

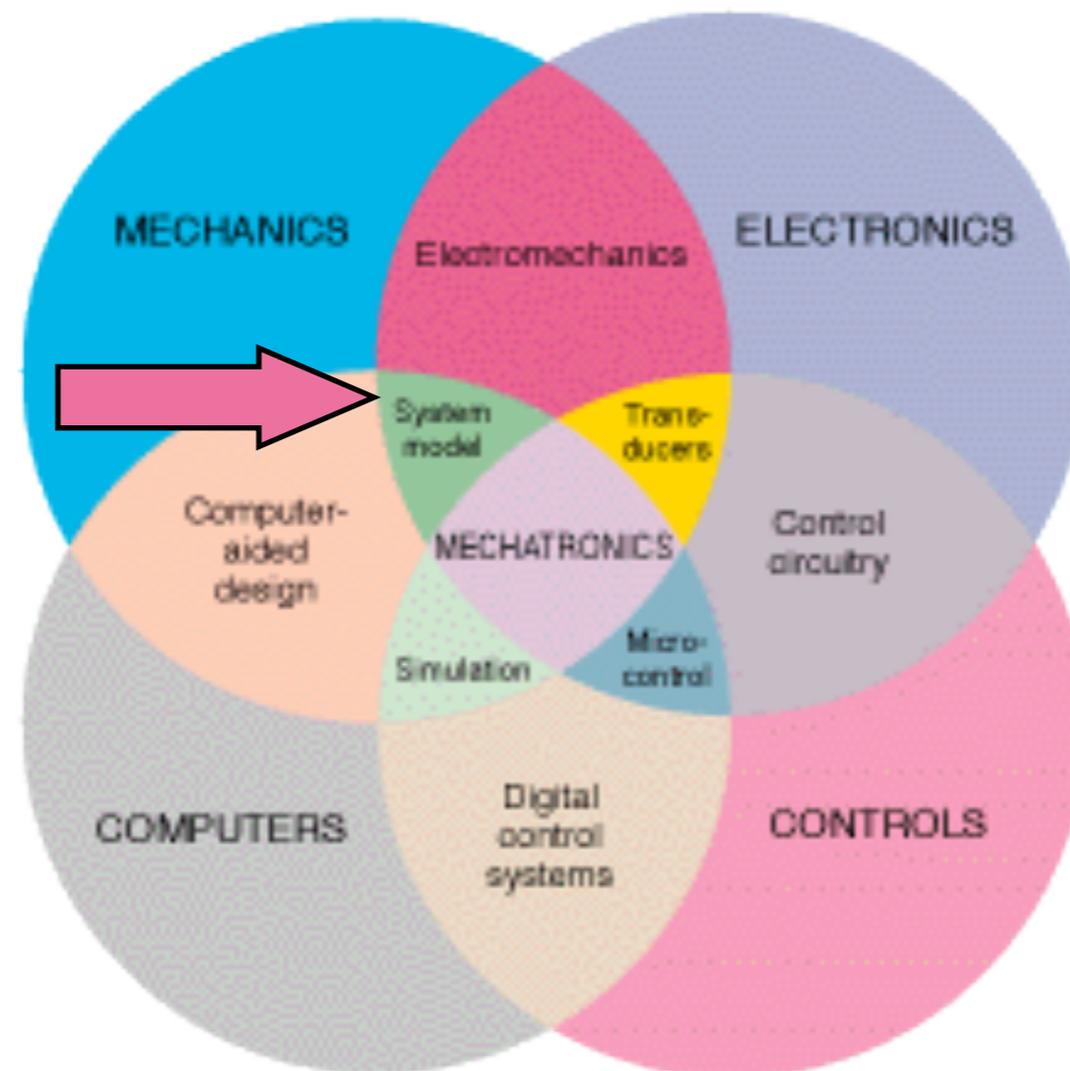
# *PMR 5237 Modelagem e Design de Sistemas Discretos em Redes de Petri*

---

## Aula I: Introdução

Prof. José Reinaldo Silva  
[reinaldo@usp.br](mailto:reinaldo@usp.br)

# Contexto da Disciplina

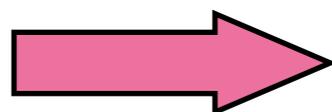


# Histórico das RdP

Criada em 1962 com a tese de doutorado de Carl Adam Petri  
(*Communication with Automata*)



Carl Adam Petri nasceu em Leipzig, em 12 de julho de 1926 e morreu aos 83 anos em 2 de Julho de 2010, deixando um dos maiores legados teórico-científicos do século XX, desenvolvido em pouco mais de 40 anos.



*Schemas*

A schema is a cognitive framework or concept that helps organize and interpret information. Schemas can be useful because they allow us to take shortcuts in interpreting a vast amount of information



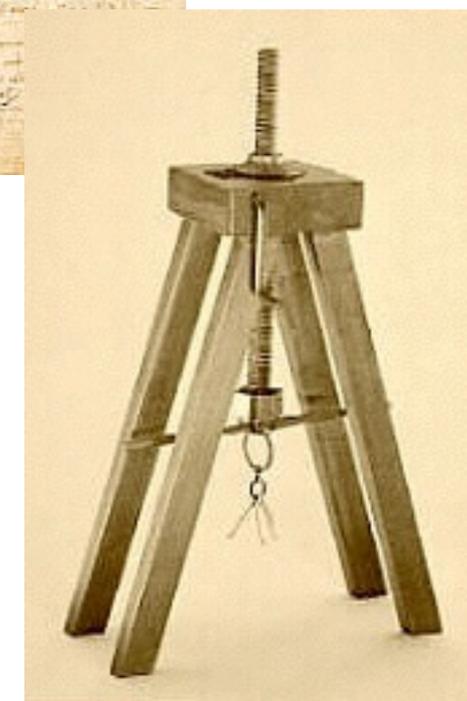
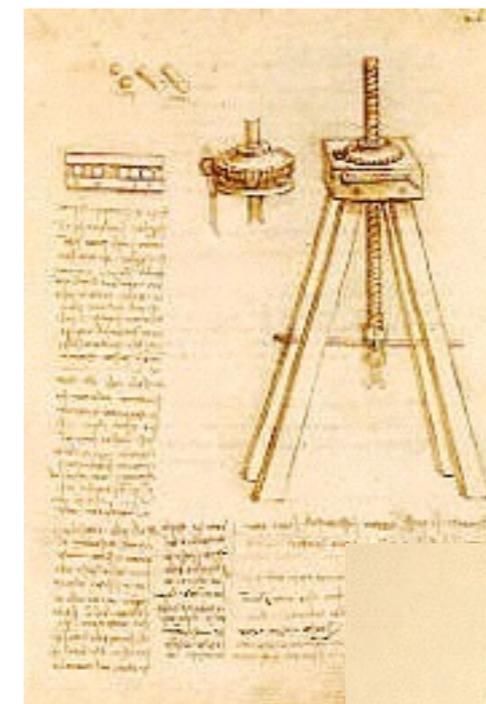
modelagem de processos

2001 UML 2.0 modelagem de requisitos

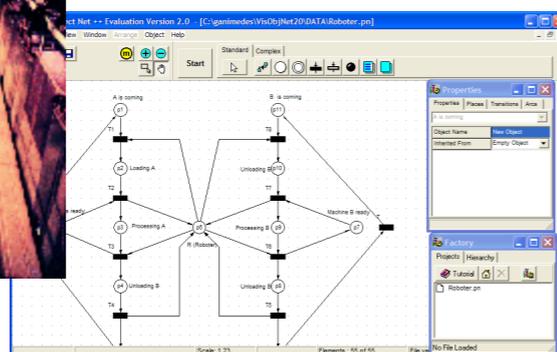
modelagem de workflow

# A “arte” de modelar

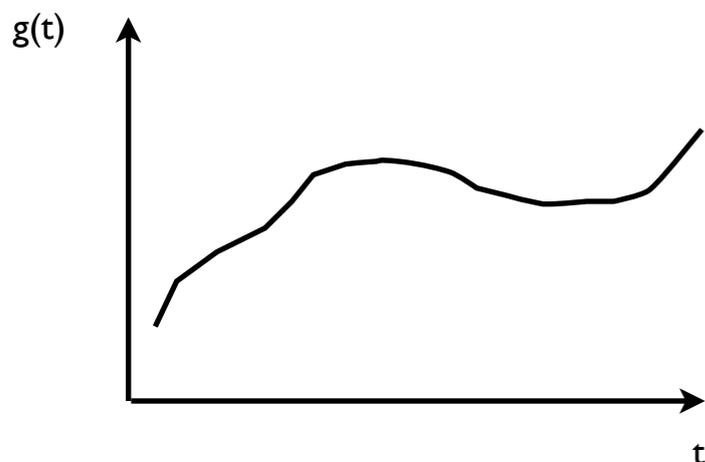
Modelar é produzir uma representação esquemática de um objeto ou sistema capturando suas propriedades mais importantes e a relação entre as partes dinâmicas, geralmente de modo a compactar as informações, aumentar o nível de abstração. Em Engenharia, a maior parte das vezes um modelo é o ponto de partida para a análise de um artefato que pode ser feita analiticamente ou através de simulação.



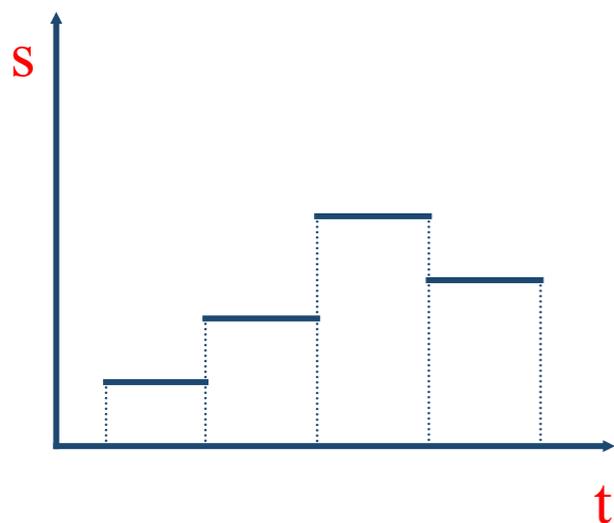
Modelo de Leonardo da Vinci



# Paradigmas para modelagem: Estado/ Transição



A descrição do comportamento de sistemas é feita identificando uma função das chamadas “variáveis de estado”, ou variáveis que caracterizam o sistema. Este tem pontos de equilíbrio que caracterizam o estado e evoluem no tempo denotando a característica do sistema. A evolução se dá quando “eventos” ocorrem.

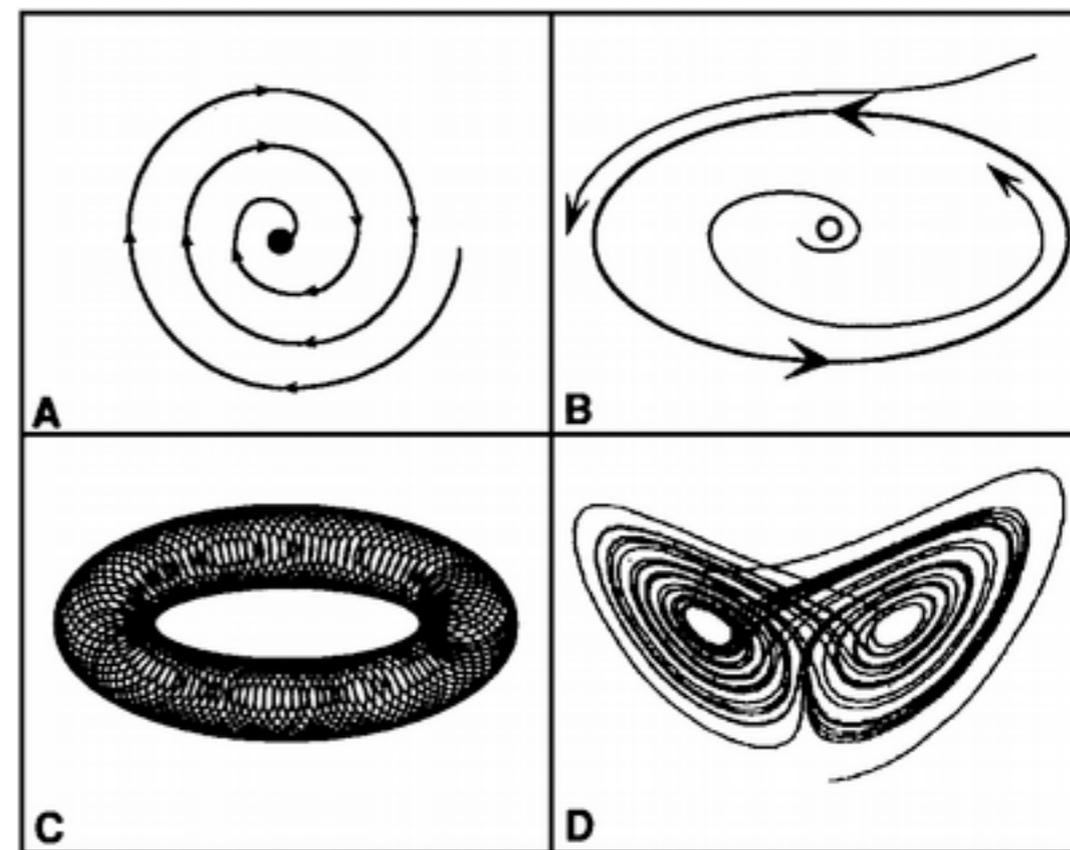


Em alguns casos a função de estado é contínua nas variáveis de estado e também no tempo caracterizando trajetórias também contínuas. Em outros casos a função de estado é discreta e evolui aos saltos caracterizando um sistema discreto.

