambda$

Flow from the state 1 is: $\pi_1 \cdot\beta$

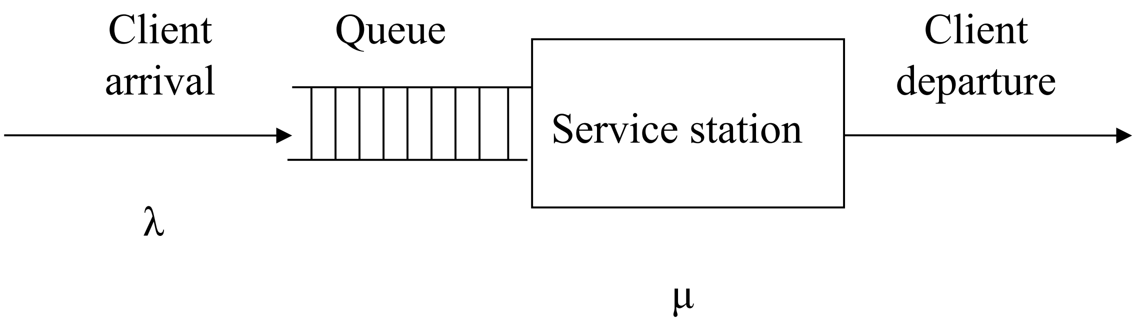
Flow to the state 1 is: $\pi_2 \cdot\alpha + \pi_3 \cdot\lambda$

In a stationary state, both flows are equal.



Performance evaluation of systems

Markov chains processes with continuous time



M/M/1 queue

M (before „/”) – a client arrival process is Poisson process with parameter λ.

Time between arrivals of i-th client and (i+1)-th client is expressed by an exponential distribution with parameter λ.

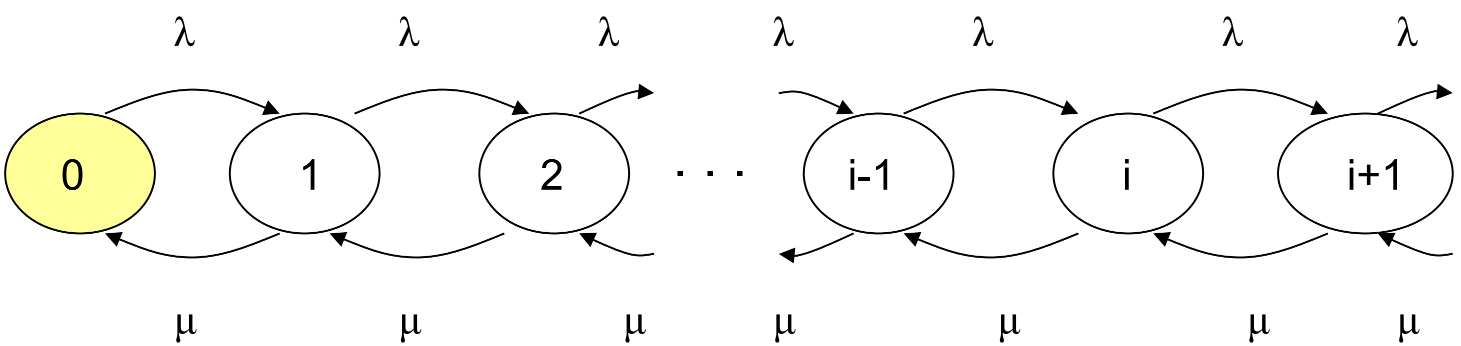
M (between „/” and „/”) – service time is described by an exponential distribution with parameter μ.

1 – one service element at the service station.

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Performance evaluation of systems

Markov chains with continuous time

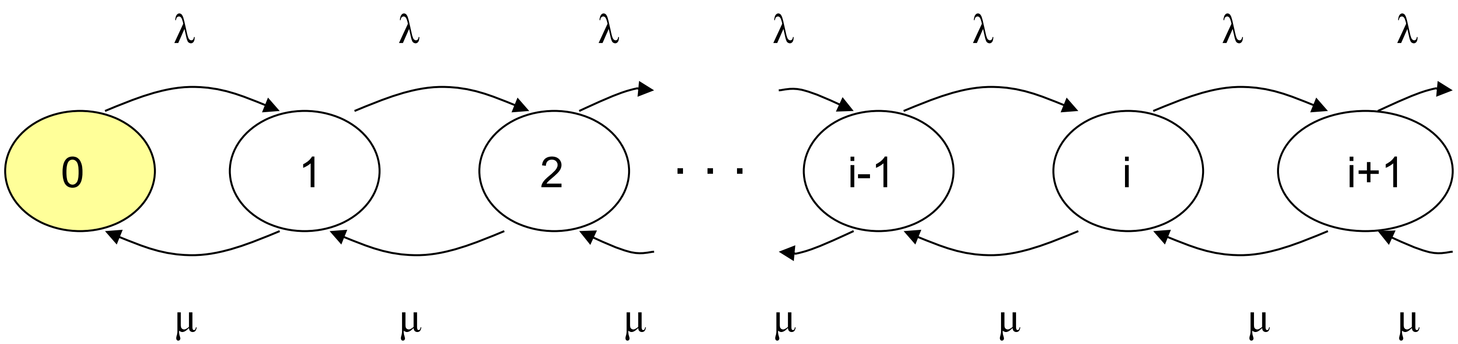


i – state that represents number of clients in the system (number of clients in the queue and client in the service station)

A transition intensities diagram of an M/M/1 queue

Performance evaluation of systems

Markov chains with continuous time



$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & \cdots \\ \mu & -(\lambda+\mu) & \lambda & \cdots \\ 0 & \mu & -(\lambda+\mu) & \cdots \\ 0 & 0 & \mu & \cdots \\ . & . & . & \cdots \\ . & . & . & \cdots \end{bmatrix}$$

A transition intensities matrix of an M/M/1 queue

Performance evaluation of systems
Markov chains with continuous time

- A stationary state condition:

$$\lambda<\mu$$

Otherwise length of the queue can be infinite.

Stationary state solution:

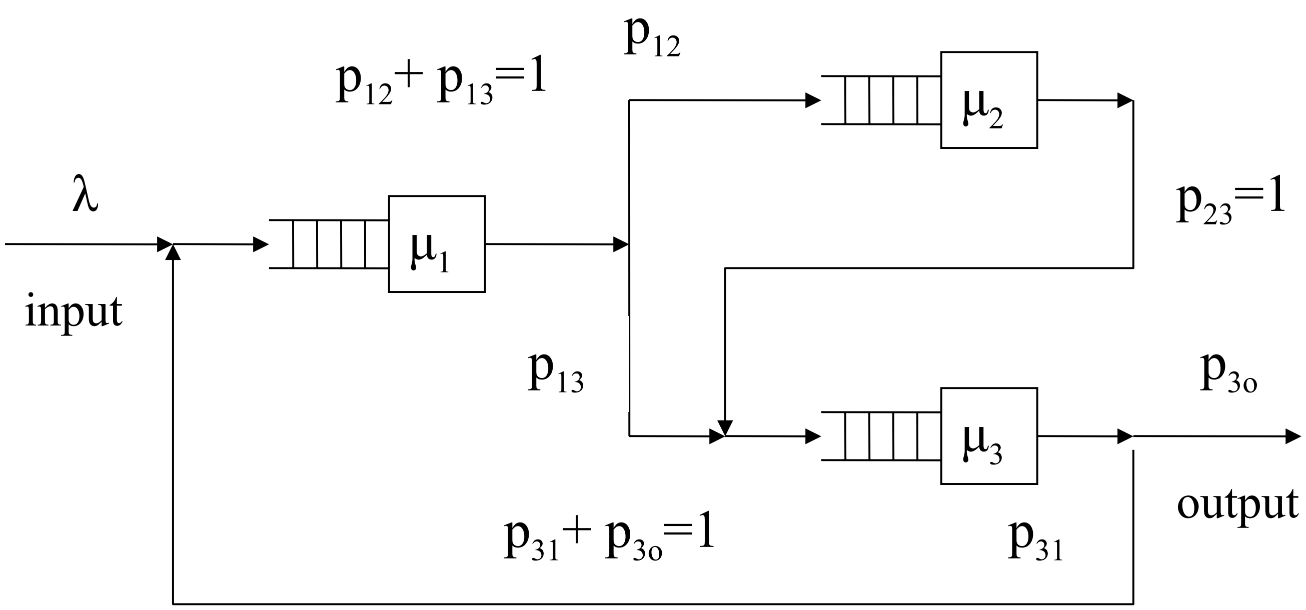
$$\pi_i = (1 - \frac{\lambda}{\mu})(\frac{\lambda}{\mu})^i, \quad i \geq 0$$

$$\frac{\lambda}{\mu} = \rho \text{ - a station utility coefficient}$$

$$\pi_0 = 1 - \rho \qquad \pi_{>0} = \rho$$

Performance evaluation of systems

Markov chains with continuous time



p_{3o} - a probability that a client leaves the system after service at station 3

A queueing network model

Performance evaluation of systems
Markov chains with continuous time

- Weeknesses of queueing network models:

  Many synchronisation modes cannot be represented, e.g. handshaking.

  Impasses cannot be expressed.

- Petri nets can express the above aspects.

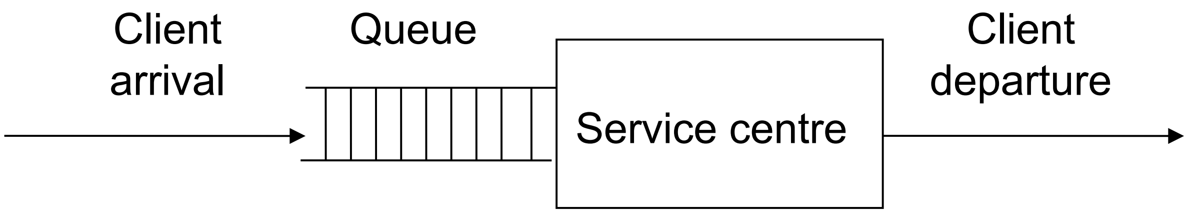
  Incorporating a time factor into Petri nets enables performance evaluation.

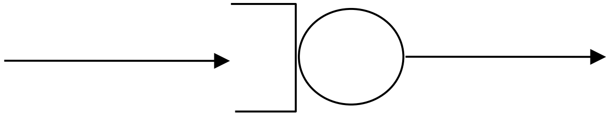
Information Systems Analysis

Jan Magott

Performance evaluation using queuing networks

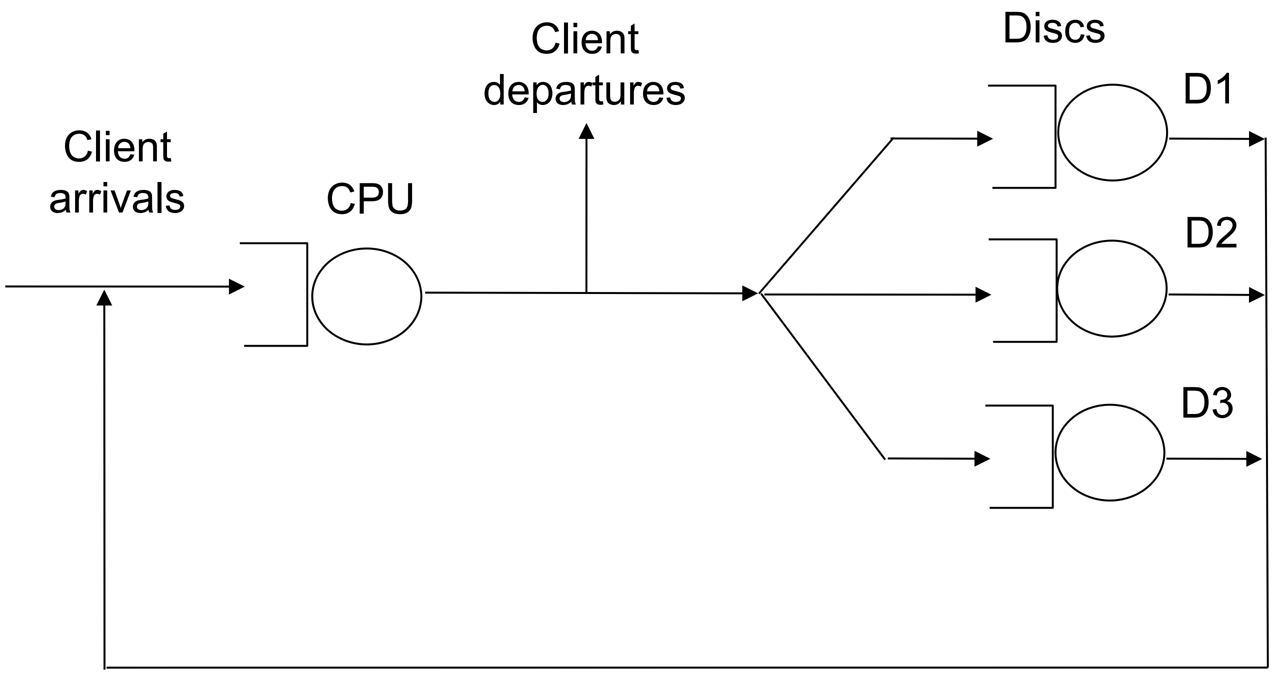
Performance evaluation using queuing networks





A service centre with a queue and its compact representation

Performance evaluation using queuing networks



A queuing network model of a computer system

Performance evaluation using queuing networks

A queuing network model of a computer system (cont.)