# Diagrama de Eventos

Vamos agora fazer uma inversão e imaginar que temos um sistema dinâmico  $f(x)$  e em um diagrama 2D onde cada ponto representa um evento. Aplicando-se cada um destes eventos a  $f(x)$  temos que o sistema muda de estado e está preparado para receber outro evento. A evolução do sistema evento a eventos (não necessariamente ligada ao tempo, mas somente à sequência admissível de eventos) também dá uma trajetória no espaço de eventos.

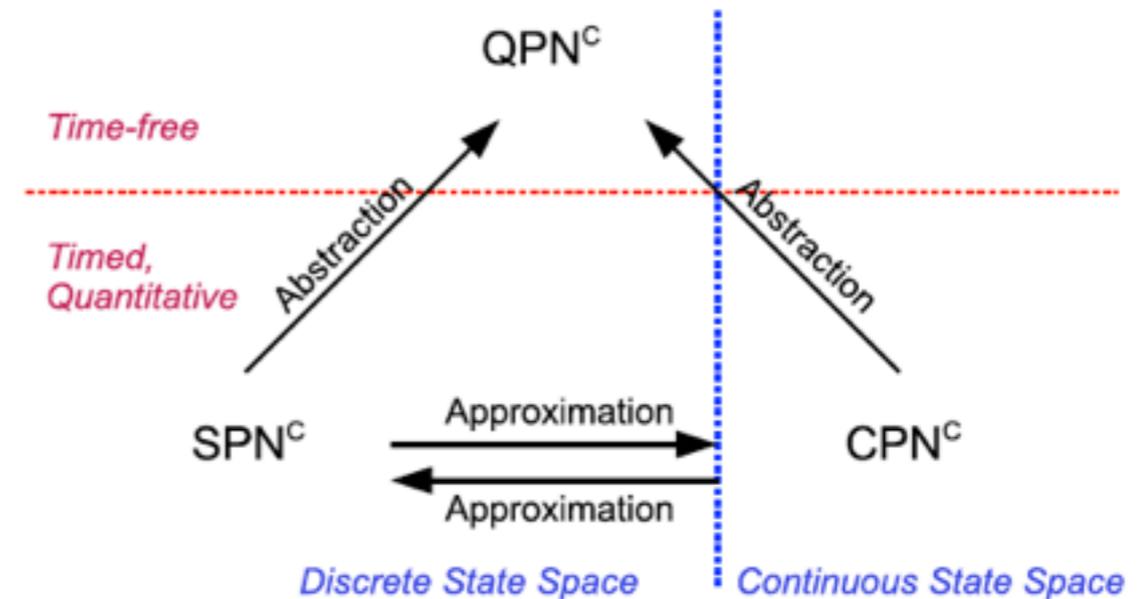
Eventualmente esta trajetória pode convergir para valores (sem necessariamente atingí-los) que são chamados atratores.



# Novas Aplicações das Redes de Petri

Hoje as redes de Petri são aplicadas em um universo ainda mais amplo e muito mais abrangente do que a comunicação entre processos, redes de computadores, etc.

As novas aplicações mais abrangentes são feitas em sistemas complexos que podem ser associados, por exemplo a sistemas complexos de manufatura, “*harzardous systms*”, sistemas de automação mais complexos e sistemas biológicos.



Liu, F. and Heiner, M.; Colored Petri Nets to Model and Simulate Biological Systems, Recent Advances in Petri Nets and Concurrency, S. Donatelli, J. Kleijn, R.J. Machado, J.M. Fernandes (eds.), CEUR Workshop Proceedings, volume 827, ISSN 1613-0073, Jan/2012, pp. 71-85.

# Aplicações históricas e Tradicionais das RdP

Além da aplicação clássica em sistemas dinâmicos as Redes de Petri foram historicamente aplicadas em Engenharia de Software desde os anos 80 com Gerald Estrin na UCLA e Stephen Yau em Illinois Urbana-Champaign. Hoje o congresso mais importante de Redes de Petri no mundo, o 33th Int. Conference on Application and Theory of Petri Nets and Cuncurrency (realizada em conjunto com a 12th Int. Conference on Application of Concurrency to Systems Design), tem um workshop (desde 2008) dedicado especificamente a Petri Nets on Software Engineering, o PNSE. Ao lado está o Call for Papers desde workshop em 2012.

Outra aplicação que se tornou clássica é em Redes de Computadores, especialmente o roteamento inteligente praticado no projeto Superhighway.

**PNSE'12**

**International Workshop on Petri Nets and Software Engineering**

**Hamburg, Germany, June 25-26, 2012**

a satellite event of Petri Nets 2012 and ACS2 2012  
10th INTERNATIONAL CONFERENCE ON APPLICATION AND THEORY OF PETRI NETS AND CONCURRENCY  
 and  
 12th INTERNATIONAL CONFERENCE ON APPLICATION OF CONCURRENCY TO SYSTEMS DESIGN

More information: <http://www.infomark.uni-hamburg.de/TOC/evnts/pnse12/>  
 Contact e-mail: [pnse12\\_at\\_infomark\\_at\\_uni-hamburg\\_de](mailto:pnse12_at_infomark_at_uni-hamburg_de)

---

**Important Dates**

Deadline for full papers: March 26th, 2012  
 Deadline for short papers: March 26th, 2012  
 Notification of paper acceptance: May 1st, 2012  
 Deadline for posters: May 18th, 2012  
 Notification of poster acceptance: May 17th, 2012  
 Deadline for final revisions: June 1st, 2012  
 Workshop: Monday-Tuesday, June 25/26, 2012

---

**Scope**

For the successful realization of complex systems of interacting and reactive software and hardware components the use of a precise language at different stages of the development process is of crucial importance. Petri nets are becoming increasingly popular in this area, as they provide a uniform language supporting the tasks of modelling, validation, and verification. Their popularity is due to the fact that Petri nets capture fundamental aspects of causality, concurrency and choice in a natural and mathematically precise way without compromising readability.

The workshop PNSE'12 (Petri nets and Software Engineering) will take place as a satellite event of Petri Nets 2012.

The use of Petri nets (PT-nets, coloured Petri nets and extensions) in the formal process of software engineering, covering modelling, validation, and verification, will be presented as well as their application and tools supporting the disciplines mentioned above.

---

**Topics**

We welcome contributions describing original research in topics related to Petri nets in combination with software engineering, addressing open problems or presenting new ideas regarding the relation of Petri nets and software engineering. Furthermore we look for surveys addressing open problems and new applications of Petri nets. Topics of interest include but are not limited to:

- **Modelling**
  - representation of formal models by intuitive modeling concepts
  - guidelines for the construction of system models
  - representative examples
  - process-, service-, state-, event-, object- and agent-oriented approaches
  - adaptation, integration, and enhancement of concepts from other disciplines
  - views and abstractions of systems
  - model-driven architecture
  - modelling software landscapes
  - web service-based software development
- **Validation and Execution**
  - prototyping
  - simulation, observation, animation
  - code generation and execution
  - testing and debugging
  - efficient implementation

- **Verification**
  - structural methods (e.g. place invariants, reduction rules)
  - results for structural subclasses of nets
  - relations between structure and behaviour
  - state space based approaches
  - efficient model checking
  - assertional and deductive methods (e.g. temporal logics)
  - process algebraic methods
  - applications of category theory and linear logic
- **Application of Petri nets in Software Engineering, in particular the use of Petri nets in the domains of**
  - flexible manufacturing,
  - logistics,
  - telecommunication,
  - workflow management and
  - embedded systems.
- **Tools in the fields mentioned above**

---

**Submissions**

The programme committee invites submissions of full contributions (up to 15 pages) or short contributions (up to 5 pages). Ongoing work (up to 2 pages) can also be presented in a special poster session.

Please note that for full contributions up to 15 pages are recommended.

Papers should be submitted in electronic form (PDF) using the Springer LNCS-format (see <http://www.springer.de/comp/lncs/authors.html>). Submissions should include title, authors' addresses, E-mail addresses, keywords and an abstract. For your submission in PDF format please use the online conference management system at

<http://www.easychair.org/conferences/2008/pnse12>

Just create a new account and then upload your paper. (Later you will be able to see your reviews there.)

The papers will be peer reviewed by at least three members of the PC. Accepted contributions will be included in the workshop proceedings, which will be available at the workshop as well as published online.

Some of the best papers from the workshop will be invited for publication in a volume of the journal sub line of Lecture Notes in Computer Science entitled "Transactions on Petri Nets and Other Models of Concurrency (ToPNOC)". The papers are expected to be thoroughly revised and they will go through a timely new round of reviewing as is standard practice for journal papers.

Papers from previous instances of this workshop (PNSE'03, PNSE'08, PNSE'10 and PNSE'11) made it into ToPNOC volumes in the Springer LNCS series (volumes 2100, 5460 and 6900).

---

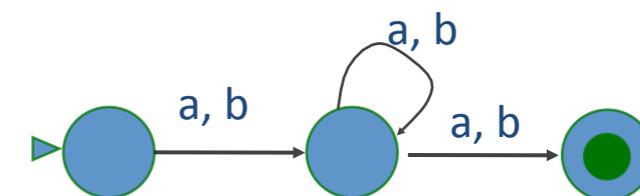
**Chairs**

- Michael Dorigo (University of Hamburg, Germany)
- Lawrence Cabec (University of Hamburg, Germany)
- Daniel Mohr (University of Hamburg, Germany)

# Formalização do processo de modelagem

A formalização do processo de modelagem de qualquer sistema dinâmico (ou de qualquer processo similar, seja qual for a aplicação) pode ser feito e Redes de Petri.

Para sistemas discretos, um formalismo alternativo e também muito genérico é proposto pela Computação Teórica com a Teoria de Autômatos. Na verdade esta representação também se enquadra no paradigma estado transição e pode descrever uma série de processos computacionais, onde o mais importante é o parser.



# CS154 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

- ◆ Finite automata are finite collections of states with transition rules that take you from one state to another.
- ◆ Original application was sequential switching circuits, where the “state” was the settings of internal bits.
- ◆ Today, several kinds of software can be modeled by FA.

# CS154 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

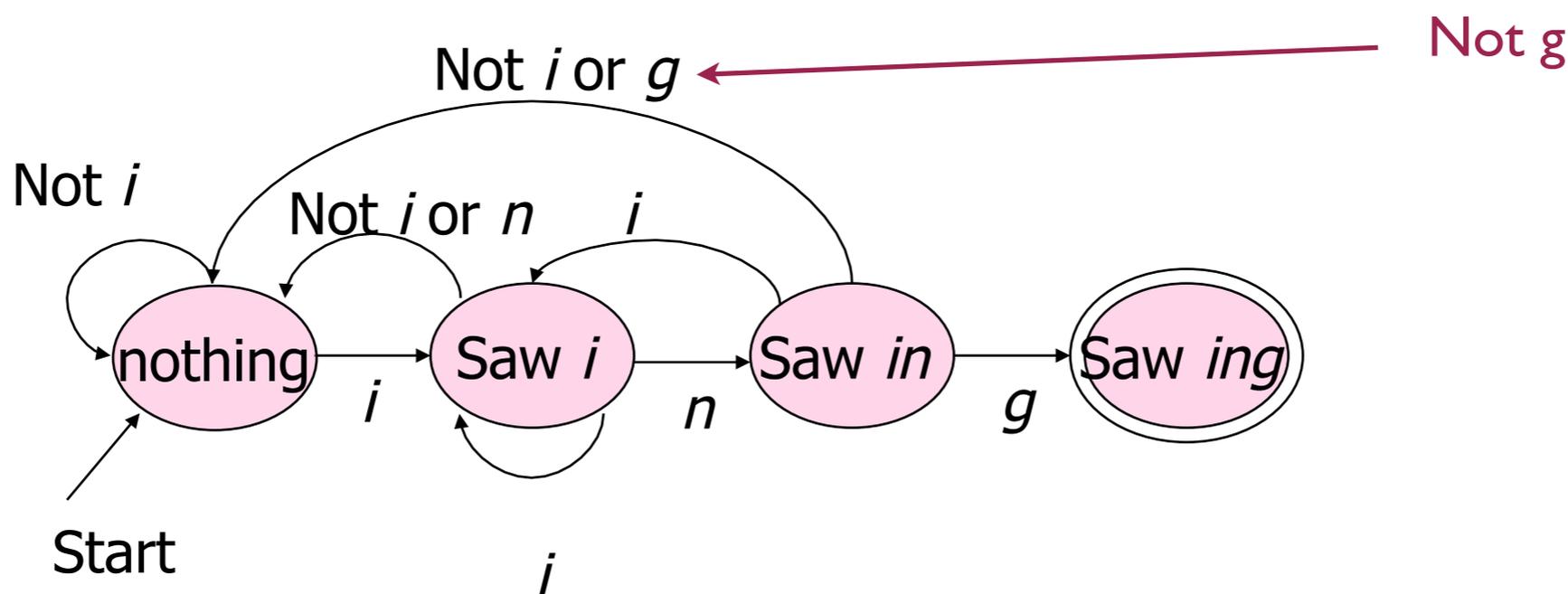
## Representing FA

- ◆ Simplest representation is often a graph.
  - ▶ Nodes = states.
  - ▶ Arcs indicate state transitions.
  - ▶ Labels on arcs tell what causes the transition.

# CSI 54 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

## Example: Recognizing Strings Ending in "ing"



Experimente agora fazer o parsing da palavra "mining".

# CS154 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

## Deterministic Finite Automata

- ◆ A formalism for defining languages, consisting of:
  1. A finite set of *states* ( $Q$ , typically).
  2. An *input alphabet* ( $\Sigma$ , typically).
  3. A *transition function* ( $\delta$ , typically).
  4. A *start state* ( $q_0$ , in  $Q$ , typically).
  5. A set of *final states* ( $F \subseteq Q$ , typically).
    - ◆ "Final" and "accepting" are synonyms.

1

# CSI 54 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

## The Transition Function

- ◆ Takes two arguments: a state and an input symbol.
- ◆  $\delta(q, a)$  = the state that the DFA goes to when it is in state  $q$  and input  $a$  is received.

7

# CSI 54 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

## Graph Representation of DFA's

- ◆ Nodes = states.
- ◆ Arcs represent transition function.
  - ▶ Arc from state  $p$  to state  $q$  labeled by all those input symbols that have transitions from  $p$  to  $q$ .
- ◆ Arrow labeled "Start" to the start state.
- ◆ Final states indicated by double circles.

8

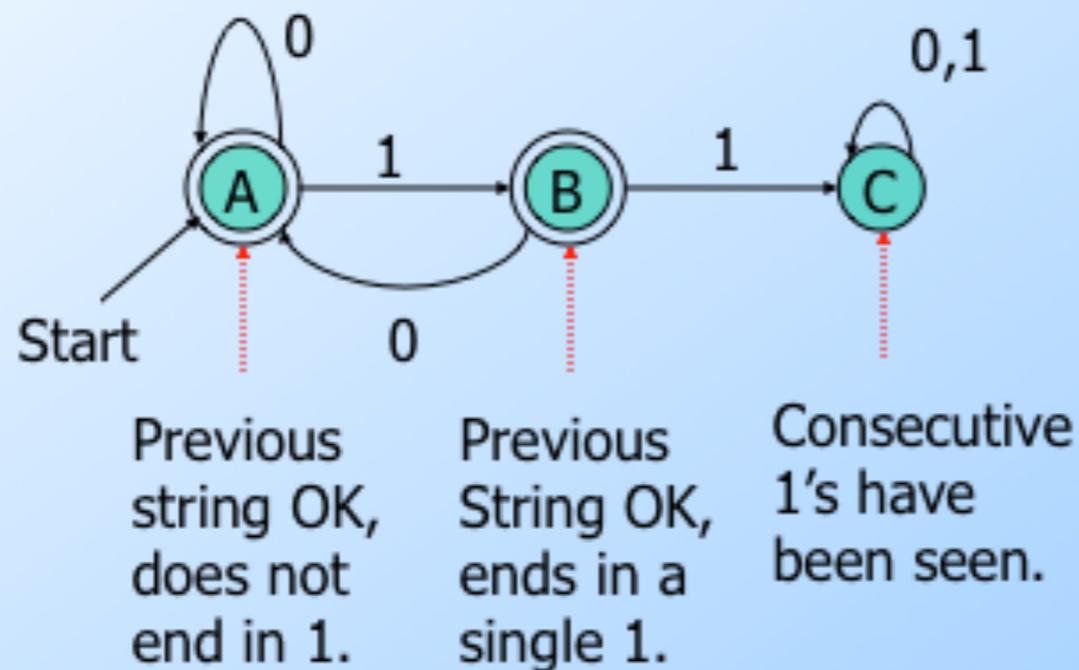
# CSI 54 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

Também chamado Transition Graph

## Example: Graph of a DFA

Accepts all strings without two consecutive 1's.

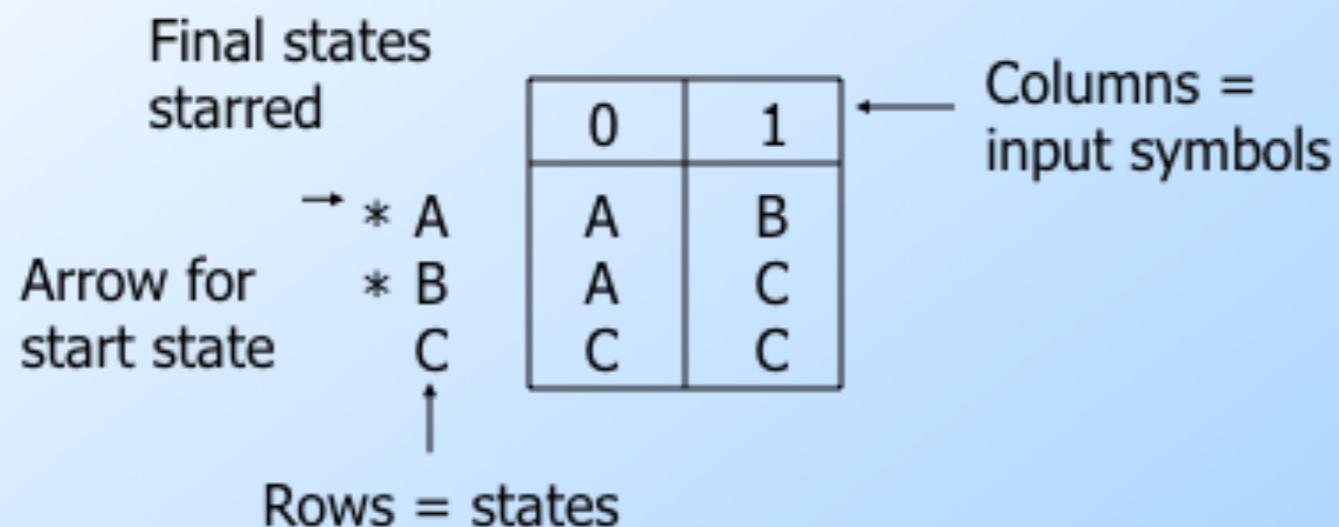


9

# CS I 54 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

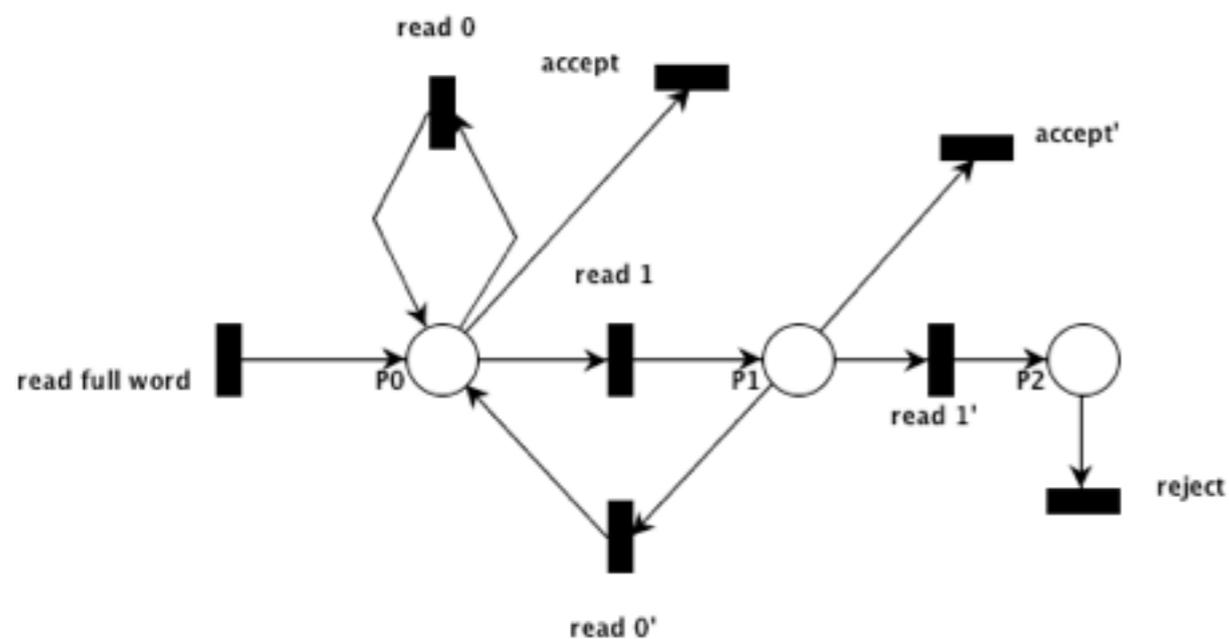
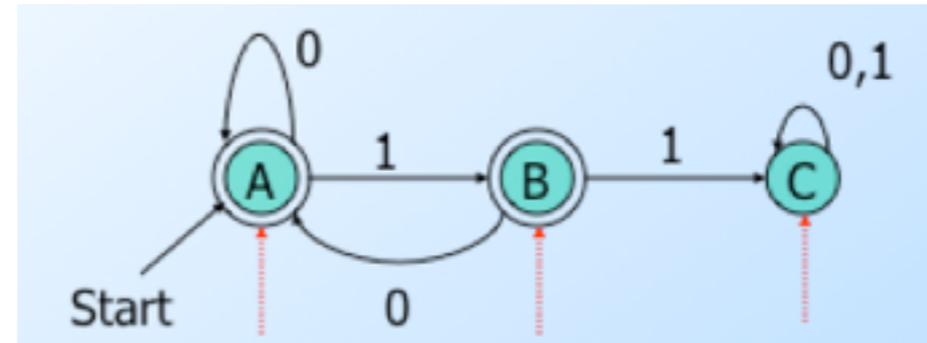
## Alternative Representation: Transition Table



<http://infolab.stanford.edu/~ullman/ialc/spr10/spr10.html#LECTURE%20NOTES>

10

Note que um TG (transition graph) também pode ser representado por uma rede de Petri Elementar (que ainda não definimos). Por enquanto vamos suportar esta analogia mapeando estados do TG com lugares de uma RdP e arcos do TG com transições da RdP.



Analise o grafo acima e a rede à esquerda. Você se sentiria tentado a “reduzir” o número de elementos desta rede?

# Modelagem Estado/Transição com Redes de Petri

- Estados e transições são noções distintas porém intercaladas;
- Ambos, estados e transições são entidades distribuídas;
- A extensão das mudanças causadas por uma transição é restrita e não depende do estado em que esta ocorre;
- Uma transição está *habilitada* em um estado sse as mudanças associadas à transição podem ocorrer neste estado, na extensão prefixada anteriormente.

## Estados e transições

O comportamento de um sistema dinâmico é representado por um estado  $S$  distribuído, formado por um conjunto de estados atômicos, como os que foram representados no exemplo simples do parser mostrado anteriormente.

Similarmente, as transições serão representadas por um conjunto de transições atômicas  $T$  desde que estejam todas habilitadas simultaneamente, de modo que estes conjuntos satisfazem à relação,

Disjunção entre estados e transições

- $S \cap T = \phi$ .

Portanto estados e transições são ambos elementos distribuídos:

Um estado distribuído é dado por um conjunto de condições válidas simultaneamente, isto é,

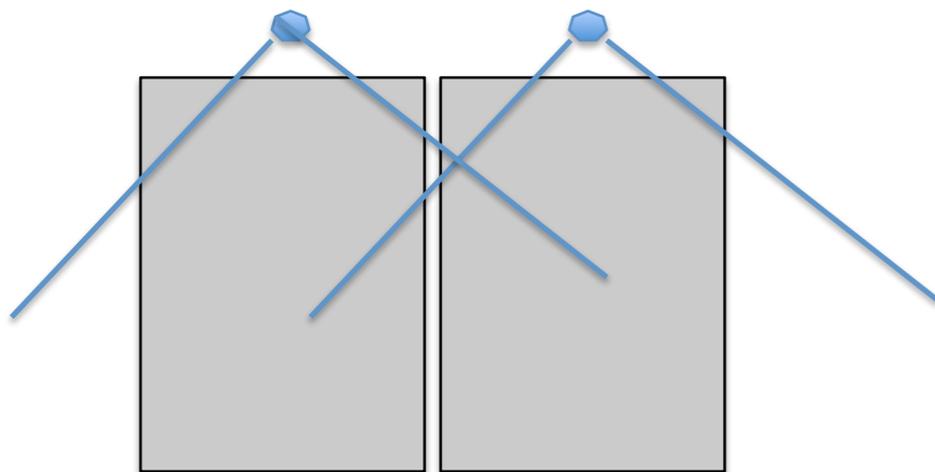
- $\{s_1, s_2, \dots, s_n\} \implies \textit{case}$ .