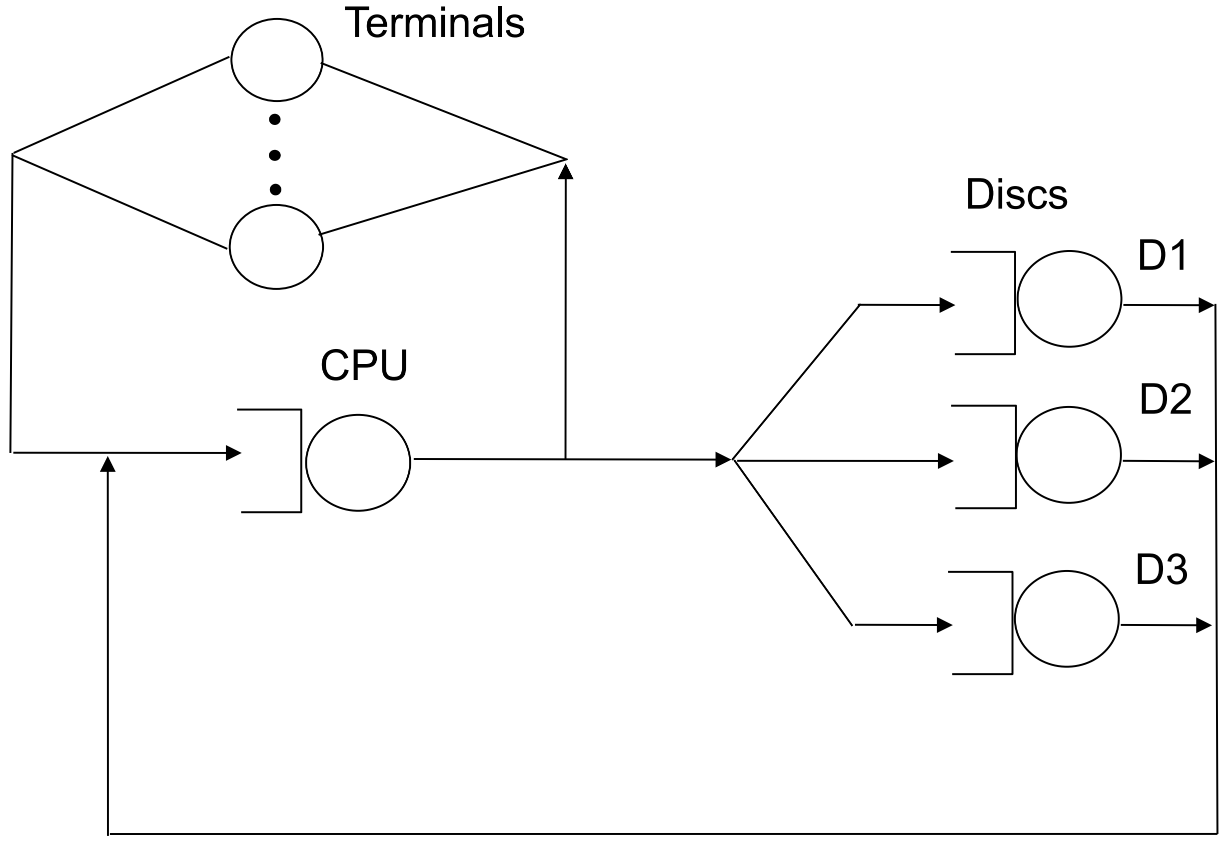
- Parameters

Workload – a *client arrival intensity* (e.g. 2 clients/sec,
A Poisson process with parameter λ= 2 clients/sec)

Service requirements – service time per one visit at service centre
(e.g. 3 sec at CPU, 1 sec at D1, 3 sec at D2,
expressed by a random variable),
a *number of visits at service centre*
(e.g. 3 at CPU,
expressed by a random variable with natural values).

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Performance evaluation using queuing networks



A queuing network model of the computer system with terminals

Performance evaluation using queuing networks

A queuing network model of the computer system with terminals (cont.)

- Parameters

Workload:

1. a *number of clients* (active terminals),
2. *mean time of work on terminal* (time between client departure from the system and client arrival to the system).

Service requirements:

3. *service time per one visit at service centre,*
4. a *number of visits at service centre.*

Performance evaluation using queuing networks

- *Queuing networks features*:

  The request (client) is moving through service centres,
  The following service centres are modelled:
  - service centres with FIFO queue (express CPU with FIFO scheduling discipline),
  - service centres with the Round-Robin service discipline,
  - delay dervice centres (represent transmission medium),

  When a service of a request at a service centre has been completed, then the request can move to the other service centre,
  A next service centre can be selected according to a discrete probability distribution.

Project co-financed from the EU European Social Fund

Performance evaluation using queuing networks

- *Elementary queuing networks features*:

  Requests compete for resources, but they do not cooperate,
  The requests are atomic in the sense that they are not combined from the smaller parts,
  Neither synchronization nor communication between requests can be described,
  If a service at a given centre has been completed then the centre is no longer allocated to the request, so deadlocks cannot be modelled.

- Expressive power of the elementary queueing networks is smaller than expressive power of Petri nets with the time factor.

- The main advantage of elementary queueing networks is their relatively small computational complexity, when comparing with Petri nets with the time factor.

Performance evaluation using queuing networks

- System characteristics:

  Utilization (the part of the time the service centre, e.g. CPU is busy),

  Mean residence time (the mean time from client arrival instant till client departure instant),

  Mean number of clients in the system (a mean number of clients at all service centres, eg. CPU, discs, and in all queues),

  System throughput (a mean number of clients that leave the system, i.e. their requests become completed, per one time unit).

Performance evaluation using queuing networks

- Continuous time Markov chains approach (features):
  - Chapman-Kolmogorov equations,
  - Linear equality system with a number of equalities equal to the number of system states,
  - Non-intuitive,
  - Complex analytical solutions.

- <u>Operational analysis approach</u> (features):
  - Analysis in terms of mean values,
  - More intuitive approach than the Markov chains approach.

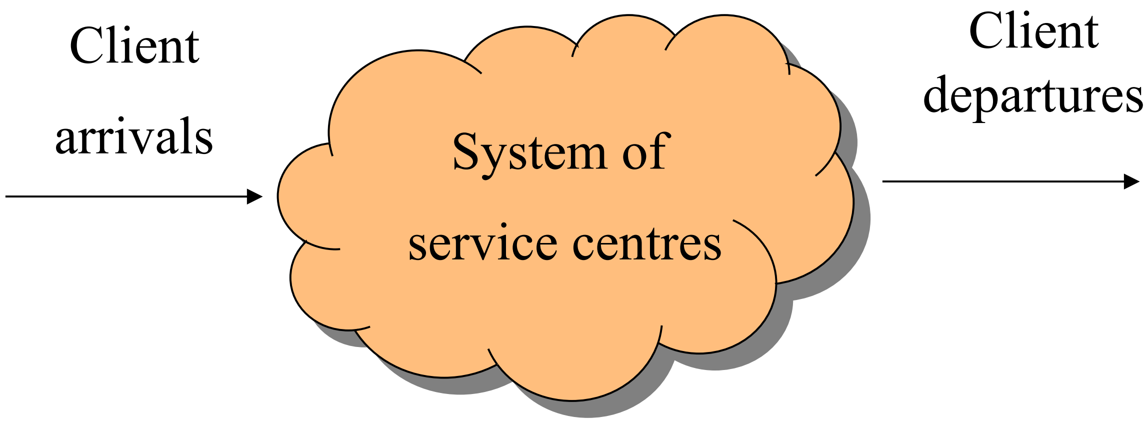
Information Systems Analysis

Jan Magott

Performance evaluation using queuing networks
Fundamental laws of operational analysis

Performance evaluation using queuing networks

Fundamental laws of operational analysis



T – length of an observation time interval,

A – a number of client (requests) arrivals in the time interval,

C – a number of clients whose service has been completed in the time interval (*number of client departures in the time interval*).

Performance evaluation using queuing networks

Fundamental laws of operational analysis

T – a system observation time interval length,

A – a number of client arrivals in the time interval,

C – a number of clients whose service has been completed in the time interval (number of client departures in the time interval).

λ – an arrival rate

$$\lambda = \frac{A}{T}$$

X – throughput

$$X = \frac{C}{T}$$

Performance evaluation using queuing networks

Fundamental laws of operational analysis

- Assumption
  The system contains one service station only.

B – busy time (lenght of time interval when the system is busy),

U – utilization,

$$U = \frac{B}{T}$$

S – mean service time

$$S = \frac{B}{C}$$

Utilization law

$$U = \frac{B}{T} = \frac{C}{T} \cdot \frac{B}{C} = X \cdot S \quad \Rightarrow \quad U = X \cdot S$$

Performance evaluation using queuing networks

Fundamental laws of operational analysis

- *Little's law*

N – a mean number of clients in system,
k – a number of intervals when number of clients in the system is constant,
I_i - a length of *i*-th time interval,
n_i - a number of clients in *i*-th time interval,

$$N = \sum_{i=1}^{k} n_i \cdot \frac{I_i}{T} = \frac{1}{T} \cdot \sum_{i=1}^{k} n_i \cdot I_i = \frac{1}{T} \cdot W$$

$W = \sum_{i=1}^{k} n_i \cdot I_i$ - total residence (in queues and at servive stations) time of all clients in a time interval length T,

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!
Wrocław University of Technology
EUROPEAN SOCIAL FUND
Project co-financed from the EU European Social Fund

Performance evaluation using queuing networks

Fundamental laws of operational analysis

- Little's law (cont.)

$$N = \frac{W}{T} = \frac{C}{T} \cdot \frac{W}{C}$$

$\frac{C}{T} = X$ - throughput

$\frac{W}{C} = R$ - mean residence time of clients that have been serviced

- Finally

$$N = X \cdot R$$

Performance evaluation using queuing networks

Fundamental laws of operational analysis

- Little's law (cont.)

$$N = X \cdot R$$

Informal explanation

During the residence time, a request is moving from the end of the queue to the output of the service centre. The mean number of clients *N* in the system is equal to the mean number of requests that enter the service centre during the mean residence time *R*. If the service centre is in a stationary state, then the mean number of requests coming to the centre per one time unit is equal to the throughput *X*. Therefore, the mean number of requests that enter the centre during the mean residence time *R* is equal to $X \cdot R$.

Performance evaluation using queuing networks

Fundamental laws of operational analysis

- Little's law (cont.)

Importance

$$N = X \cdot R$$

If two from the three values: *N, X, R* are known from measurements of a service centre, then the third one can be calculated.

Little's law is widely used in queueing model analysis. Assumptions to apply the law are rather weak. In queueing network analysis, memoryless property (Markov property) of a stochastic process is often required. Such assumption is not required for Little's law. The law can be applied even for systems with so strongly past dependent behaviour as systems with deterministic models of input stream and service.