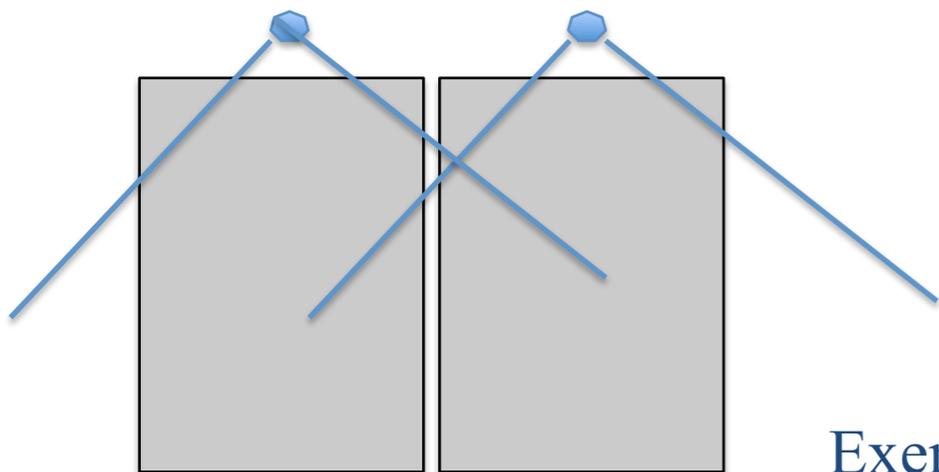
Uma transição distribuída é dada por um conjunto de transições válidas simultaneamente, isto é,

- $\{t_1, t_2, \dots, t_m\} \implies \textit{passo}$ .

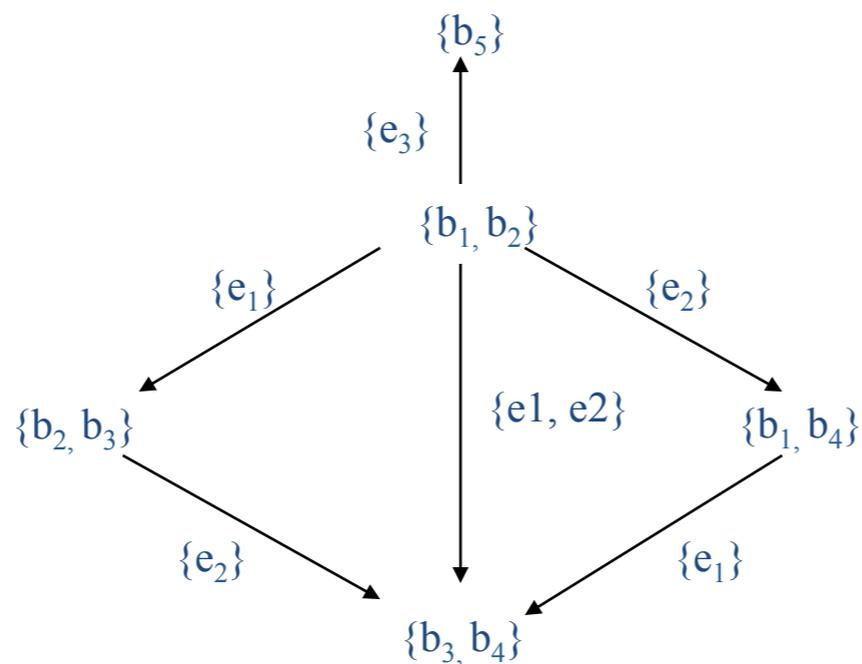
Um exemplo simples do que estamos falando, aplicado a sistemas de automação é a representação do comportamento de uma porta automática, que aciona motores para abertura e fechamento dependendo de sinal de sensores.

Um modelo bastante simples (e dificilmente encontrado na prática) é dado pelo esquema abaixo, onde cada porta tem um sensor com varredura determinada. Ao detectar a aproximação de uma pessoa a porta correspondente se abre, ou os dois lados se abrem no caso das duas detectarem simultaneamente a aproximação.

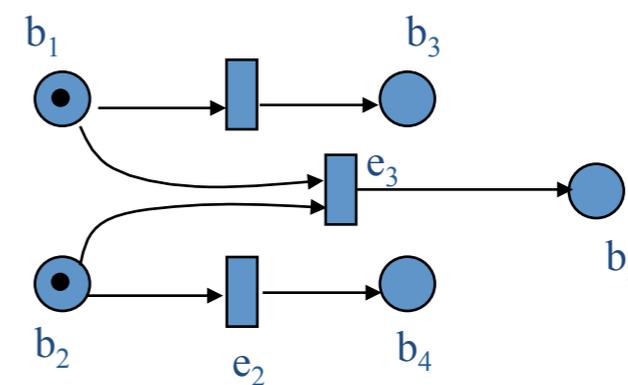




Exemplo (Thiagarajan) :



Grafo de atingibilidade



$$B = \{ b_1, b_2, b_3, b_4, b_5 \}$$

$$E = \{ e_1, e_2, e_3 \}$$

$$C = \{ \{b_1, b_2\}, \{ \{b_2, b_3\}, \{ \{b_1, b_4\}, \{ \{b_3, b_4\}, \{ \{b_5\} \} \}$$

$$U = \{ \{e_1\}, \{e_2\}, \{e_1, e_2\}, \{e_3\} \}$$

# Definição de Rede de Petri

## Definition

Definition 1] Uma rede de Petri é um grafo direcionado, simples, bipartido e conexo, representado pela n-upla  $N = (S, T; F)$ , onde  $S$  é um conjunto de estados  $\{s_i\}$ ,  $T$  é um conjunto de transições  $\{t_j\}$ , e  $F$  é uma relação de transição (o relação de fluxo), tal que:

i)  $S \cap T = \emptyset$  e  $S \cup T \neq \emptyset$ ;

ii)  $F \subseteq (S \times T) \cup (T \times S)$ ;

iii)  $dom(F) \cup ran(F) = S \cup T$ , onde

$$dom(F) = \{x \in (S \cup T) \mid \exists y \in (S \cup T). (x, y) \in F\},$$

$$ran(F) = \{y \in (S \cup T) \mid \exists x \in (S \cup T). (x, y) \in F\}.$$

# Princípios para modelagem em Redes de Petri

As redes possuem propriedades típicas dos esquemas que as tornam  
Uma excelente representação formal para sistemas (dinâmicos) discretos,  
Entre os quais figuram :

- o princípio da dualidade
- o princípio da localidade
- o princípio da concorrência
- o princípio da representação gráfica
- o princípio da representação algébrica

# O princípio da Dualidade

Existem dois tipos de elementos na rede: os elementos ditos passivos ou P-elementos (S-elementos), que representam estados, e os elementos ditos ativos ou T-elementos que representam mudança de estado ou transição.

Estes elementos são disjuntos, isto é, não existe na natureza nada que possa Possuir ambas as propriedades e questione que  $P \cap T = \emptyset$

# Exemplo: a largada na fórmula

I



Carro A



Carro B



Girault, C. & Valk, R.; Petri Nets for Systems Engineering, Springer, 2003

# Identificando os Estados

- $p_1$  = carro A: preparando-se para começar;
- $p_2$  = carro A: esperando o sinal de largada;
- $p_3$  = carro A: correndo;
- $p_4$  = sinal de prontidão do carro A enviado;
- $p_5$  = sinal de largada para o carro A enviado;
- $p_6$  = operador: esperando sinal de prontidão dos pilotos;
- $p_7$  = operador: sinal de largada enviado;
- $p_8$  = sinal de prontidão do carro B enviado;
- $p_9$  = sinal de largada para o carro B enviado;
- $p_{10}$  = carro B: preparando-se para começar;
- $p_{11}$  = carro B: esperando o sinal de largada;
- $p_{12}$  = carro B: correndo;

# Identificando as transições

- $t_1$  = carro A: envia sinal de prontidão
- $t_2$  = carro A: acelera
- $t_3$  = operador: manda sinal de largada
- $t_4$  = carro B: envia sinal de prontidão
- $t_5$  = carro B: acelera

## O estado inicial

O estado inicial (valor verdade das condições que compõem o estado):

$$M_1 = [p_1=T, p_2=F, p_3=F, p_4=F, p_5=F, p_6=T, p_7=F, p_8=F, p_9=F, p_{10}=T, p_{11}=F, p_{12}=F]$$

$p_1$  = carro A: preparando-se para começar;

$p_2$  = carro A: esperando o sinal de largada;

$p_3$  = carro A: correndo;

$p_4$  = sinal de prontidão do carro A enviado;

$p_5$  = sinal de largada para o carro A enviado;

$p_6$  = operador: esperando sinal de prontidão dos pilotos;

$p_7$  = operador: sinal de largada enviado;

$p_8$  = sinal de prontidão do carro B enviado;

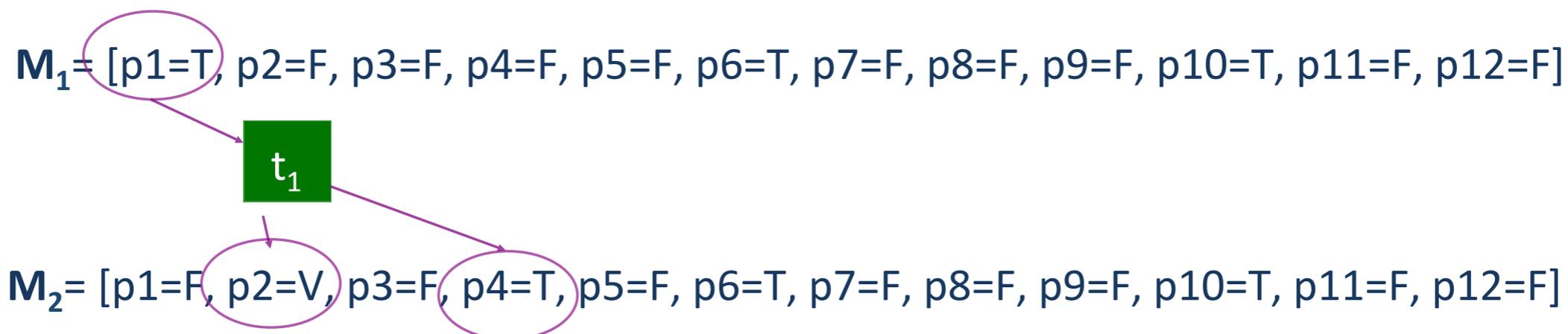
$p_9$  = sinal de largada para o carro B enviado;

$p_{10}$  = carro B: preparando-se para começar;

$p_{11}$  = carro B: esperando o sinal de largada;

$p_{12}$  = carro B: correndo;

# A dinâmica Estado/Transição

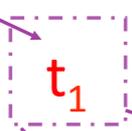


$p_1$  = carro A: preparando-se para começar;  
 $p_2$  = carro A: esperando o sinal de largada;  
 $p_3$  = carro A: correndo;  
 $p_4$  = sinal de prontidão do carro A enviado;  
 $p_5$  = sinal de largada para o carro A enviado;  
 $p_6$  = operador: esperando sinal de prontidão dos pilotos;  
 $p_7$  = operador: sinal de largada enviado;  
 $p_8$  = sinal de prontidão do carro B enviado;  
 $p_9$  = sinal de largada para o carro B enviado;  
 $p_{10}$  = carro B: preparando-se para começar;  
 $p_{11}$  = carro B: esperando o sinal de largada;  
 $p_{12}$  = carro B: correndo;

$t_1$  = carro A: envia sinal de prontidão  
 $t_2$  = carro A: acelera  
 $t_3$  = operador: manda sinal de largada  
 $t_4$  = carro B: envia sinal de prontidão  
 $t_5$  = carro B: acelera

# Transições independentes

$M_1 = [p_1=T, p_2=F, p_3=F, p_4=F, p_5=F, p_6=T, p_7=F, p_8=F, p_9=F, p_{10}=T, p_{11}=F, p_{12}=F]$



$M_2 = [p_1=F, p_2=T, p_3=F, p_4=T, p_5=F, p_6=T, p_7=F, p_8=T, p_9=F, p_{10}=F, p_{11}=T, p_{12}=F]$

- $p_1$  = carro A: preparando-se para começar;
- $p_2$  = carro A: esperando o sinal de largada;
- $p_3$  = carro A: correndo;
- $p_4$  = sinal de prontidão do carro A enviado;
- $p_5$  = sinal de largada para o carro A enviado;
- $p_6$  = operador: esperando sinal de prontidão dos pilotos;
- $p_7$  = operador: sinal de largada enviado;
- $p_8$  = sinal de prontidão do carro B enviado;
- $p_9$  = sinal de largada para o carro B enviado;
- $p_{10}$  = carro B: preparando-se para começar;
- $p_{11}$  = carro B: esperando o sinal de largada;
- $p_{12}$  = carro B: correndo;

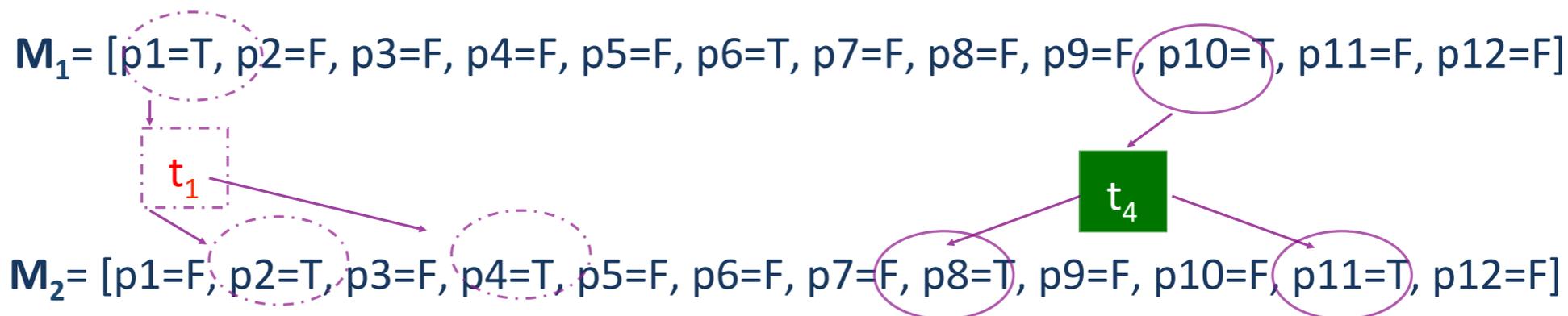
- $t_1$  = carro A: envia sinal de prontidão
- $t_2$  = carro A: acelera
- $t_3$  = operador: manda sinal de largada
- $t_4$  = carro B: envia sinal de prontidão
- $t_5$  = carro B: acelera