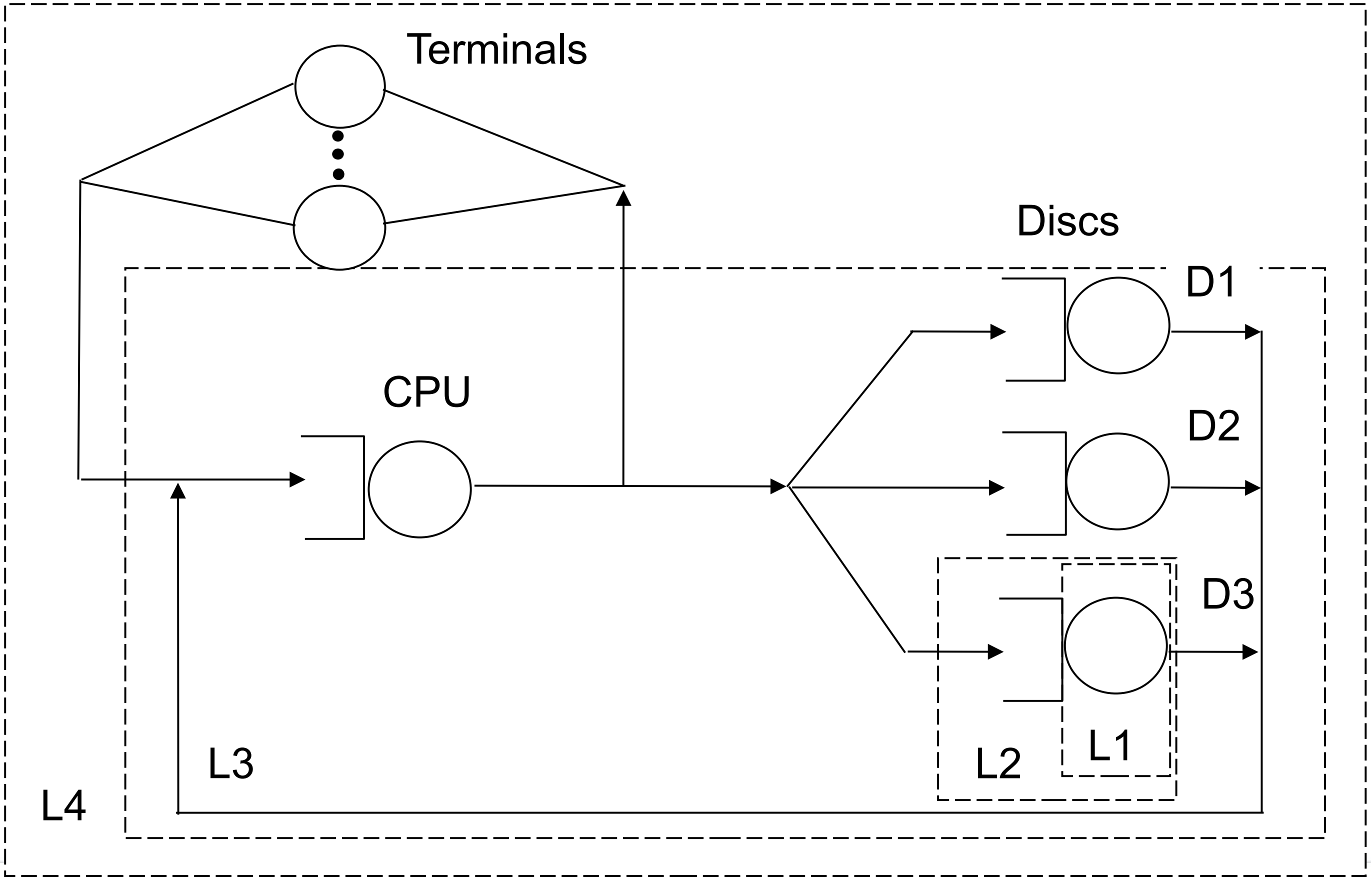
The law can be applied at different levels of system organization.

Performance evaluation using queuing networks

Fundamental laws of operational analysis

- Little's law (cont.)

- Levels
  L1, L2, L3, L4
  of system analysis



Performance evaluation using queuing networks

Fundamental laws of operational analysis

- L1 - disc D3 without its queue

$$N = X \cdot R$$

N – a mean number of clients at D3 without its queue,

X – throughput of D3,

R – mean residence time at D3, i.e., mean service time S at D3.

There can be 0 or 1 client at D3.

$$N = 0 \cdot \frac{I_0}{T} + 1 \cdot \frac{I_1}{T}$$

I_0 (I_1) - total length of intervals when there is 0 (1) clients (client) at D3

$$N = \frac{I_1}{T} = \frac{B}{T} = U$$

Hence,

$$U = X \cdot S$$

Conclusion: Utilization law is a particular case of Little's law.

Performance evaluation using queuing networks

Fundamental laws of operational analysis

- L2 - disc D3 with its queue
  N – a mean number of clients at D3 and in its queue,
  X – throughput of D3,
  R – mean residence time at D3 and in its queue,
  S – mean service time at D3.

$$N = X \cdot R$$

Assumption: $N, X,$ and S are given. Hence, $R = \frac{N}{X}$

Q – mean time in the queue $\quad Q = R - S$

Therefore, $\quad XQ = XR - XS$

$XQ = N_Q$ – a mean number of clients in the queue,

$XR = N$

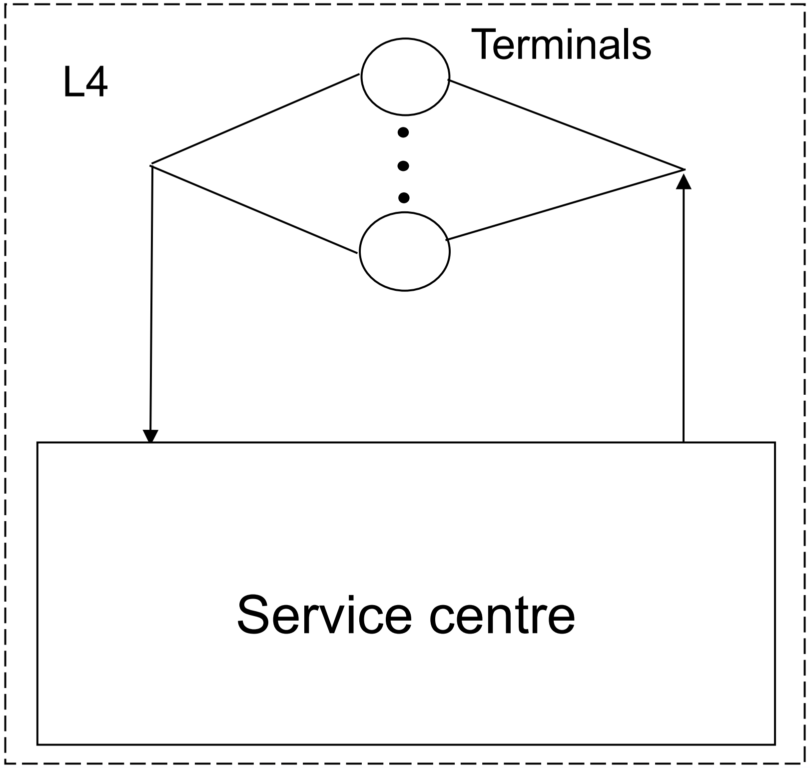
$XS = U$ – utilization.

Finally: $\quad N_Q = N - U$

Performance evaluation using queuing networks

Fundamental laws of operational analysis

- L4 – a service centre with terminals as the *interactive system* that Little's law is applied to



Performance evaluation using queuing networks

Fundamental laws of operational analysis

N – a number of requests (clients) in the interactive system,

X – throughput, i.e., the rate of interaction flow between terminals and service centre,

R – the mean residence time in the service center, i.e., the response time,

Z – the mean thinking time,

R' – the mean residence time in the interactive system:

$$R' = R + Z$$

According to Little's law:

$$N = XR' = X(R + Z)$$

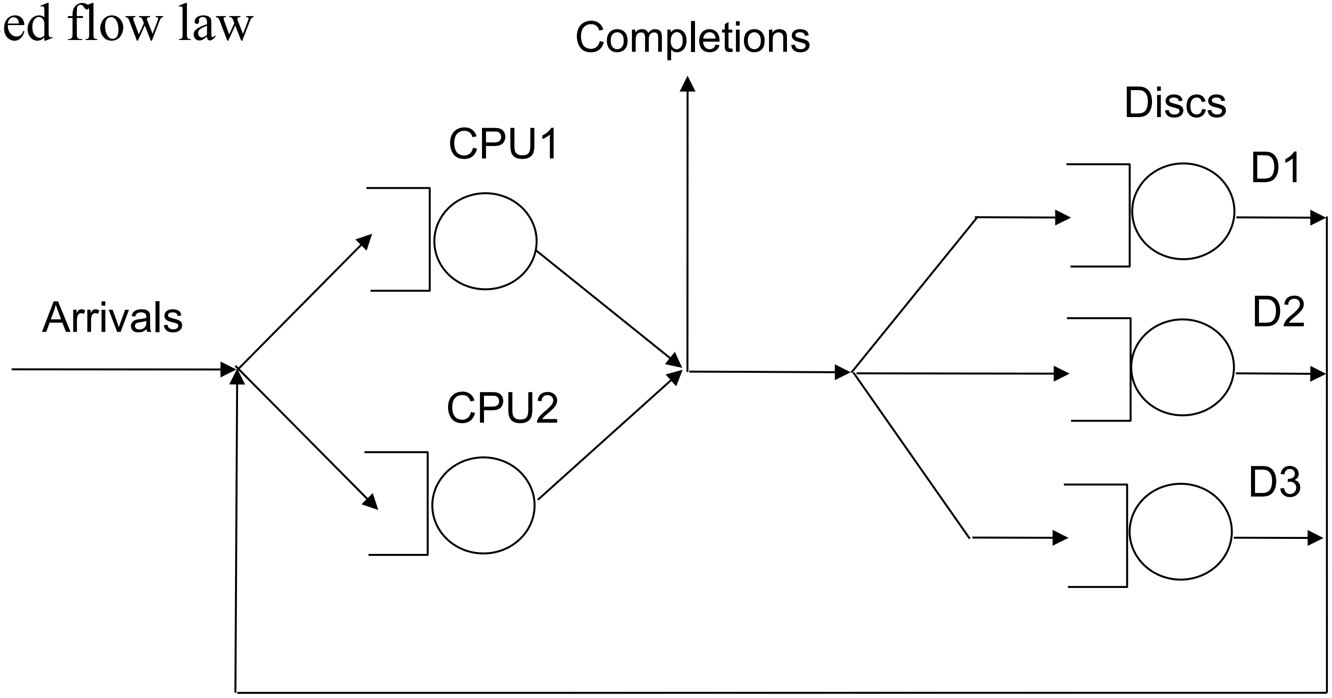
Response time law:

$$R = \frac{N}{X} - Z$$

Performance evaluation using queuing networks

Fundamental laws of operational analysis

- Forced flow law



Performance evaluation using queuing networks

Fundamental laws of operational analysis

C – a number of requests completed by the system during observation time,

C_k – a number of requests completed at the *i*-th service centre during the observation time,

V_k – mean number of visits of a request at the *i*-th service centre

$$V_k = \frac{C_k}{C}$$

Hence,

$$C_k = V_k \cdot C \quad \Rightarrow \quad \frac{C_k}{T} = V_k \cdot \frac{C}{T}$$

Forced flow law:

$$X_k = V_k \cdot X$$

X_k – throughput of the *i*-th service centre

X – throughput of the system

Performance evaluation using queuing networks

Fundamental laws of operational analysis

S_k - the mean service time for one visit of a request at the i-th service centre,
D_k - the mean total service time of the request at the i-th service centre,

Utilization of the *i*-th service centre:

$$U_k = X_k \cdot S_k = X \cdot V_k \cdot S_k = X \cdot D_k$$

„Bottleneck" of queuing network

λ - client arrival rate, X - system throughput,

$$u_k = X \cdot D_k$$

where

$$D_k = V_k \cdot S_k$$

$$X_k = V_k \cdot X$$

where X_k - k-th service centre throughput, S_k - mean service time for one visit at k-th service centre,

D_k - mean service time for all visits of one request at k-th service centre,

$UB(X)$ - the upper bound of throughput X ,

$$UB(X) = 1/D_{max}$$

where $D_{max} = \max\{D_k : k \in I\}$ I - the set of indices of stations,

For „bottleneck":

$$u_j = 1$$

Open system (transactional)

Assumption:

1. $\lambda < \lambda_{Sat} = 1/D_{max}$
2. Service remaining time does not depend on passed service time; it is true for exponential distribution,

$$R_k(\lambda) = V_k[S_k + S_k \cdot A_k(\lambda)]$$

$A_k(\lambda)$ - mean number of requests at k-th service centre and in its queue when new request arrives this centre for request arrival rate λ,

$$R_k(\lambda) = D_k[1 + A_k(\lambda)]$$

$$A_k(\lambda) = N_k(\lambda)$$

From Little's law $\quad N_k(\lambda) = X(\lambda) \cdot R_k(\lambda)$

In stationary state: $\quad X(\lambda) = \lambda$

$$R_k(\lambda) = D_k[1 + \lambda \cdot R_k(\lambda) \quad]$$

Open system (transactional)

$$R_k(\lambda) = D_k[1 + \lambda \cdot R_k(\lambda) \quad]$$

$$R_k(\lambda) - D_k \cdot \lambda \cdot R_k(\lambda) = D_k$$

$$R_k(\lambda) = \frac{D_k}{1 - \lambda \cdot D_k}$$

$$R_k(\lambda) = \frac{D_k}{1 - u_k(\lambda)}$$

λ - client arrival rate

for station k (In stationary state)

$$U_k = \lambda \cdot D_k$$

where $D_k = V_k \cdot S_k$

$$X_k = V_k \cdot X$$

station throughput — system throughput

S_k - mean service time for one visit

D_k - mean service time for all visits of one client at this station

$$UB(X) = \frac{1}{D_{max}}$$

$$D_{max} = \max\{D_k : k \in I\}$$

I - set of indices of stations

UB(X) - upper bound of throughput

... bottleneck

2. Service remaining time does not depend on passed service time

(It is true for exponential distribution)

$$R_k(\lambda) = V_k[S_k + S_k \cdot A_k(\lambda)]$$

$A_k(\lambda)$ – mean number of clients at service station and in the queue when new client arrives the station

$$R_k(\lambda) = D_k[1 + A_k(\lambda)]$$

Hence $A_k(\lambda) = Q_k(\lambda)$

$$R_k(\lambda) = D_k[1 + Q_k(\lambda)]$$

$$R_k(\lambda) = D_k[1 + \lambda \cdot R_k(\lambda)]$$

Information Systems Analysis

Jan Magott

Stochastic Petri nets

Choose yourself and new tech

Stochastic Petri nets

- Alternatives of introducing a time factor into Petri nets

  Nature of time specification:
  - Deterministic,
  - Non-deterministic,
  - Probabilistic.

  Petri net elements that the time factor is assigned to:
  - Place,
  - Transition,
  - Arc.

Stochastic Petri nets

- Transition firing semantics

With regard to a transition firing rule:
Atomic firing (tokens are removed from input places of transition and added to output places in a single indivisible operation)
Firing in three phases:
firing initialization (tokens are removed from input places),
time passing (elapsing),
firing completion (tokens are added to output places).

With regard to transition firing multiplicity:
single server semantics (at each time instant transition can be fired at most once),
multiple server semantics (at each time instant transition can be fired more than once).

Stochastic Petri nets

- Definition

A stochastic Petri net (basic model) is a 5-tuple:

$$SPN = < P, T, F, M_0, \Lambda >$$

P – a set of places,

T – a set of transitions,

$F \subseteq (P \times T) \cup (T \times P)$ – a set of arcs,

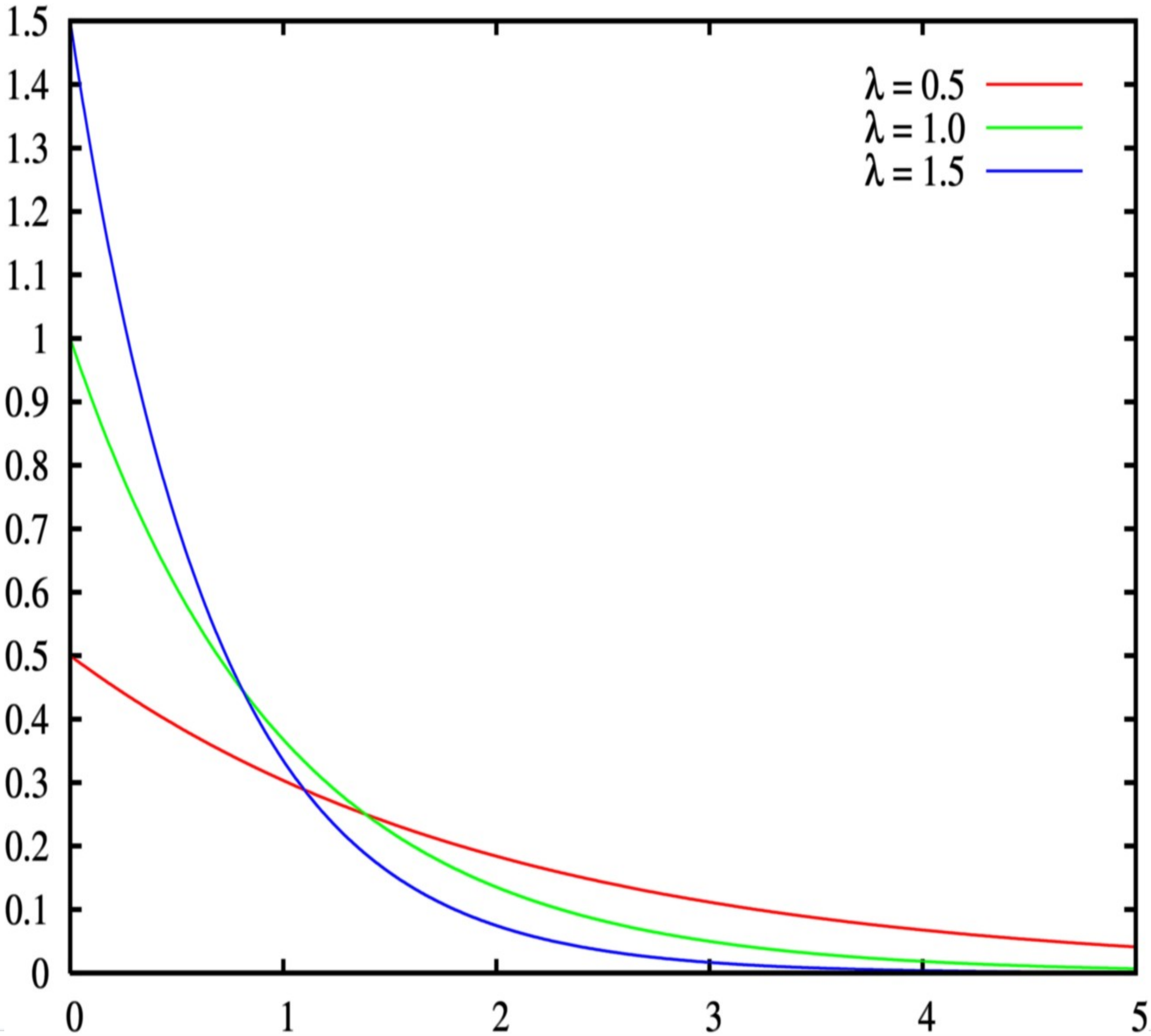
$M_0 : P \rightarrow \{0,1,2,...\}$ – an initial marking function,

$\Lambda = < \lambda_1, ..., \lambda_i, ..., \lambda_{|T|} >$ – a transition firing intensity vector,

firing intensity is the parameter of exponential random variable of firing time.

The arc weight function is omitted because $W : F \rightarrow \{1\}$

The place capacity function is omitted because $C : P \rightarrow \{\infty\}$



Probability density function of exponential random variable X

$$f_X(x) = \lambda e^{-\lambda x}$$

Mean value of X

$$m_X = \frac{1}{\lambda_X}$$

Variance of X

$$\sigma^2 = \frac{1}{{\lambda_X}^2}$$

Figure from Wikipedia

Stochastic Petri nets

- Entries of the transition firing intensity vector

$$\Lambda = <\lambda_1, \ldots, \lambda_i, \ldots, \lambda_{|T|}>$$

can be constants or functions.

Entry of the transition firing intensity vector as a function of marking:

$$\lambda_i : R(M_0) \rightarrow R_+$$

$R(M_0)$ is the reachability set for the initial marking M_0.

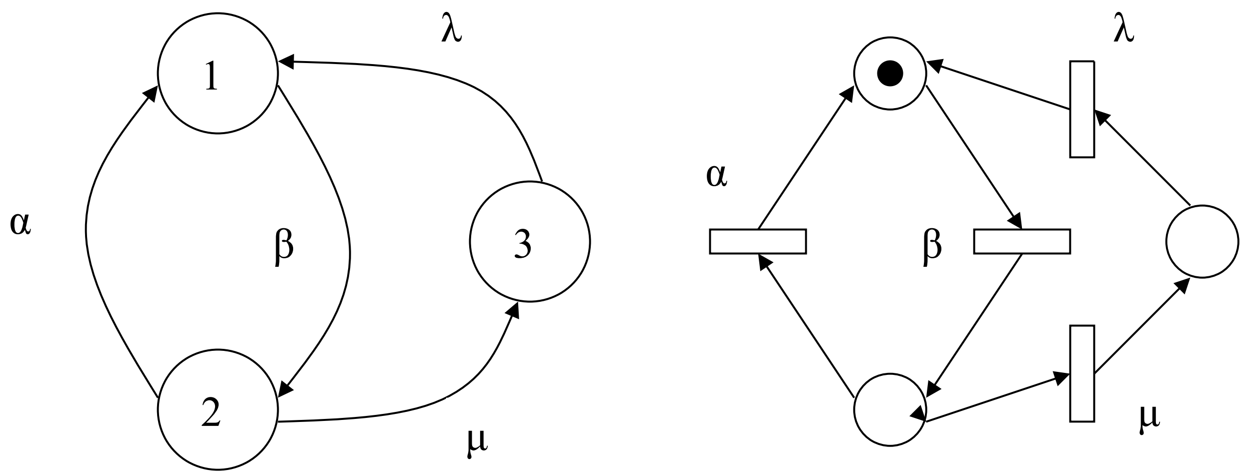
Firing is atomic (tokens are removed from input places of transition and added to output places in a single indivisible operation).

Stochastic Petri nets

- Firing is atomic (tokens are removed from input places of transition and added to output places in one indivisible operation).
- In given time instant, at most one transition can be fired. It is the consequence of the fact that transition firings process is Markov process.
- Firing time of a transition is the lentgh of time interval from instant when the transition became enabled till instant when the transition is fired.
- Firing time of transition t_i is expressed by exponential random variable with constant or functional, respectively, parameter λ_i or $\lambda_i(M_j)$
- For functional parameter, mean firing time of transition t_i in marking M_j :

$$\frac{1}{\lambda_i(M_j)}$$

Stochastic Petri nets



Markov process with continuous time and with „1" as its initial state represented by a stochastic Petri net

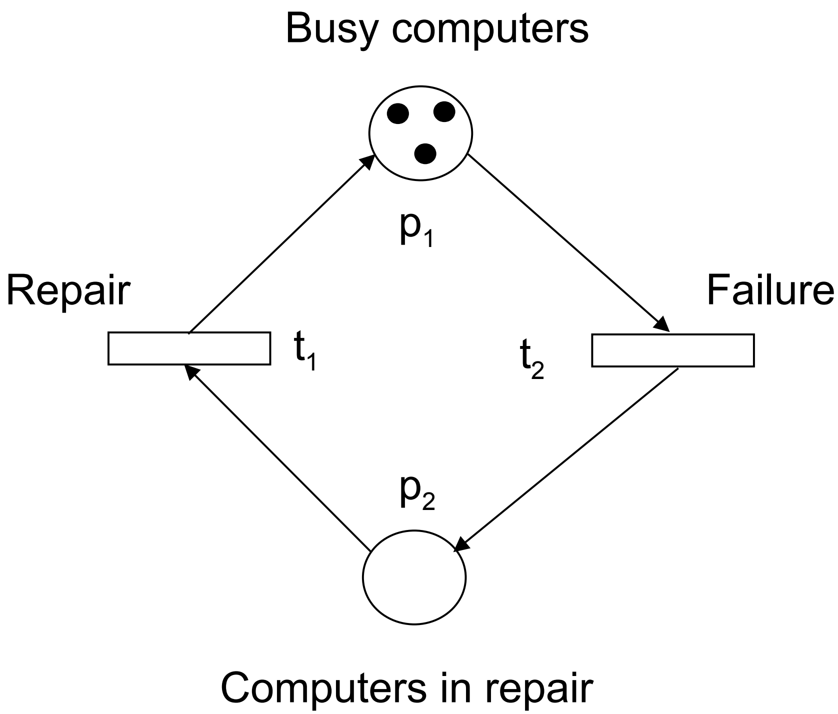
Stochastic Petri nets

- An example of a computer system which will be modelled by a stochastic Petri net

  Features:

  - System is combined from three computers.
  - For each computer, its time to failure is described by an exponential random variable with parameter λ.
  - Repair time of any computer is expressed by an exponential random variable with parameter μ. One computer only can be repaired at any given time instant.

Stochastic Petri nets



- Transition firing intensities

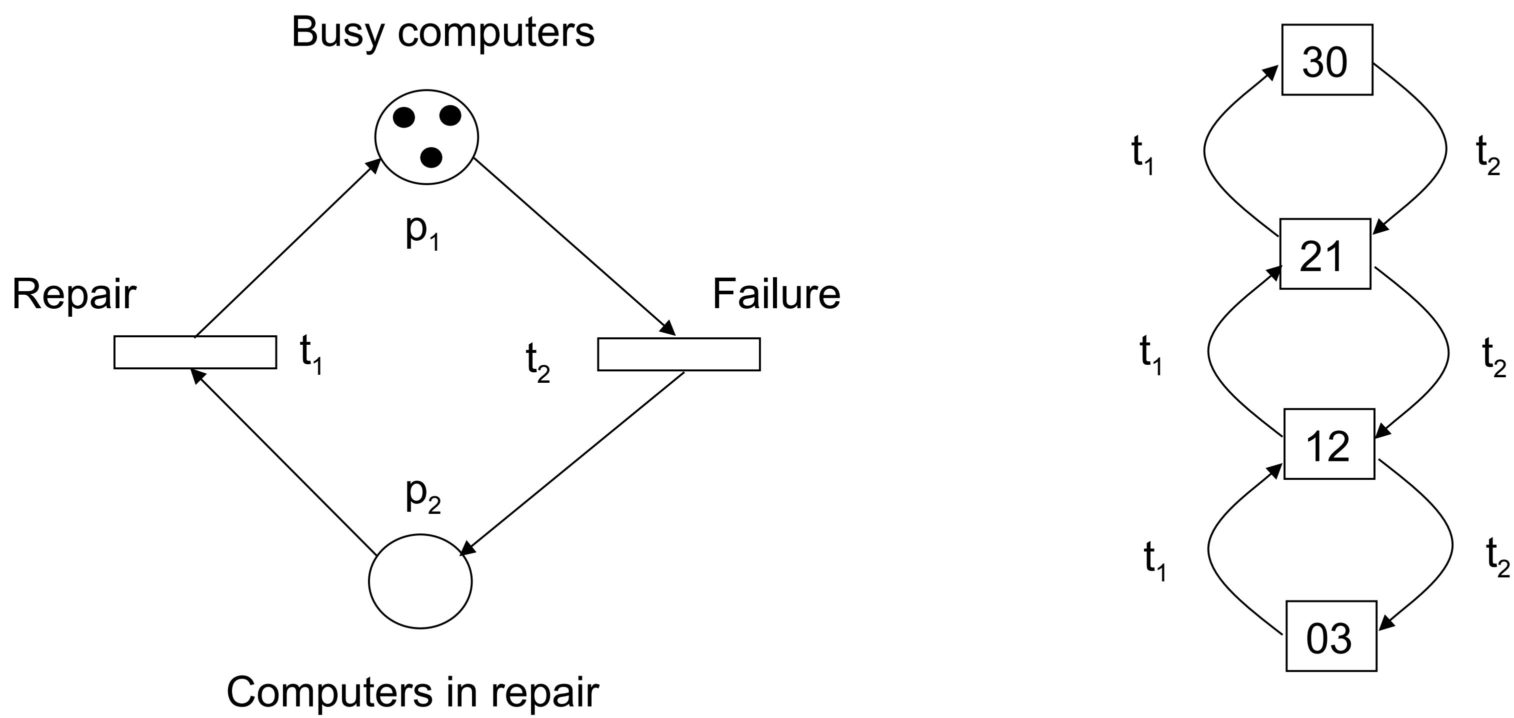
t_2 $\lambda_2(M) = M(p_1) \cdot \boldsymbol{\lambda}$