# O princípio da localidade

O princípio da localidade é atribuído a transições

A localidade de uma transição é dada pelo conjunto das suas pré-condições unido ao conjunto das pós-condições

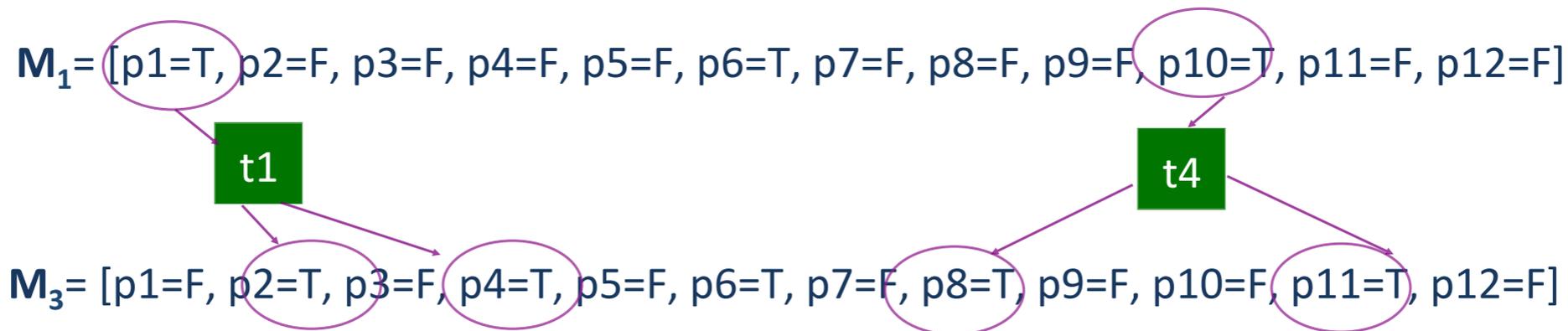
Pelo princípio da localidade o comportamento das transições depende unicamente da sua localidade.



- $p_1$  = carro A: preparando-se para começar;
- $p_2$  = carro A: esperando o sinal de largada;
- $p_3$  = carro A: correndo;
- $p_4$  = sinal de prontidão do carro A enviado;
- $p_5$  = sinal de largada para o carro A enviado;
- $p_6$  = operador: esperando sinal de prontidão dos pilotos;
- $p_7$  = operador: sinal de largada enviado;
- $p_8$  = sinal de prontidão do carro B enviado;
- $p_9$  = sinal de largada para o carro B enviado;
- $p_{10}$  = carro B: preparando-se para começar;
- $p_{11}$  = carro B: esperando o sinal de largada;
- $p_{12}$  = carro B: correndo;

- $t_1$  = carro A: envia sinal de prontidão
- $t_2$  = carro A: acelera
- $t_3$  = operador: manda sinal de largada
- $t_4$  = carro B: envia sinal de prontidão
- $t_5$  = carro B: acelera

# Eventos independentes



$p_1$  = carro A: preparando-se para começar;  
 $p_2$  = carro A: esperando o sinal de largada;  
 $p_3$  = carro A: correndo;  
 $p_4$  = sinal de prontidão do carro A enviado;  
 $p_5$  = sinal de largada para o carro A enviado;  
 $p_6$  = operador: esperando sinal de prontidão dos pilotos;  
 $p_7$  = operador: sinal de largada enviado;  
 $p_8$  = sinal de prontidão do carro B enviado;  
 $p_9$  = sinal de largada para o carro B enviado;  
 $p_{10}$  = carro B: preparando-se para começar;  
 $p_{11}$  = carro B: esperando o sinal de largada;  
 $p_{12}$  = carro B: correndo;

$t_1$  = carro A: envia sinal de prontidão  
 $t_2$  = carro A: acelera  
 $t_3$  = operador: manda sinal de largada  
 $t_4$  = carro B: envia sinal de prontidão  
 $t_5$  = carro B: acelera

# Princípio da concorrência

## Princípio da Concorrência

- Duas transições  $t_1$  e  $t_2$  são cocorrentes se e somente se suas localidades são disjuntas, isto é,  
$$Indep(t_1, t_2) \Leftrightarrow \{Loc(t_1) \cap Loc(t_2) = \emptyset\}$$

# Princípio da Representação Gráfica

Qualquer sistema representado em Redes de Petri admite uma representação gráfica, e esta representação obedece a regras rígidas que dão consistência ao modelo.

## Elementos para representação gráfica

- Os elementos ditos passivos (por convenção) são chamados lugares (ou S-elementos, do termo sahlen em alemão) e são representados graficamente por círculos ou elipses.
- Os elementos ditos ativos (por convenção) são chamados transições e são representados por retângulos ou barras.
- Lugares e transições são ligados por arcos orientados de modo que um arco liga sempre um lugar a uma transição ou vice-versa, mas nunca dois lugares ou duas transições.

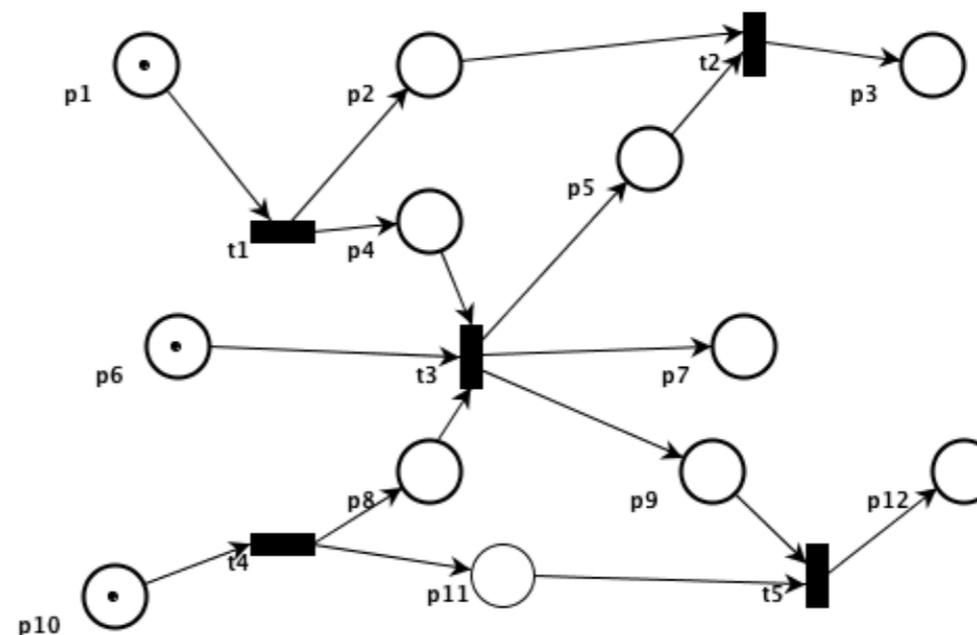
# Modelagem Estado/Transição X Modelagem em Redes de Petri



- Estados e transições são noções distintas porém intercaladas;
- Ambos, estados e transições são entidades distribuídas;
- A extensão das mudanças causadas por uma transição é restrita e não depende do estado em que esta ocorre;
- Uma transição está *habilitada* em um estado sse as mudanças associadas à transição podem ocorrer neste estado, na extensão prefixada anteriormente.

# Voltando ao exemplo

$p_1$  = carro A: preparando-se para começar;  
 $p_2$  = carro A: esperando o sinal de largada;  
 $p_3$  = carro A: correndo;  
 $p_4$  = sinal de prontidão do carro A enviado;  
 $p_5$  = sinal de largada para o carro A enviado;  
 $p_6$  = operador: esperando sinal de prontidão dos pilotos;  
 $p_7$  = operador: sinal de largada enviado;  
 $p_8$  = sinal de prontidão do carro B enviado;  
 $p_9$  = sinal de largada para o carro B enviado;  
 $p_{10}$  = carro B: preparando-se para começar;  
 $p_{11}$  = carro B: esperando o sinal de largada;  
 $p_{12}$  = carro B: correndo;

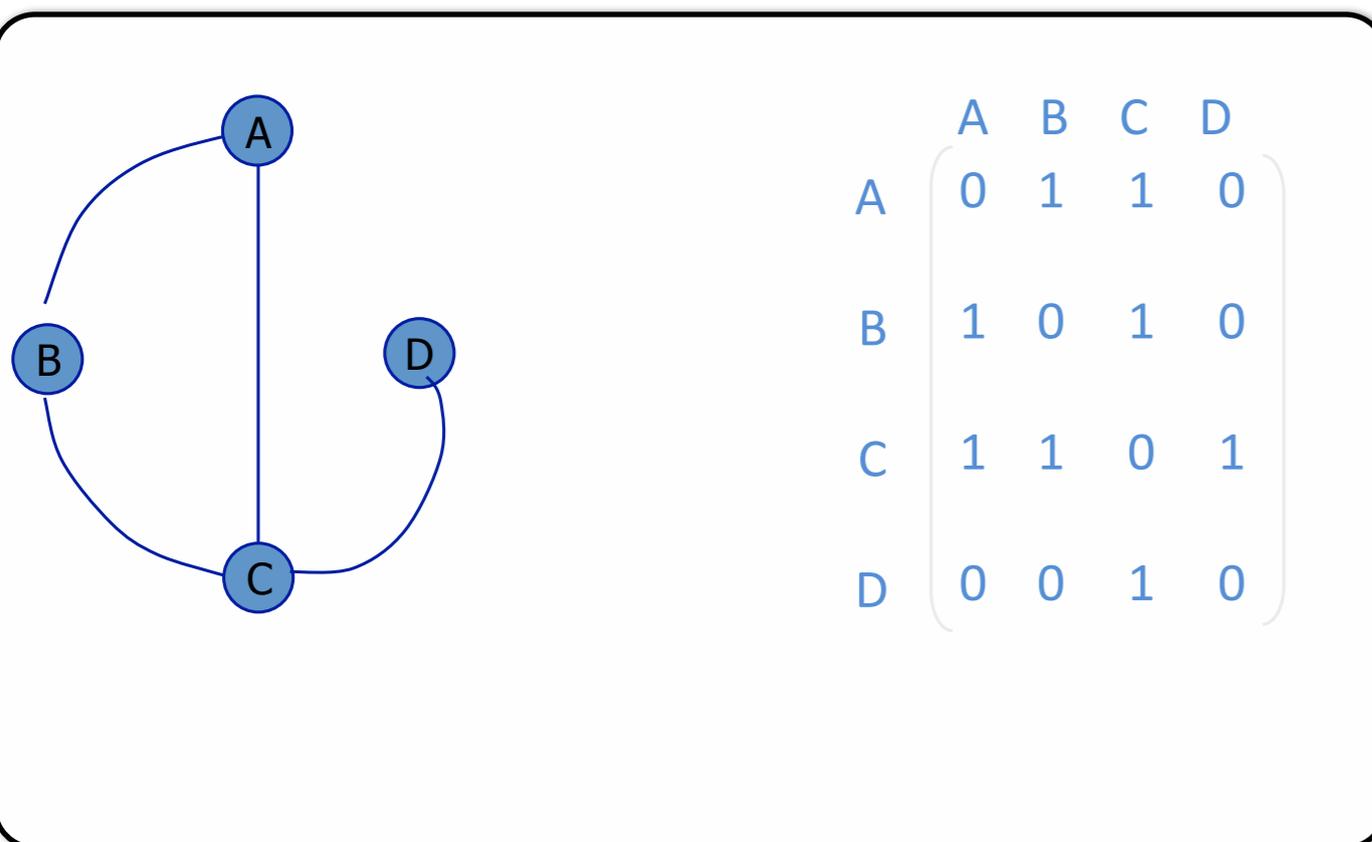


$t_1$  = carro A: envia sinal de prontidão  
 $t_2$  = carro A: acelera  
 $t_3$  = operador: manda sinal de largada  
 $t_4$  = carro B: envia sinal de prontidão  
 $t_5$  = carro B: acelera

# Princípio da Representação Algébrica

Todo sistema modelado em Redes de Petri admite uma e somente uma representação algébrica, que é dada pelas matrizes que representam a estrutura do modelo (estados e transições distribuídas) e por uma equação de estado que representa a evolução dos estados pela ocorrência de transições habilitadas.

A relação entre representação gráfica e matrizes é dada pela teoria de grafos, onde um grafo simples não-orientado pode ser representado por uma matriz.