(Each busy computer can fail.)

t_1 $\lambda_1(M) = \boldsymbol{\mu}$

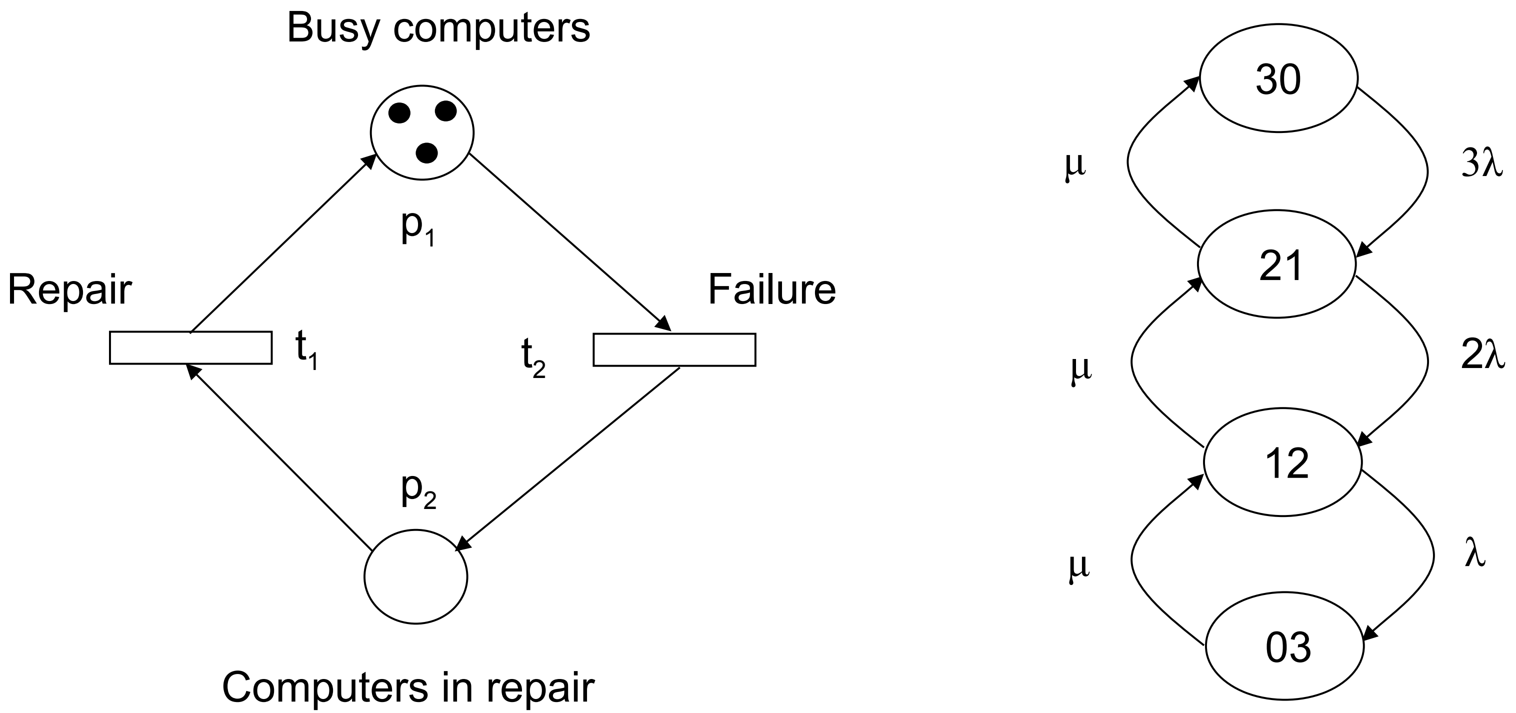
(One computer only can be repaired at any given time instant.)

Stochastic Petri nets



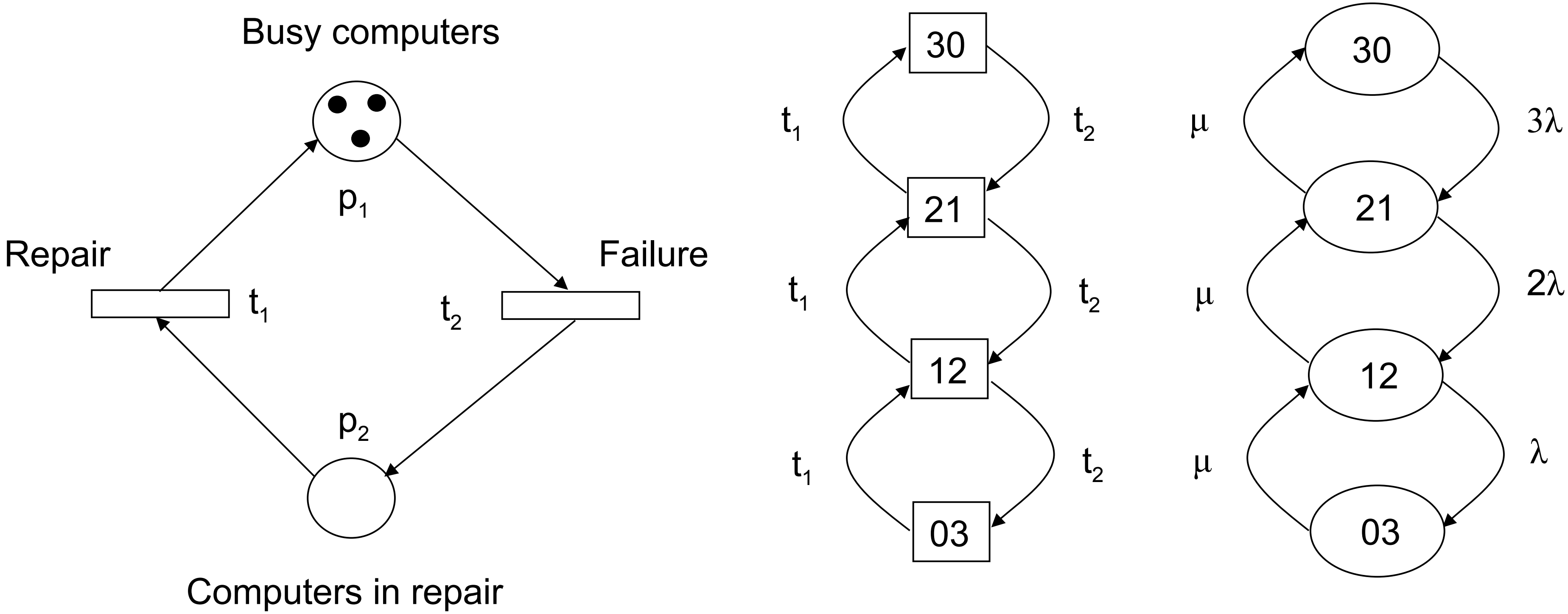
A stochastic Petri net and its reachability graph

Stochastic Petri nets



A stochastic Petri net and its transition intensities diagram

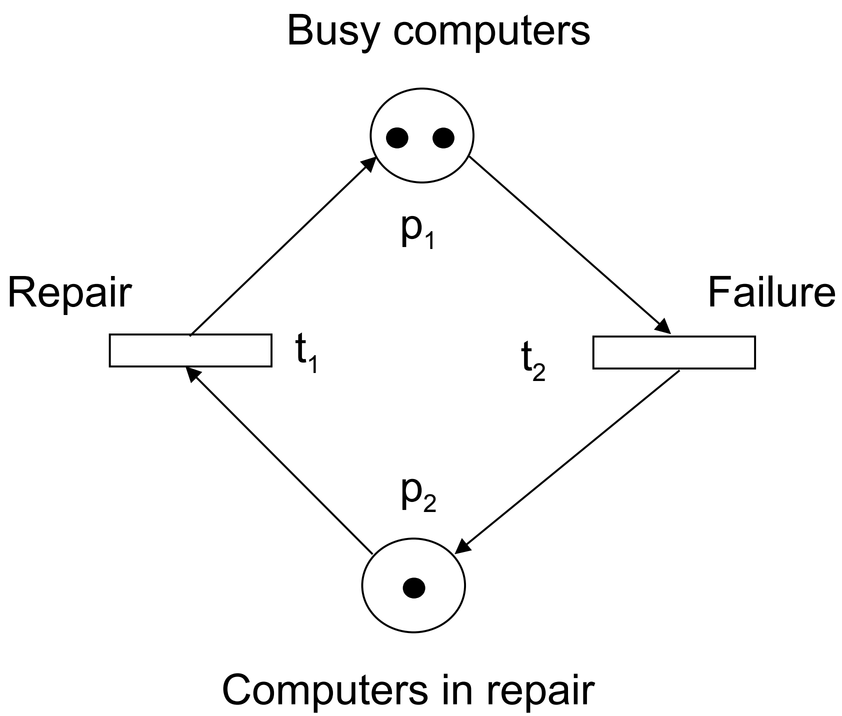
Stochastic Petri nets

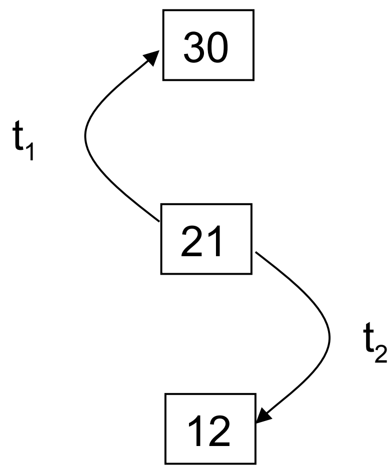


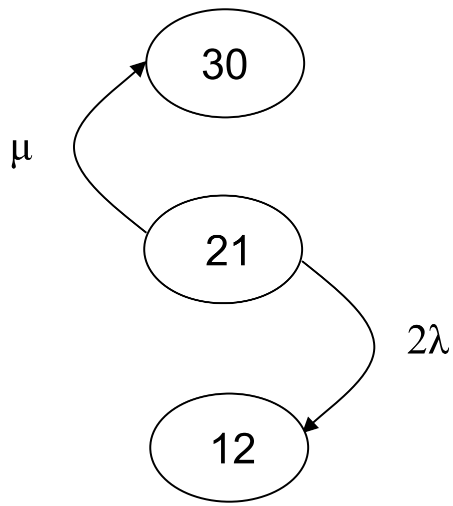
States of the Markov chain are markings of the stochastic Petri net reachability graph

Stochastic Petri nets

Mean firing time of transition t_i in marking M_j is $\dfrac{1}{\lambda_i(M_j)}$







$$\frac{1}{\lambda_1(21)} = \frac{1}{\mu} \qquad \frac{1}{\lambda_2(21)} = \frac{1}{2\lambda}$$