No caso dos grafos todos os nós podem ser conectados por um arco. Em princípio, portanto a matriz mapeia as vizinhanças (a existência de arcos) entre os nós e de fato é uma tabela de dupla entrada. Para os pares de  $V \times V$  que sejam vizinhos a posição respectiva recebe um “1”; se não existe vizinhança a posição recebe “0”. Naturalmente, para grafos simples, a diagonal principal é nula, e, se o grafo é não-direcionado, a matriz é simétrica com relação a diagonal principal.

## Matriz de Incidência

No caso das redes de Petri temos que:

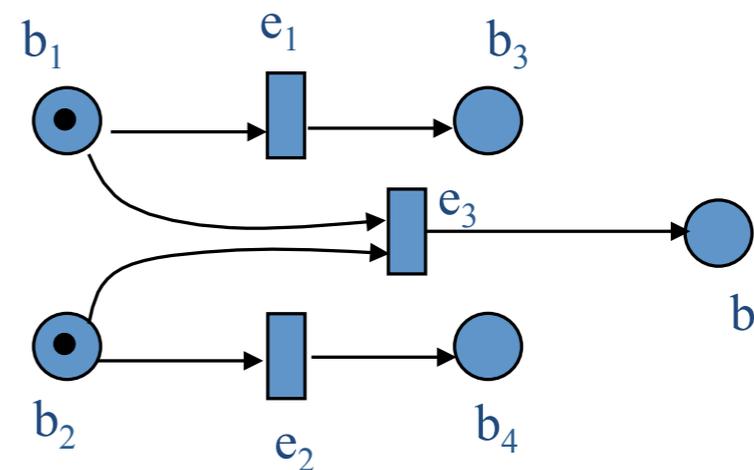
i) se a rede é elementar a matriz de incidência será também unimodular como no caso dos grafos, mas para outros tipos de rede poderemos ter mais de um arco entre lugares e transições e os elementos da matriz poderão assumir valores maiores que a unidade;

ii) as redes de Petri são um grafo bi-partido, isto é, um grafo onde o conjunto de vértices  $V$  é dividido em dois conjuntos (o dos lugares e o das transições) e só pode haver vizinhança entre elementos destes dois conjuntos. Assim, se o número de transições e lugares não coincide a matriz não é necessariamente quadrada e não necessariamente tem um determinante.

## Exemplo:

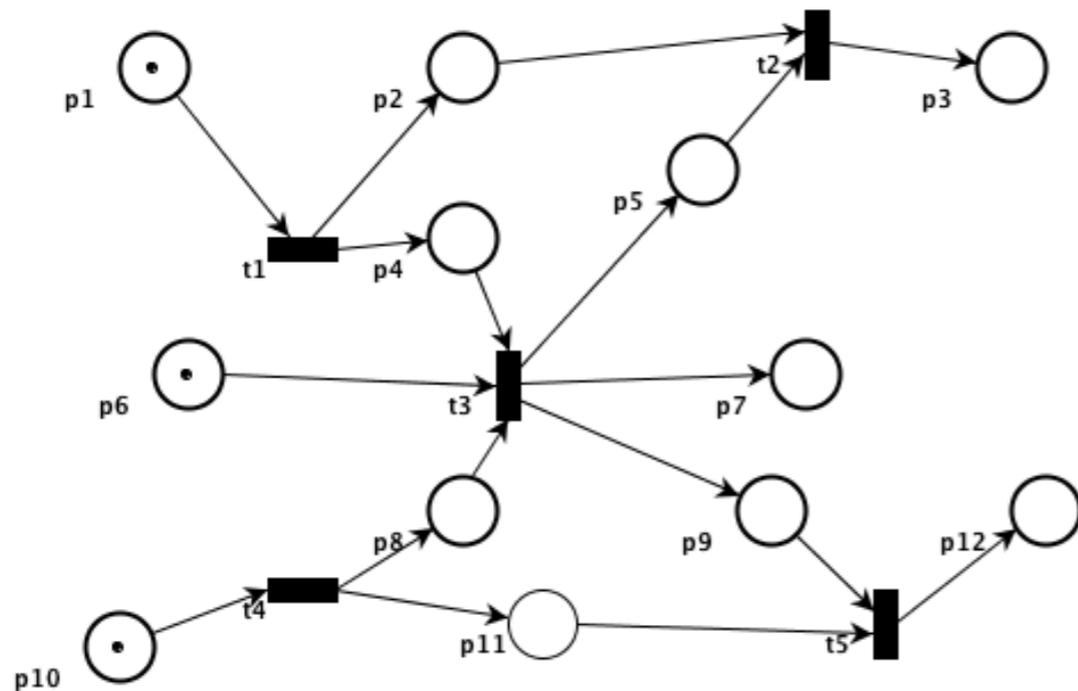
Usando o mesmo exemplo da porta automática podemos construir a matriz de incidência desta rede.

No caso das redes se faz uma tabela de dupla entrada das transições contra os lugares. Cada elemento da matriz indica se o lugar é incidente, -1, emergente, 1, ou se não há conexão de nenhum tipo, 0.



	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$
$e_1$	-1	0	1	0	0
$e_2$	0	-1	0	1	0
$e_3$	-1	-1	0	0	1

# Exemplo da largada de Fórmul



Incidence & Marking

Source net

Use current net    Filename:    

---

Results

**Combined Incidence matrix  $I$**

	t1	t2	t3	t4	t5
p1	-1	0	0	0	0
p2	1	-1	0	0	0
p6	0	0	-1	0	0
p7	0	0	1	0	0
p4	1	0	-1	0	0
p3	0	1	0	0	0
p5	0	-1	1	0	0
p10	0	0	0	-1	0
p11	0	0	0	1	-1
p8	0	0	-1	1	0
p9	0	0	1	0	-1
p12	0	0	0	0	1

# Redes de Petri: Definição

## Definition

Definition 1] Uma rede de Petri é um grafo direcionado, simples, bipartido e conexo, representado pela n-upla  $N = (S, T; F)$ , onde  $S$  é um conjunto de estados  $\{s_i\}$ ,  $T$  é um conjunto de transições  $\{t_j\}$ , e  $F$  é uma relação de transição (o relação de fluxo), tal que:

i)  $S \cap T = \emptyset$  e  $S \cup T \neq \emptyset$ ;

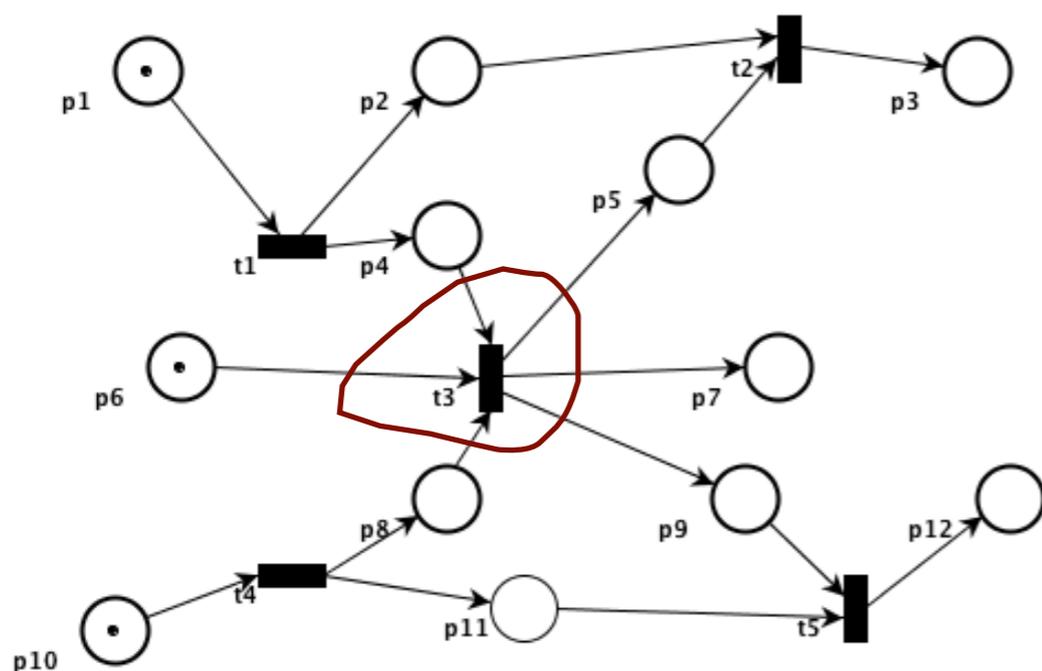
ii)  $F \subseteq (S \times T) \cup (T \times S)$ ;

iii)  $dom(F) \cup ran(F) = S \cup T$ , onde

$$dom(F) = \{x \in (S \cup T) \mid \exists y \in (S \cup T). (x, y) \in F\},$$

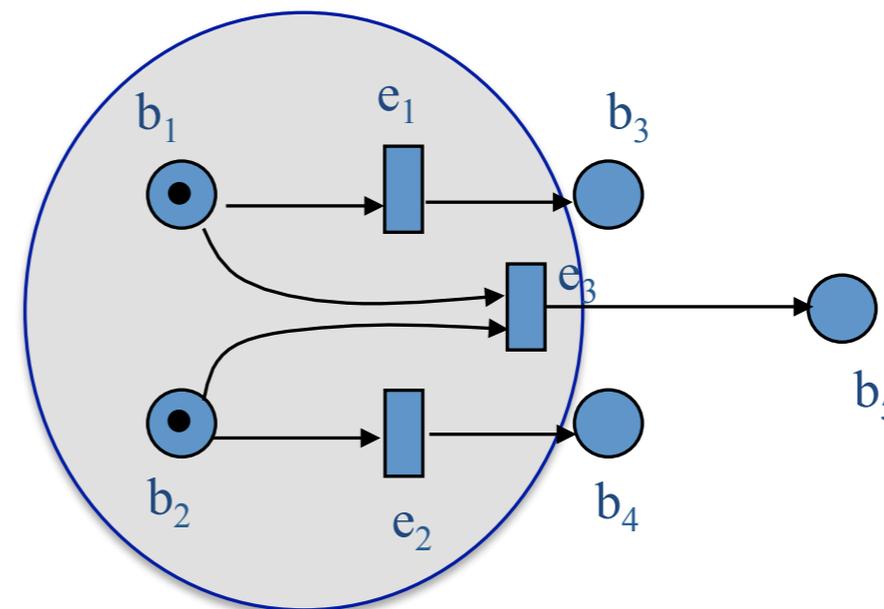
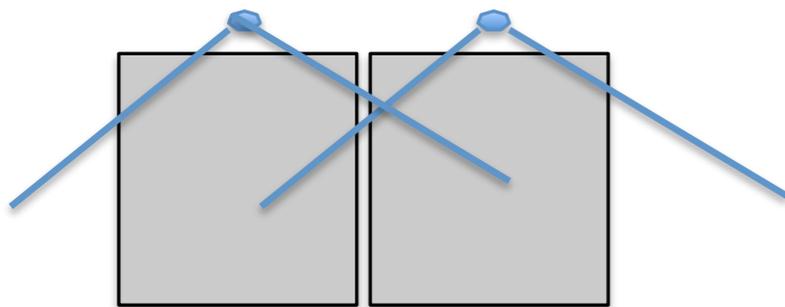
$$ran(F) = \{y \in (S \cup T) \mid \exists x \in (S \cup T). (x, y) \in F\}.$$

## Configurações especiais: sincronismo



A convergência de diversos caminhos na rede para uma única transição sincroniza as ações, uma vez que a evolução dos estados por esta transição só ocorre quando todos os lugares que antecedem a transição estão marcados.