Stochastic Petri nets

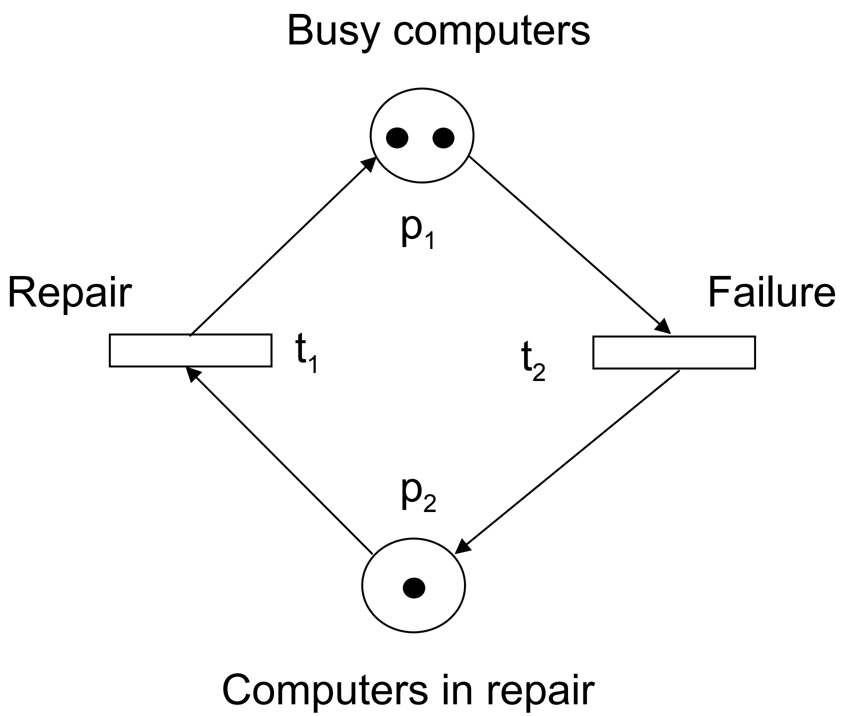
- Mean sojourn time in marking M_j

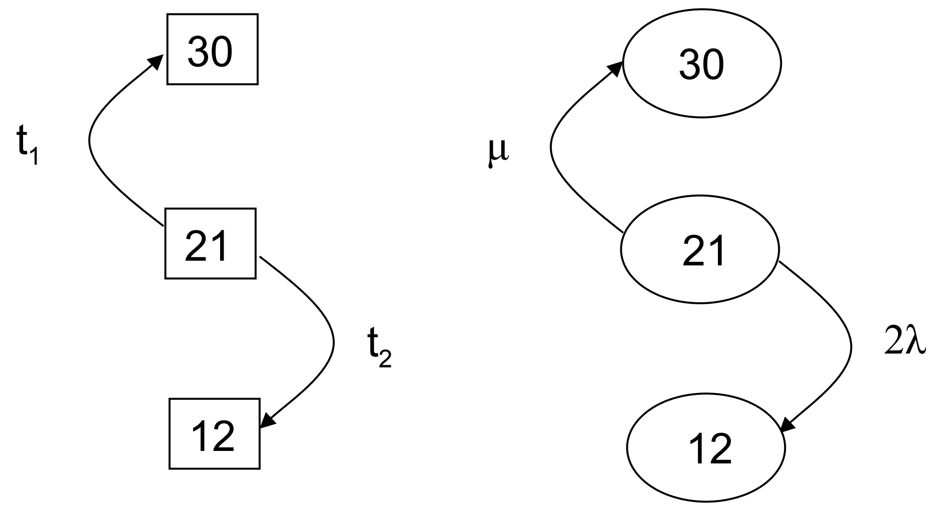
$$\left[\sum_{t_i \in E(M_j)} \lambda_i(M_j) \right]^{-1}$$

$E(M_j)$ - a set of transitions that are enabled in marking M_j

Stochastic Petri nets

Mean sojourn time in marking M=[21]





$$\frac{1}{\lambda_1(21)+\lambda_2(21)}=\frac{1}{\mu+2\lambda}$$

Stochastic Petri nets

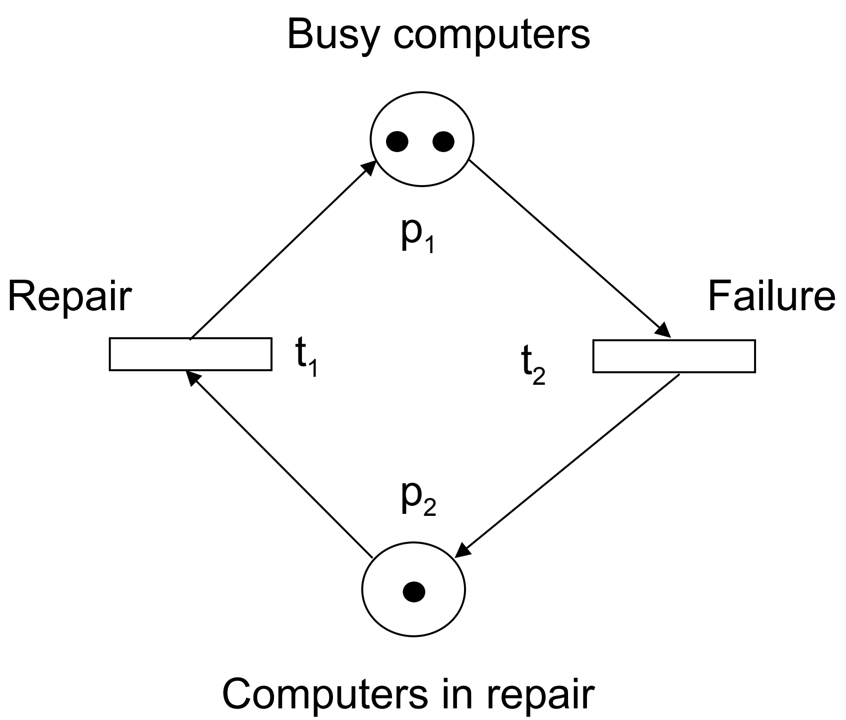
- A firing probability of transition t_k in marking M_j

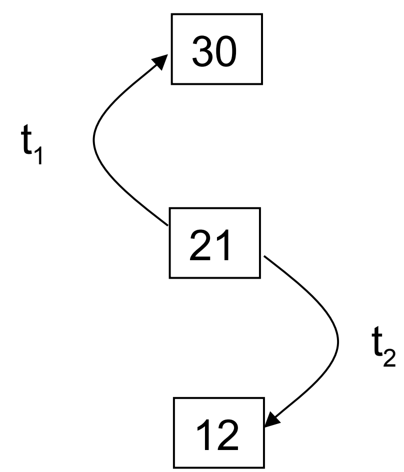
$$P\{t_k \mid M_j\} = \frac{\lambda_k(M_j)}{\sum_{t_i \in E(M_j)} \lambda_i(M_j)}, \qquad t_k \in E(M_j)$$

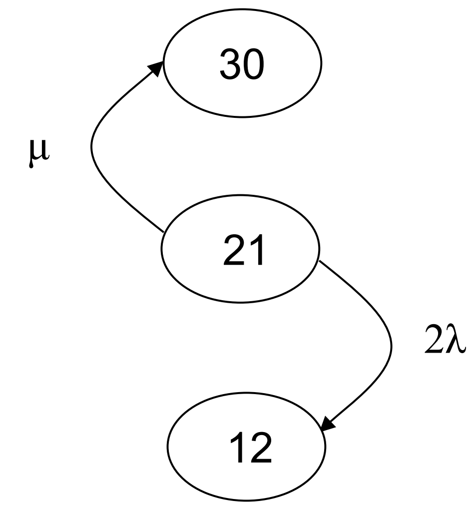
$E(M_j)$ - a set of transitions that are enabled in marking M_j

Stochastic Petri nets

Firing probability of transitions t_1, t_2 in marking M=[21]

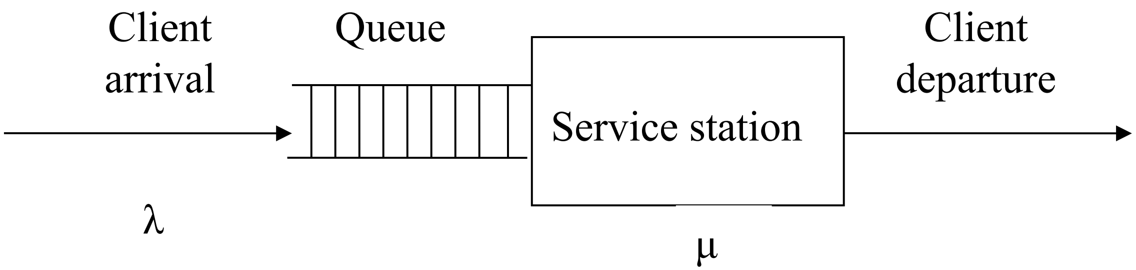




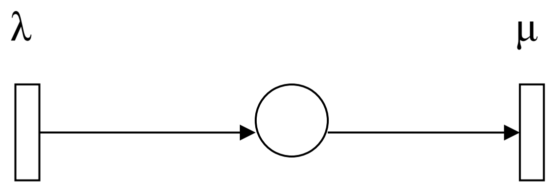


$$P\{t_1|[21]\} = \frac{\lambda_1(21)}{\lambda_1(21)+\lambda_2(21)} = \frac{\mu}{\mu+2\lambda}, \quad P\{t_2|[21]\} = \frac{\lambda_2(21)}{\lambda_1(21)+\lambda_2(21)} = \frac{2\lambda}{\mu+2\lambda}$$

Stochastic Petri nets



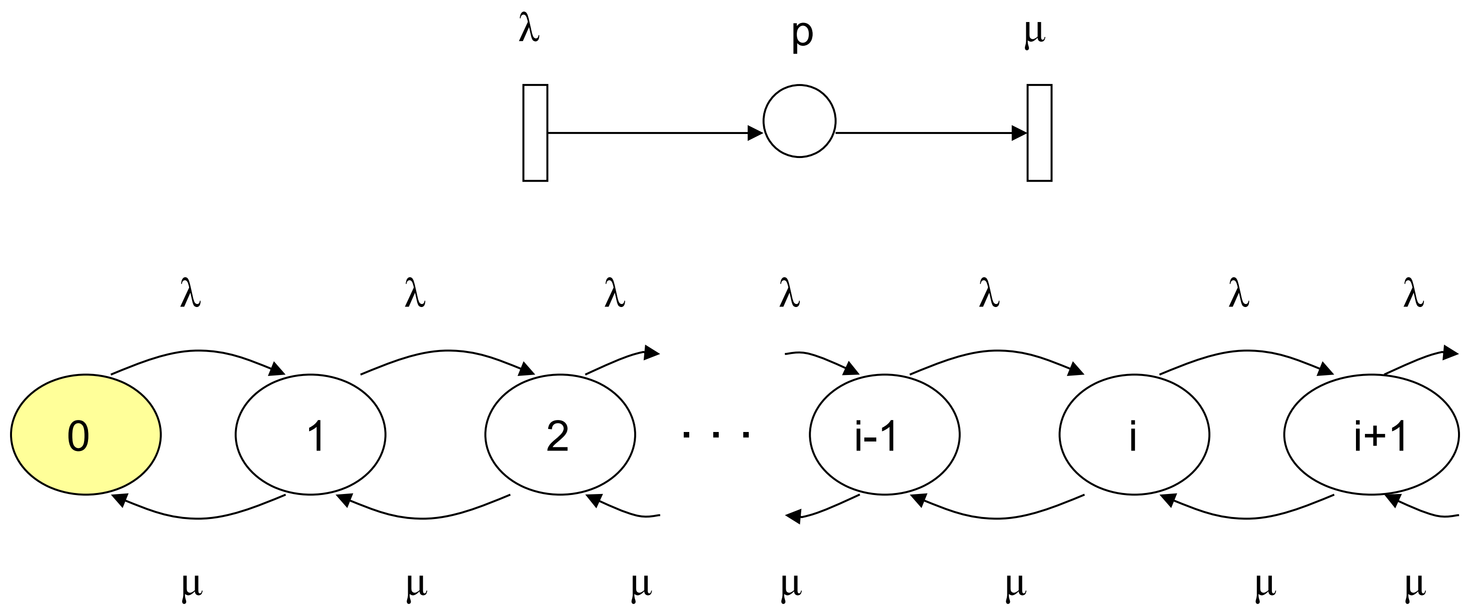
M/M/1 queue



A stochastic Petri net that represents the M/M/1 queue.

Transitions with single server semantics (at each time instant, firing time can elapse for at most one firing).

Stochastic Petri nets



A stochastic Petri net which expresses M/M/1 queue and its transition intensities diagram

Stochastic Petri nets

- Comparison of reachability set of stochastic Petri net $SPN = <P, T, F, M_0, \Lambda>$ and reachability set of Petri net $N = <P, T, F, M_0>$

Let $R(M_0)_{SPN}$ and $R(M_0)_N$ be reachability sets of *SPN* and *N* respectively.

Let $E(M_j)_{SPN}$ and $E(M_j)_N$ be sets of transitions that are enabled in marking M_j of SPN and N respectively.

$$(\forall M_j \in R(M_0)_N)(\forall t_k \in E(M_j)_N)(P\{t_k \mid M_j\}_{SPN} > 0)$$

Hence, $R(M_0)_{SPN} = R(M_0)_N$

Stochastic Petri nets

- Disadvantages of stochastic Petri nets:

- In oder to find a stationary state solution, a linear equation system that contains *n* linear equations, where *n* is the number of states of Markov process (number of markings reachable from initial marking), has to be solved.

- Often in real life systems there are two time scales. The first one is associated with long activities, e.g. transmission of messages, service preparing, data base transactions. The second one is connected with short activities, e.g. operating system decisions. In a stochastic Petri net, execution time of both is expressed by exponential random variables. Hence, short and long activities have similar influence on a size of the state space.