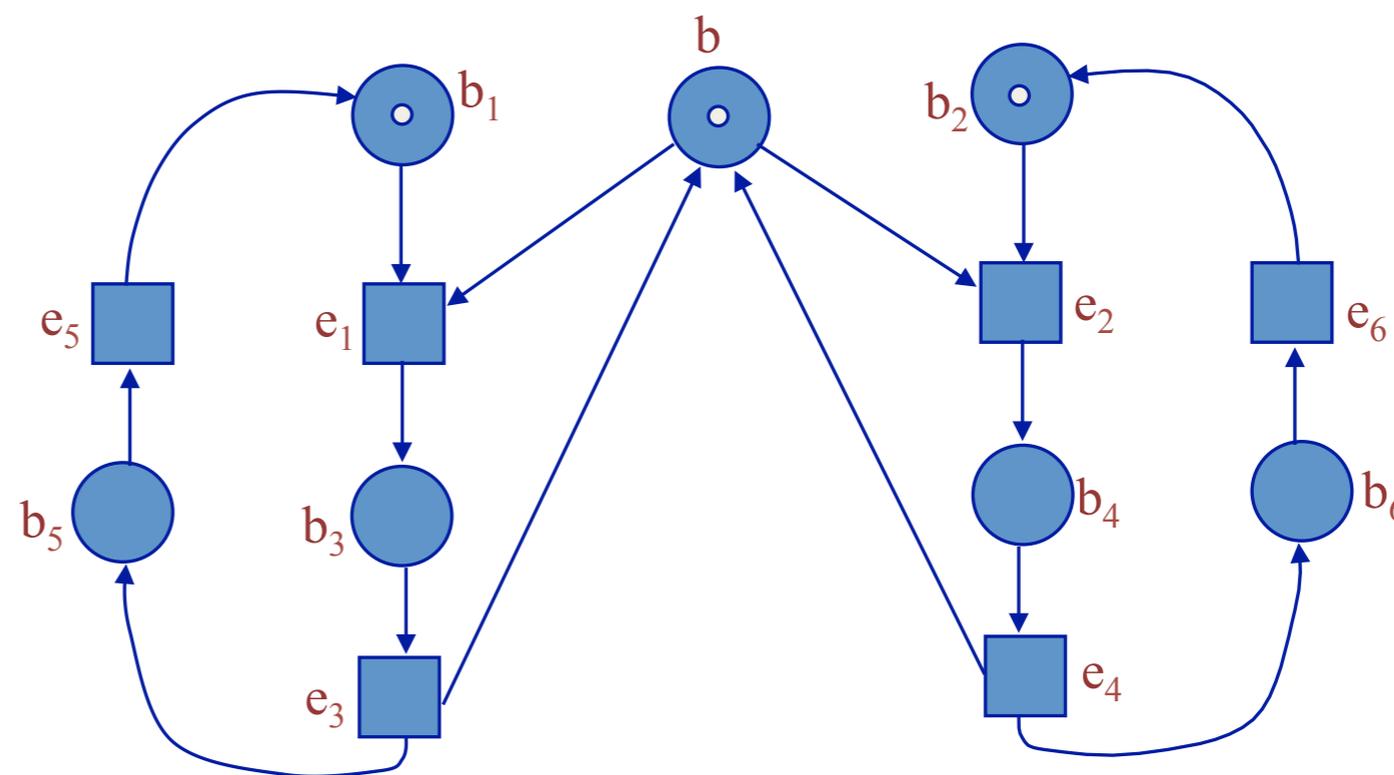
# Configurações especiais: conflito



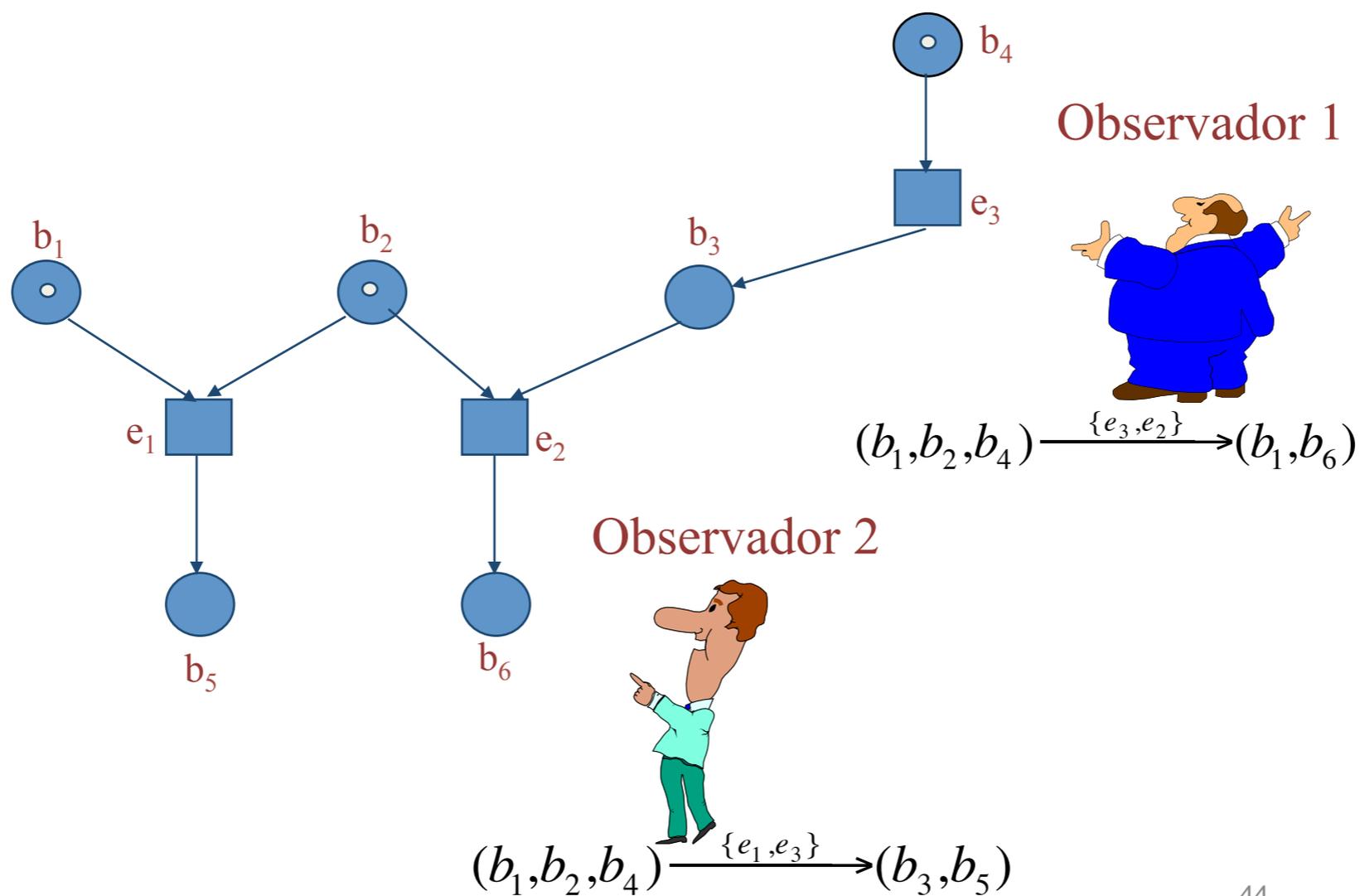
O Conflito denota uma situação de exclusão mútua (mutex), onde a ocorrência de uma transição inibe outras. No caso do exemplo as transições  $\{e_1, e_2\}$  são mutuamente exclusivas à transição  $\{e_3\}$ .

# Semáforo: o mutex clássico

Em uma situação genérica a mútua exclusão (mutex) é caracterizada por processos (transições independentes que compartilham pelo menos um lugar. O início de um processo desabilita este lugar e faz com que o outro seja desativado.



# Configurações especiais: contato ou confusão



44

# Definindo pre-set e pos-set

Para formalizar a definição de localidade precisamos do conceito de pre-set e pos-set de um elemento da rede (lugar ou transição):

## Def. 2

Seja uma rede de Petri  $N = (S, T; F)$  e sejam  $X_N = S \cup T$ ,  $x \in X_N$  e  $Y_N \subseteq X_N$ :

- $\bullet x = \{y \in X_N \mid (y, x) \in F\}$
- $x \bullet = \{y \in X_N \mid (x, y) \in F\}$
- $\bullet Y = \bigcup_{(x \in Y)} \bullet x$  e  $Y \bullet = \bigcup_{(x \in Y)} x \bullet$

## Definindo localidade

A localidade de um elemento da rede pode ser definida do seguinte modo:

### Def. 3

Seja uma rede de Petri  $N = (S, T; F)$  e seja  $X_N = S \cup T$ ,  $x \in X_N$ . A localidade de um elemento genérico  $x$  é definida como:

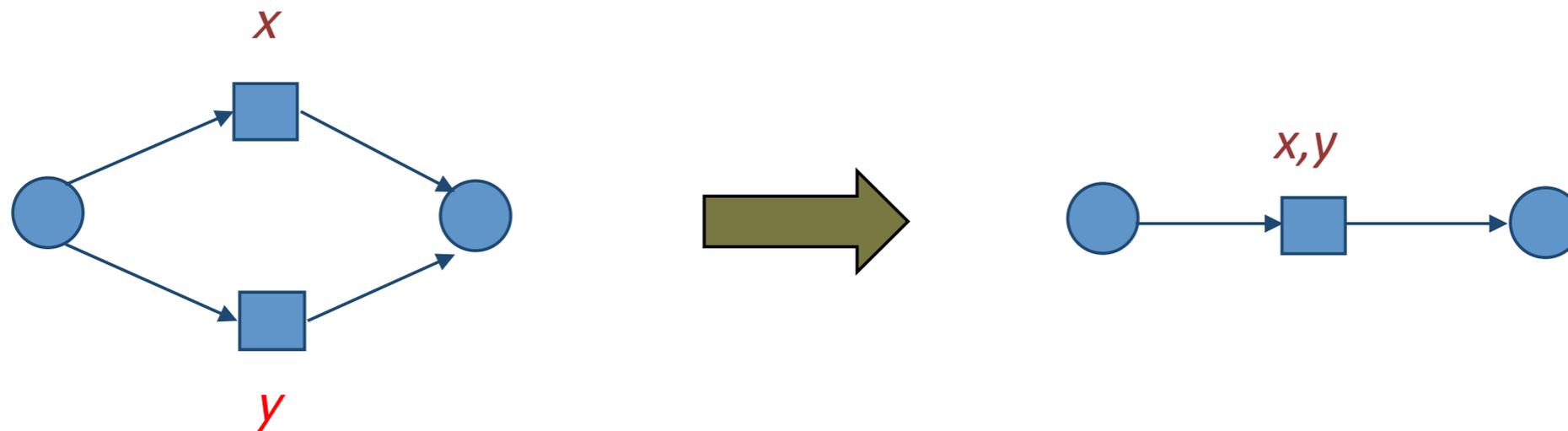
$$Loc(x) = \{x\} \cup \bullet x \cup x \bullet$$

A convenção de considerar sempre as transições como elemento ativo nos leva a considerar de forma privilegiada a localidade das transições como elemento de análise.

## Def. 4

Seja uma rede de Petri  $N = (S, T; F)$  e sejam  $X_N = S \cup T$ ,  $x \in X_N$ .

- $N$  é dita uma rede pura se e somente se  $\forall x \in X_N \bullet x \cap x \bullet = \emptyset$ ;
- $N$  é dita uma rede simples se e somente se  $\forall x, y \in X_N [(\bullet x = \bullet y) \wedge (x \bullet = y \bullet) \implies (x = y)]$ .



## Estados distribuídos

### Def. 5

Seja uma rede de Petri  $N = (S, T; F)$ . Um subconjunto  $M \subseteq S$  de condições válidas em um dado instante é dito um estado (distribuído) do sistema modelado por  $N$ .

Um estado também pode ser representado por um vetor

$$\vec{M} = \begin{pmatrix} s_1 \\ \vdots \\ s_m \end{pmatrix} \text{ onde } s_i = 1 \text{ se o } s_i \in M.$$

# Transições elementares

## Def. 6

Seja uma rede de Petri  $N = (S, T; F)$ . Uma transição elementar  $t \in T$  está habilitada em um estado  $M \subseteq S$  se e somente se  $(\bullet t \subseteq M) \wedge (t \bullet \cap M = \emptyset)$ .

Esta definição insere uma regra de habilitação estrita. Mais tarde poderemos flexibilizar esta regra considerando somente o primeiro termo da conjunção.

## Evolução Estado/Transição

### Def. 7

Seja uma rede de Petri  $N = (S, T; F)$ . Se uma transição elementar  $t \in T$  habilitada em um estado  $M \subseteq S$  ocorre, o estado  $M$  evolui para o estado  $M' = (M \setminus \bullet t) \cup t \bullet$ .

Até aqui estamos nos restringindo à rede mais intuitiva, isto é, as redes elementares, onde cada lugar admite no máximo uma marca. Embora seja conveniente para processos lógicos e sequenciais e circuitos digitais, esta restrição é muito forte e será removida mais adiante para ter uma rede mais geral, chamada Place/Transition ou simplesmente P/T.

## Informal Introduction to Petri Nets, W.Reisig, G. Rozemberg, LNCS 1491, 1-10.

3<sup>rd</sup>. Advanced Course in Petri Nets,  
Dagstuhl, Germany, October, 1996



# PMR 5237

1a. ciclo de 2013

12 aulas

5 listas

4 milestones

1 artigo

1 prova

$L$  - média das listas de exercício  
 $n_l$  - número de listas feitas  
 $n_t$  - número total de listas (5)  
 $T_f$  - média do trabalho final (artigo)  
 $n_m$  - número de milestones feitos  
 $n_r$  - número total de milestones  
 $P$  - Nota da prova

$$M_f = (4 \cdot (n_l/n_t)L + 4 \cdot (n_m/n_r)T_f + 2 \cdot P) / 10$$

## Stoa - USP

As Stoa eram espaços públicos onde as pessoas expunham quadros, obras de artesanato, vendiam produtos, víveres, declamavam, etc. Eram comuns na Grecia em vários períodos, incluindo o Helenístico.

O projeto Stoa-USP é uma stoa virtual onde a comunidade USP pode debater, postar opiniões, criar blogs, compartilhar notícias, grupos de discussão e também disciplinas, estas através do sistema Moodle de educação a distância. Vários cursos de graduação e pós-graduação se encontram no Stoa-USP incluindo PMR 5237.

Neste caso o sistema está acoplado diretamente com o sistema Janus e deve usar como e-mail o endereço oficial no sistema USP. Os alunos ouvintes devem se registrar no sistema STOA e serão incluídos depois no sistema.



Stoa de Atallos, Grecia, 159-138 AC, reformado recentemente

# STOA - USP

# Suporte a disciplinas : Moodle STOA

The screenshot shows a web browser window displaying a Moodle course page. The browser's address bar shows the URL: `disciplinas.stoa.usp.br/course/view.php?id=3078&notifyeditingon=1`. The page title is "Modelagem e Design de Sistemas Discretos em Redes de Petri (Modelagem e Design de Sistemas)".