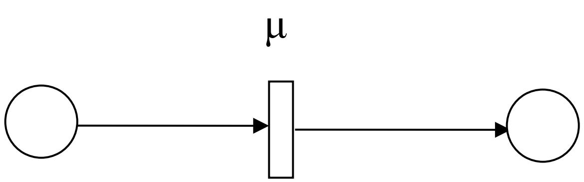
Information Systems Analysis

Jan Magott

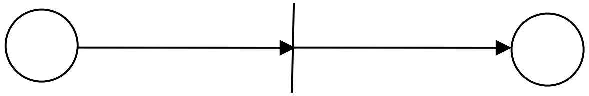
Generalized stochastic Petri nets

Generalized stochastic Petri nets

- In generalized stochastic Petri nets, there are two types of transitions:
- 1. Timed transitions - their firing time is expressed by exponential random variables,

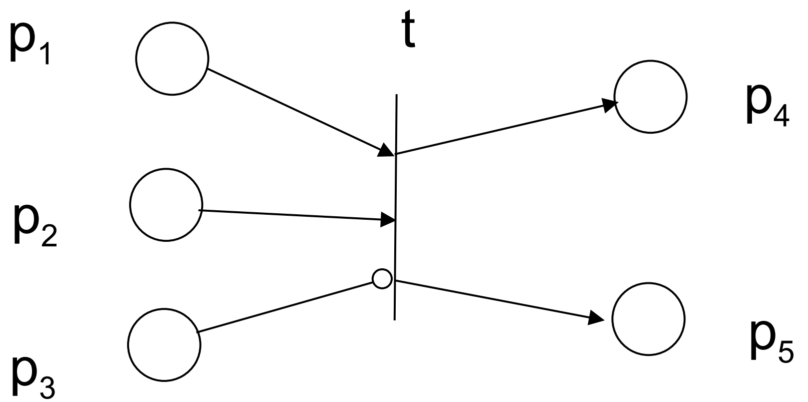


- 2. Immediate transitions – their firing time is equal zero.



Generalized stochastic Petri nets

- Expressive power of Petri net is smaller than expressive power of Turing machines. In order to increase expressive power of Petri nets, inhibitor arcs have been introduced.



- The transition t can be fired provided there is no token in place p_3 .
  The arc < p_3 , t > is called an inhibitor arc.
  „Testing for zero" is expressed this way.

Generalized stochastic Petri nets

- Definition

A *generalized stochastic Petri net* is an 8-tuple:

$$GSPN = < P, T, F, H, Pr, M_0, \Lambda, W >$$

P, T, F, M_0 have similar meaning as in definition of stochastic Petri net,

$H \subset P \times T$ – a set of inhibitor arcs,

$Pr : T \to \{0,1,2,\ldots\}$ – a priority function:

$Pr(t) = 0$ – for timed transitions $t \in T_t$ (lowest priority),

$Pr(t) > 0$ – for immediate transitions $t \in T_i$

$\Lambda = < \lambda_1, \ldots, \lambda_i, \ldots, \lambda_{|T_t|} >$ – a timed transition firing intensity vector,

$W = < w_1, \ldots, w_i, \ldots, w_{|T_i|} >$ – an immediate transition weight vector.

Generalized stochastic Petri nets

- Priorities of immediate transitions are higher than priorities of timed transitions. Hence, immediate transitions are fired first. If there are no enabled immediate transition then enabled timed transition can be fired. Immediate transitions are fired according to their priorities.

- Timed transitions are fired in the similar way transitions of stochastic Petri nets are fired.

- Weights are used in calculations of immediate transitions firing probabilities for transtions with the same priority.

t_k

Generalized stochastic Petri nets

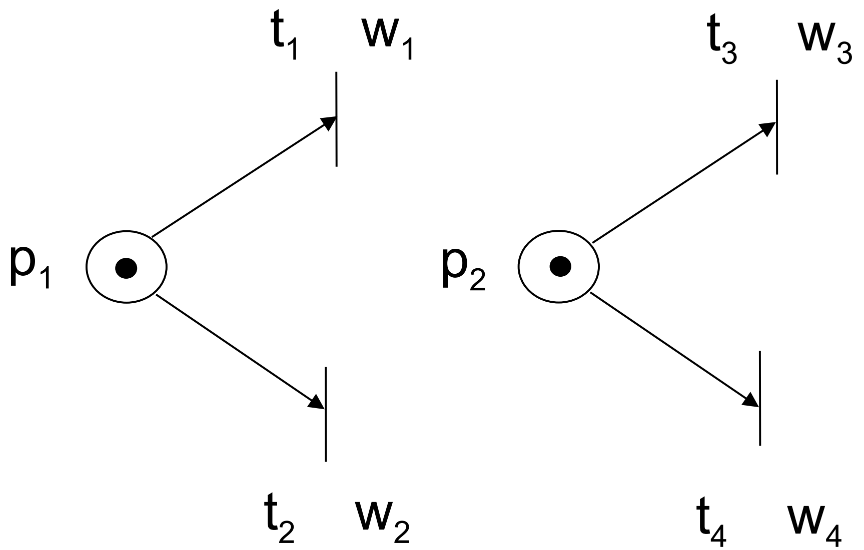
- A firing probability of immediate transition t_k in marking M_j

$$P\{t_k \mid M_j\} = \frac{w_k(M_j)}{\sum_{t_i \in E(M_j)} w_i(M_j)}, \qquad t_k \in E(M_j)$$

$E(M_j)$ - a set of immediate transitions with the highest priority from the transitions enabled in marking M_j

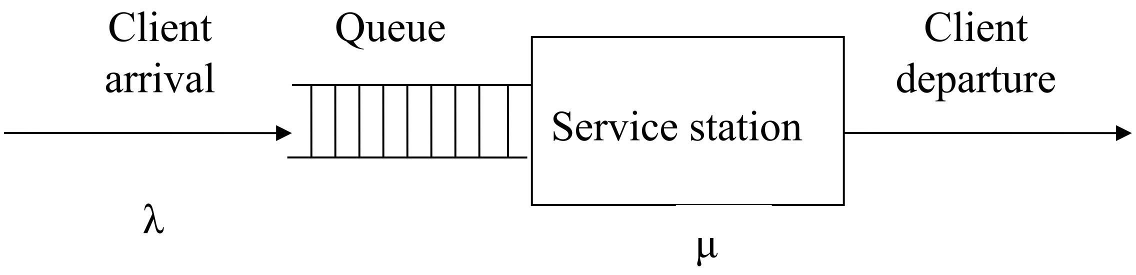
Generalized stochastic Petri nets

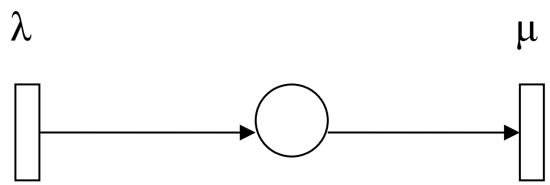
- A firing probability of immediate transition t_k, $k \in \{1,2,3,4\}$, in marking [11] provided the transitions have equal priorities



$$p\{t_i \mid [11]\} = \frac{w_i}{\sum_{i=1}^{i=4} w_i}$$

Generalized stochastic Petri nets

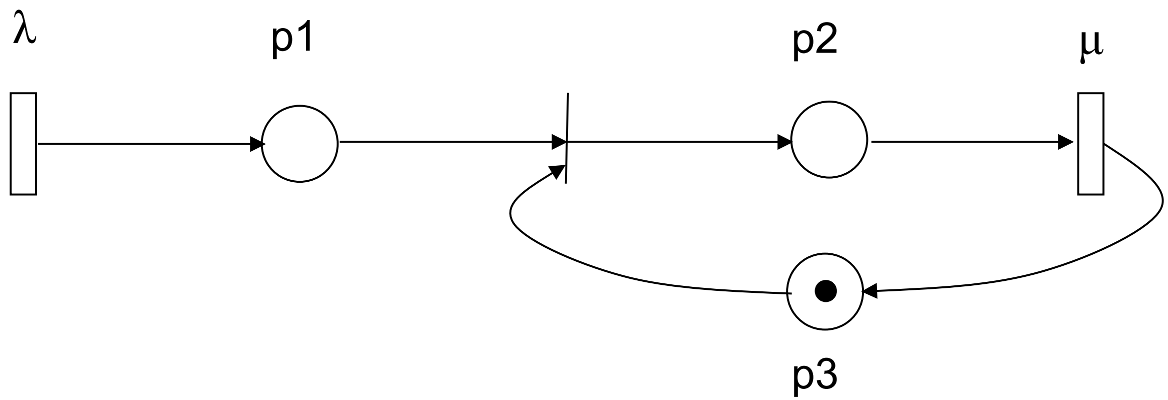




The place contains: a client who is being serviced and clients in the queue

A stochastic Petri net model of queue M/M/1

Generalized stochastic Petri nets



p1 – a queue; a token in this place is a client who is waiting on service
p2 – a service; a token in this place is a client who is being serviced
p3 – an idle station; a token in this place denotes that the station is idle
λ – a parameter of arrival Poisson process
μ – a parameter of service process

A generalized stochastic Petri net model of M/M/1 queue

Generalized stochastic Petri nets

- Queue $M/M/k/m/n$

n – a number of clients (size of the source of clients),

First M – the client whose service has been completed is ready to arrive again to the queue after time described by an exponential random variable,
this time can represent a client's activity which does not require the station,

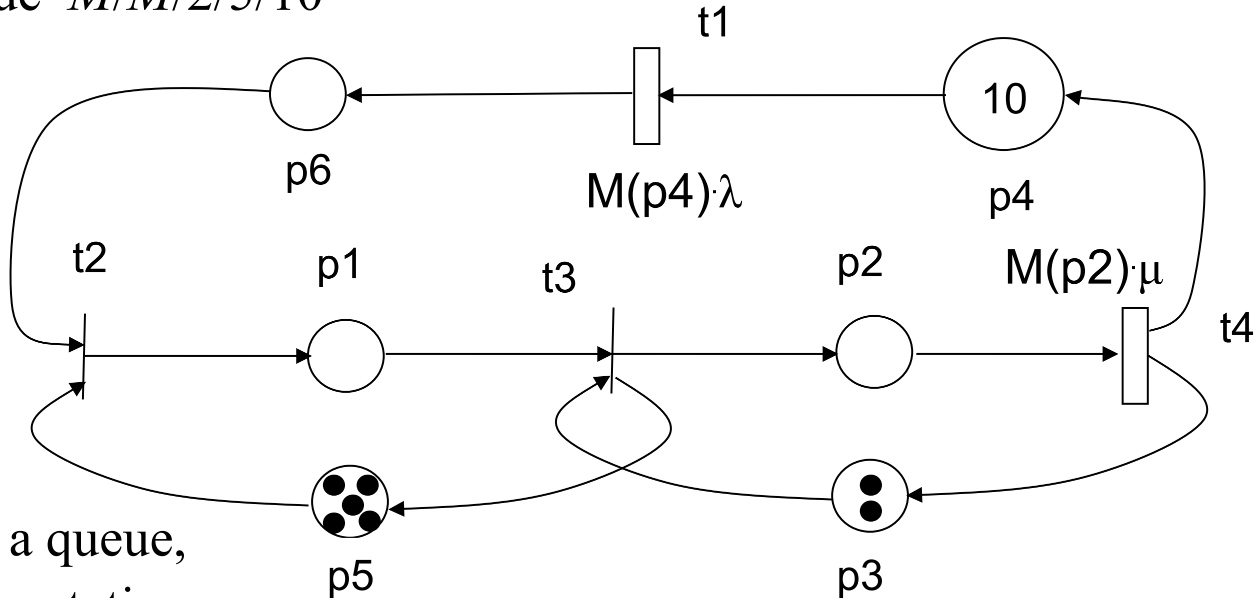
k – a station with k servers,

Second M – a service process at the server lasts time expressed by an exponential random variable,

m – a number of positions in the queue,

Generalized stochastic Petri nets

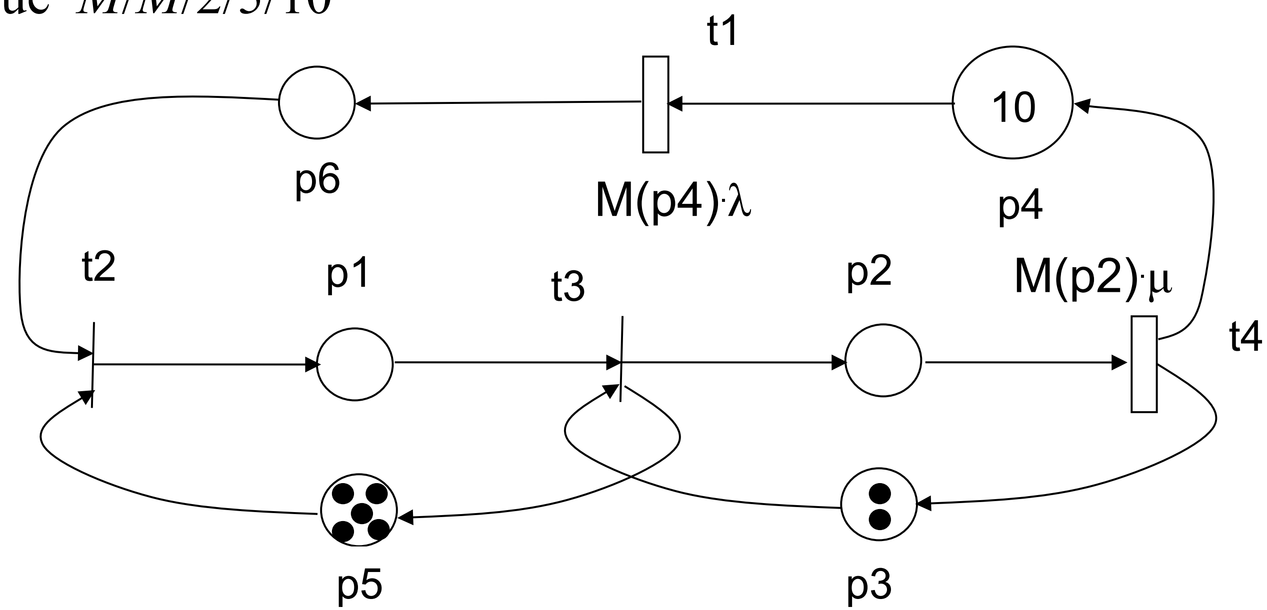
- Queue *M*/*M*/2/5/10



- p1 – a queue,
- p2 – a station,
- p3 – idle servers,
- p4 – a client activity,
- p5 – free places in the queue,
- p6 – the client after its activity and before arriving into the queue.