The page layout includes a navigation menu with links for "Disciplinas", "Suporte", "Stoa", "USP", and "Acesso". Below this is a breadcrumb trail: "Início > Meus Ambientes > EP > PMR > PMR5237 - 3".

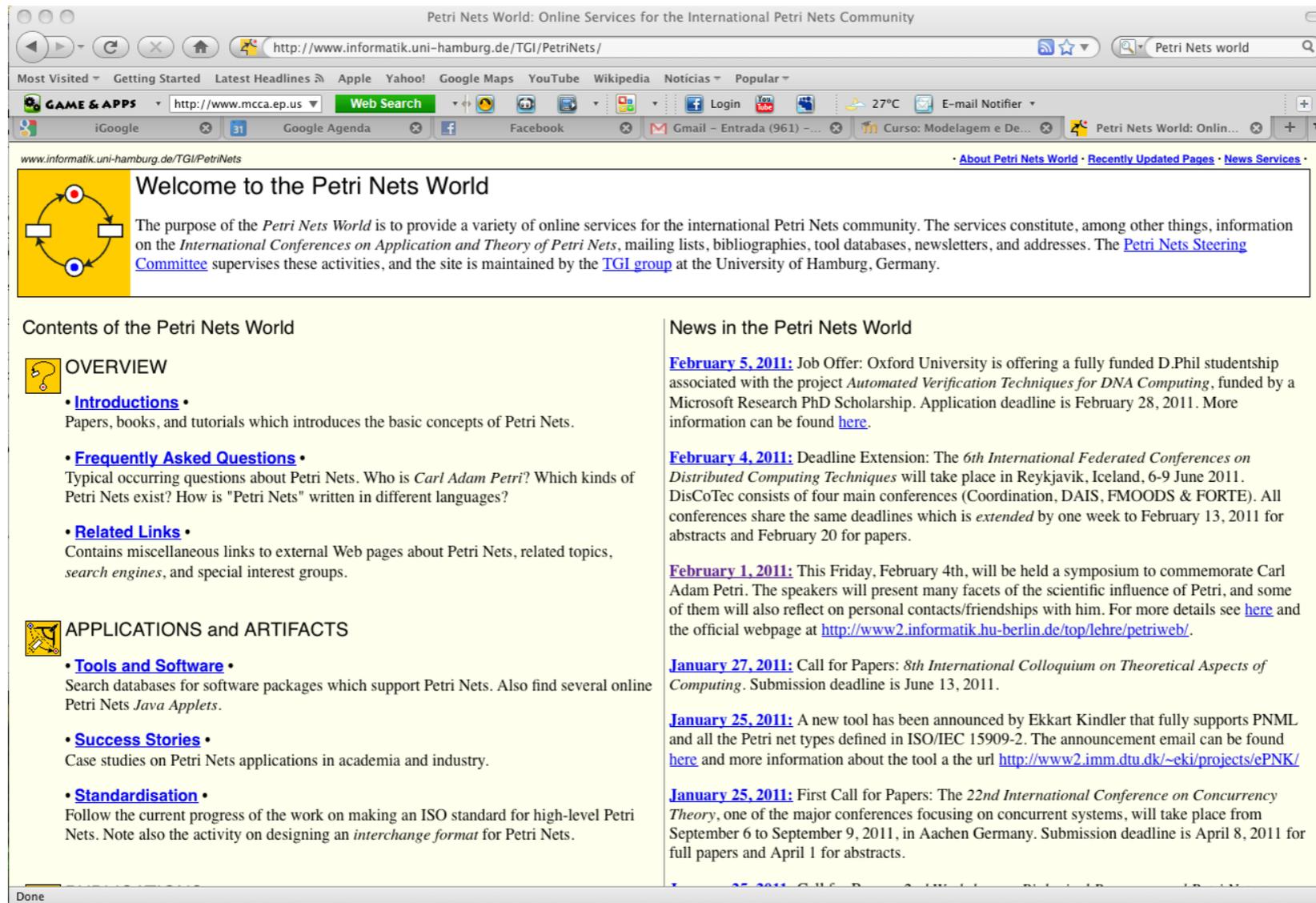
The main content area is titled "Introdução (Aula 1)" and features a large image showing a classroom with students and a Petri net diagram. The text below the image discusses the history and application of Petri nets, mentioning their use in formal verification and simulation in industrial settings.

On the left side, there is an "Administração" sidebar with options like "Desativar edição", "Escolhedor de atividade ligado", "Editar configurações", "Usuários", "Filtros", "Relatórios", "Notas", "Resultado da aprendizagem", "Badges", "Backup", "Restaurar", "Importar", and "Banco de questões".

On the right side, there are three widgets: "Pesquisar nos Fóruns" with a search box and "Vai" button; "Últimas notícias" with a link to "Acrescentar um novo tópico..."; and "Próximos eventos" with a notice for "CONFIRMAÇÃO DE MATRÍCULA PARA OS INGRESSANTES CONVOCADOS PELA FUVEST" scheduled for 01:00 on Monday, 18:00.

# Referência na Internet: Petri Nets World

<http://www.informatik.uni-hamburg.de/TGI/PetriNets/>



Petri Nets World: Online Services for the International Petri Nets Community

<http://www.informatik.uni-hamburg.de/TGI/PetriNets/>

Most Visited Getting Started Latest Headlines Apple Yahoo! Google Maps YouTube Wikipedia Noticias Popular

GAME & APPS <http://www.mcca.ep.us> Web Search

iGoogle 31 Google Agenda Facebook Gmail - Entrada (961) ... Curso: Modelagem e De... Petri Nets World: Onlin...

[About Petri Nets World](#) [Recently Updated Pages](#) [News Services](#)

## Welcome to the Petri Nets World

The purpose of the *Petri Nets World* is to provide a variety of online services for the international Petri Nets community. The services constitute, among other things, information on the *International Conferences on Application and Theory of Petri Nets*, mailing lists, bibliographies, tool databases, newsletters, and addresses. The [Petri Nets Steering Committee](#) supervises these activities, and the site is maintained by the [TGI group](#) at the University of Hamburg, Germany.

### Contents of the Petri Nets World

**OVERVIEW**

- Introductions** • Papers, books, and tutorials which introduces the basic concepts of Petri Nets.
- Frequently Asked Questions** • Typical occurring questions about Petri Nets. Who is *Carl Adam Petri*? Which kinds of Petri Nets exist? How is "Petri Nets" written in different languages?
- Related Links** • Contains miscellaneous links to external Web pages about Petri Nets, related topics, *search engines*, and special interest groups.

**APPLICATIONS and ARTIFACTS**

- Tools and Software** • Search databases for software packages which support Petri Nets. Also find several online Petri Nets *Java Applets*.
- Success Stories** • Case studies on Petri Nets applications in academia and industry.
- Standardisation** • Follow the current progress of the work on making an ISO standard for high-level Petri Nets. Note also the activity on designing an *interchange format* for Petri Nets.

### News in the Petri Nets World

**February 5, 2011:** Job Offer: Oxford University is offering a fully funded D.Phil studentship associated with the project *Automated Verification Techniques for DNA Computing*, funded by a Microsoft Research PhD Scholarship. Application deadline is February 28, 2011. More information can be found [here](#).

**February 4, 2011:** Deadline Extension: The *6th International Federated Conferences on Distributed Computing Techniques* will take place in Reykjavik, Iceland, 6-9 June 2011. DisCoTec consists of four main conferences (Coordination, DAIS, FMOODS & FORTE). All conferences share the same deadlines which is *extended* by one week to February 13, 2011 for abstracts and February 20 for papers.

**February 1, 2011:** This Friday, February 4th, will be held a symposium to commemorate Carl Adam Petri. The speakers will present many facets of the scientific influence of Petri, and some of them will also reflect on personal contacts/friendships with him. For more details see [here](#) and the official webpage at <http://www2.informatik.hu-berlin.de/top/lehre/petriweb/>.

**January 27, 2011:** Call for Papers: *8th International Colloquium on Theoretical Aspects of Computing*. Submission deadline is June 13, 2011.

**January 25, 2011:** A new tool has been announced by Ekkart Kindler that fully supports PNML and all the Petri net types defined in ISO/IEC 15909-2. The announcement email can be found [here](#) and more information about the tool a the url <http://www2.imm.dtu.dk/~eki/projects/ePNK/>

**January 25, 2011:** First Call for Papers: The *22nd International Conference on Concurrency Theory*, one of the major conferences focusing on concurrent systems, will take place from September 6 to September 9, 2011, in Aachen Germany. Submission deadline is April 8, 2011 for full papers and April 1 for abstracts.

# Ferramentas de software

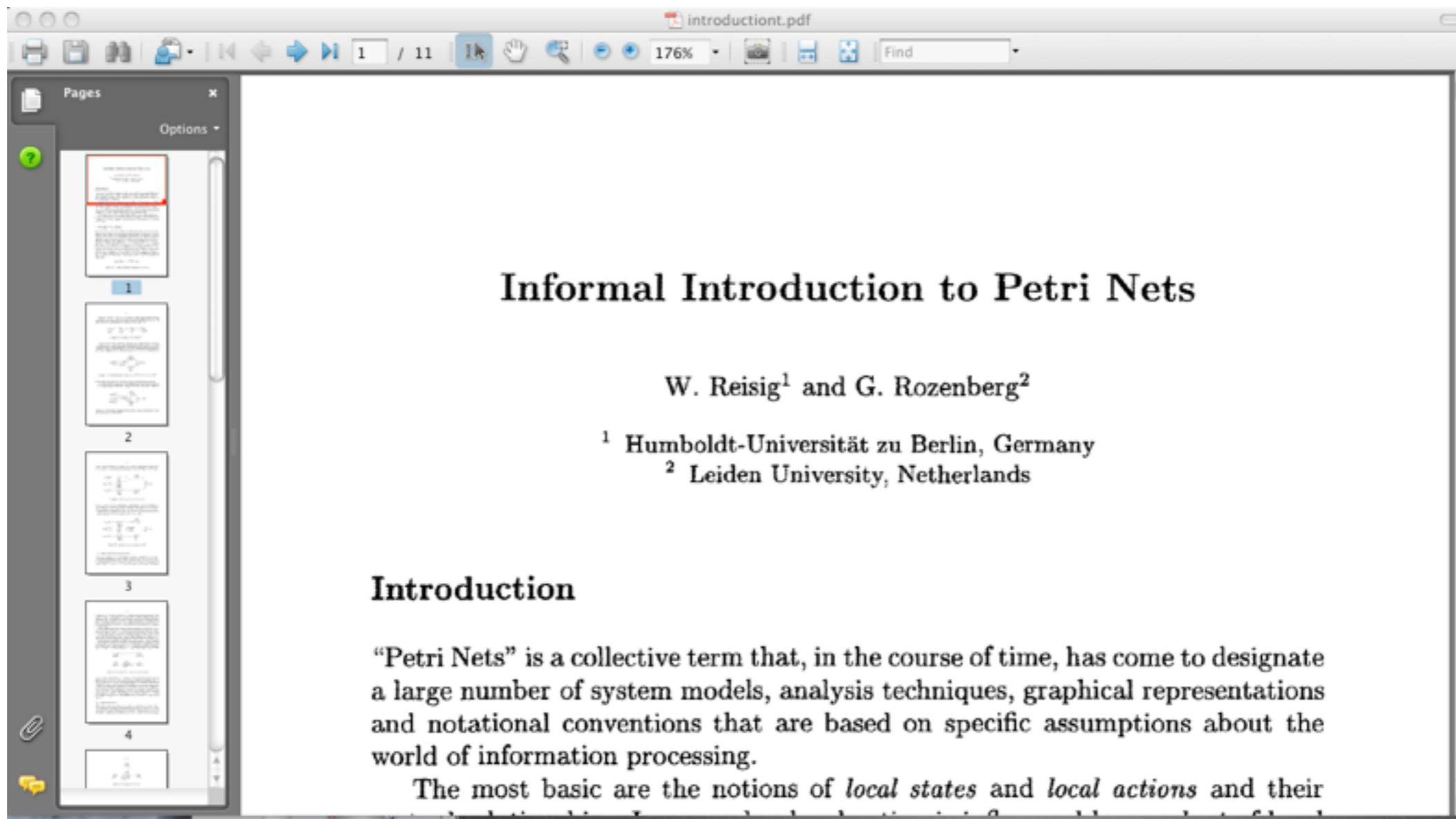
PIPE (Windows, Linux, Mac)

~~HPSIM (Windows)~~

CPN Tools

GHENeSys

# Leitura da semana



The image shows a PDF viewer window titled 'introductiont.pdf'. The main content is the title page of a document. The title is 'Informal Introduction to Petri Nets'. The authors are 'W. Reisig<sup>1</sup> and G. Rozenberg<sup>2</sup>'. The affiliations are '<sup>1</sup> Humboldt-Universität zu Berlin, Germany' and '<sup>2</sup> Leiden University, Netherlands'. The section 'Introduction' begins with the text: '“Petri Nets” is a collective term that, in the course of time, has come to designate a large number of system models, analysis techniques, graphical representations and notational conventions that are based on specific assumptions about the world of information processing. The most basic are the notions of *local states* and *local actions* and their

The left sidebar shows a 'Pages' view with thumbnails for pages 1, 2, 3, and 4. Page 1 is selected. The top toolbar shows navigation and search icons, and the status bar indicates '1 / 11' pages and '176%' zoom.

*Fim*