Generalized stochastic Petri nets

- Queue *M*/*M*/2/5/10



M(p2)·μ – a transition firing intensity of transition t4, where M(p2) is the number of servers of the service station that are busy,

M(p4)·λ – a transition firing intensity of transition t1, where M(p4) is the number of clients executing their activity.

Generalized stochastic Petri nets

- A reachability set of generalized stochastic Petri net (GSPN) is divided into two classes:

- Tangible markings (only timed transitions can be enabled),

- Vanishing markings (at least one immediate transition is enabled).
  Sojourn time in a vanishing marking is equal to zero because immediate transitions are fired before timed ones, and firing time of immediate transitions is equal to zero.

Generalized stochastic Petri nets
Transition firing

- In vanishing markings (at least one immediate transition is enabled).
  1. Enabled transitions with highest priority are selected,
  2. Transitions from $E(M_j)$ - a set of immediate transitions with the highest priority from transitions enabled in marking M_j are fired

$$P\{t_k \mid M_j\} = \frac{w_k(M_j)}{\sum_{t_i \in E(M_j)} w_i(M_j)}, \qquad t_k \in E(M_j)$$

- In tangible markings (only timed transitions can be enabled).

  Enabled timed transitions are fired as for stochastic Petri nets.

  Transitions are not fired simultaneously.

Generalized stochastic Petri nets

- A transition probability matrix A of *GSPN* with finite reachability set

Kv – a number of vanishing markings

Kt – a number of tangible markings

$$C = \left[c_{ij} \right]_{Kv \times Kv}$$

c_{ij} - a transition probability from the *i*-th vanishing marking to the *j*-th vanishing marking

$$D = \left[d_{ij} \right]_{Kv \times Kt}$$

d_{ij} - a transition probability from the *i*-th vanishing marking to the *j*-th tangible marking

Generalized stochastic Petri nets

- A transition probability matrix A of *GSPN* with finite reachability set

Kv – a number of vanishing markings
Kt – a number of tangible markings

$$C = \left[c_{ij} \right]_{Kv \times Kv} \qquad D = \left[d_{ij} \right]_{Kv \times Kt} \qquad E = \left[e_{ij} \right]_{Kt \times Kv} \qquad F = \left[f_{ij} \right]_{Kt \times Kt}$$

$$A = \begin{bmatrix} C & D \\ E & F \end{bmatrix}$$

$$A = \left[a_{ij} \right]_{(Kv + Kt) \times (Kv + Kt)}$$

Generalized stochastic Petri nets

A transition probability matrix A of *GSPN* with finite reachability set

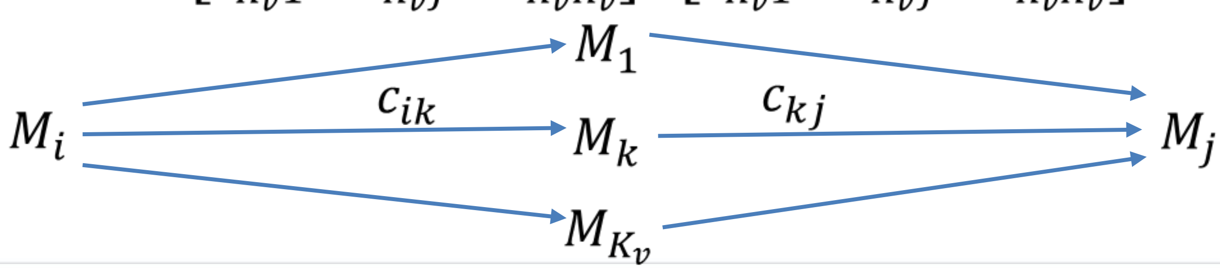
Kv – a number of vanishing markings

Kt – a number of tangible markings

$$C = \left[c_{ij}\right]_{K_v \times K_v}$$

c_{ij} - transition probability from vanishing marking M_i to vanishing marking M_j

$$C^2 = C \cdot C = \begin{bmatrix} c_{11} & c_{1j} & c_{1K_v} \\ c_{i1} & c_{ij} & c_{iK_v} \\ c_{K_v1} & c_{K_vj} & c_{K_vK_v} \end{bmatrix} \cdot \begin{bmatrix} c_{11} & c_{1j} & c_{1K_v} \\ c_{i1} & c_{ij} & c_{iK_v} \\ c_{K_v1} & c_{K_vj} & c_{K_vK_v} \end{bmatrix} = \left[p_{ij}\right] = \left[\sum_{k=1}^{K_v} c_{ik} \cdot c_{kj}\right]$$



Generalized stochastic Petri nets

- In order to reduce computational complexity, we want to find the transition probability matrix between tangible markings only, however, taking into account transitions through vanishing markings.

w_{ij} - a transition probability from the *i*-th tangible state to the *j*-th tangible state provided transitions through vanishing markings are taken into account.

$$w_{ij} = f_{ij} + \sum_{v \in V} e_{iv} \cdot p_{vj}$$

V – a set of vanishing markings,

f_{ij} – a transition probability from the *i*-th tangible marking to the *j*-th tangible one,

e_{iv} – a transition probability from the *i*-th tangible marking to *v*-th vanishing one,

p_{vj} – a transition probability from the *v*-th vanishing marking to the *j*-th tangible one, in any number of transitions through vanishing markings only.

Generalized stochastic Petri nets

$$C = \left[c_{ij} \right]_{Kv \times Kv} \qquad D = \left[d_{ij} \right]_{Kv \times Kt}$$

$$G = \sum_{h=0}^{k-1} C^h D \qquad G = \left[g_{ij} \right]_{Kv \times Kt}$$

g_{ij} - a transition probability from the *i*-th vanishing marking to the *j*-th tangible marking by trajectories of length not greater than *k* transition firings provided only vanishing markings are reachable before the *j*-th tangible marking

Generalized stochastic Petri nets

- Assumption
  There are no loops in the set of vanishing states.

Hence, $(\exists k_0 < Kv)(k_0 < k)(C^k = 0)$

Finally: $$W = F + E \sum_{h=0}^{k_0} C^h D$$

where

w_{ij} - a transition probability from the *i*-th tangible state to the *j*-th tangible state provided transitions through vanishing markings are taken into account,

$C = \left[c_{ij} \right]_{Kv \times Kv}$ $D = \left[d_{ij} \right]_{Kv \times Kt}$ $E = \left[e_{ij} \right]_{Kt \times Kv}$ $F = \left[f_{ij} \right]_{Kt \times Kt}$

$$W = \left[w_{ij} \right]_{Kt \times Kt}$$

Generalized stochastic Petri nets

Transient behaviour

Definition

A *final marking* is a marking that there are no transitions from it.

Assumption

There is a single final marking with index *f*.

$\overline{W}$ - a matrix obtained from the matrix W by removing *f*-th row and *f*-th column.

$$\overline{W} = \left[\, \overline{w}\, ij \right]_{(Kt-1)\times(Kt-1)}$$

Matrix multiplication $\overline{W}^{\,n} = \overline{W}^{\,n-1} \cdot \overline{W}$

Generalized stochastic Petri nets

Transient behaviour

$$\overline{W}^{n} = \overline{W}^{n-1} \cdot \overline{W}$$

$$\overline{W}^{n} = \left[w'_{ij} \right]_{(Kt-1)\times(Kt-1)}$$

w'_{ij} - a mean number of occurrences of the j-th marking on (n+1)-position in a sequence of $n+1$ markings provided the i-th marking is the intial one.

$$Z = \left[z_{ij} \right]_{(Kt-1)\times(Kt-1)}$$

z_{ij} - a mean number of occurrences of the j-th marking on all positions in a sequence of markings provided the i-th marking is the intial one.

Hence $Z = \sum_{n=0}^{\infty} \overline{W}^{n}$ where $\overline{W}^{0} = I = \begin{bmatrix} 1\,0\,0 \ldots 0 \\ 0\,1\,0 \ldots 0 \\ 0\,0\,1\,0 \ldots 0 \\ . \quad . \quad . \\ 0 \quad \ldots 0\,1 \end{bmatrix}_{(Kt-1)\times(Kt-1)}$

Generalized stochastic Petri nets

Transient behaviour

$$Z = \sum_{n=0}^{\infty} \overline{W}^{n} = I + \overline{W} + \overline{W}^{2} + \ldots..$$

The above matrix series is equal to $Z = (I - \overline{W})^{-1}$

Mean sojourn time in the *j*-th marking M_j

$$ST(M_j) = \left[\sum_{t_i \in E(M_j)} \lambda_i(M_j) \right]^{-1}$$

$E(M_j)$ - a set of timed transitions enabled in the tangible marking M_j

Mean time of reaching the final f-th marking provided the initial marking is i-th one is:

$$\tau_{if} = \sum_{k=1, k \neq f}^{K_t} z_{ik} \cdot ST(M_k)$$

Generalized stochastic Petri nets

Cyclic behaviour

- A stationary solution is obtained by solving the following linear equation system:

$$\Pi \cdot W = 0$$

where

W - a transition probability matrix from tangible marking to tangible one taking into account transitions through vanishing markings

$$\Pi = [\pi_1, \ldots, \pi_i, \ldots, \pi_{Kt}]$$

Kt - number of tangible markings

$$\sum_{i=1}^{i=Kt} \pi_i = 1$$

Generalized stochastic Petri nets
Cyclic behaviour

- Mean ratio of sojourn time in tangible marking M_j is

$$\upsilon_j = \frac{ST(M_j) \cdot \pi_j}{\sum_{k=1}^{K_t} ST(M_k) \cdot \pi_k}$$

$ST(M_j)$ - mean sojourn time in the tangible marking M_j

π_j - an entry of vector Π which is the solution of $\Pi \cdot W = 0$

Generalized stochastic Petri nets

Cyclic behaviour

- In order to compute mean cycle time, a marking is selected as the reference one.

  Let M_k be the reference marking.

  Mean number of transitions through marking M_j between two subsequent transitions through the reference marking M_k

$$q_{jk} = \frac{\pi_j}{\pi_k}$$

Mean cycle time

$$\gamma_k = \sum_{j=1}^{K_t} q_{jk} \cdot ST(M_j)$$

Generalized stochastic Petri nets
Performance metrics

- π_j – a probability that in stationary state, *GSPN* is in marking M_j

π_j is very detailed information. Usually, one is interested in more general metrics obtained by aggregation of many states.

Examples

Event that in stationary state there are no tokens in given subset of places.
Event that in stationary state there exists at least one token in given place.
Event that in stationary state there are exactly *k* tokens in given place.

Generalized stochastic Petri nets
Performance metrics

- A probability Π_k of an event that there are no tokens in a subset $P_k \subset P$ of all places.

- M - a set of such markings that there are no tokens in a subset $P_k \subset P$ of all places.

$$M = \{M_j \mid M_j \in R(M_0) \wedge (\forall p_i \in P_k)(M_j(p_i) = 0)\}$$

$$\Pi_k = \sum_{M_j \in M} \pi_j$$

Generalized stochastic Petri nets

- Comparison of reachability sets of a generalized stochastic Petri net $GSPN = \langle P, T, F, H, Pr, M_0, \Lambda, W \rangle$ and of a Petri net $N = \langle P, T, F, M_0 \rangle$

In *GSPN* there are inhibitor arcs and a priority relation.
Hence:

$$R(M_0)_{GSPN} \subset R(M_0)_N$$