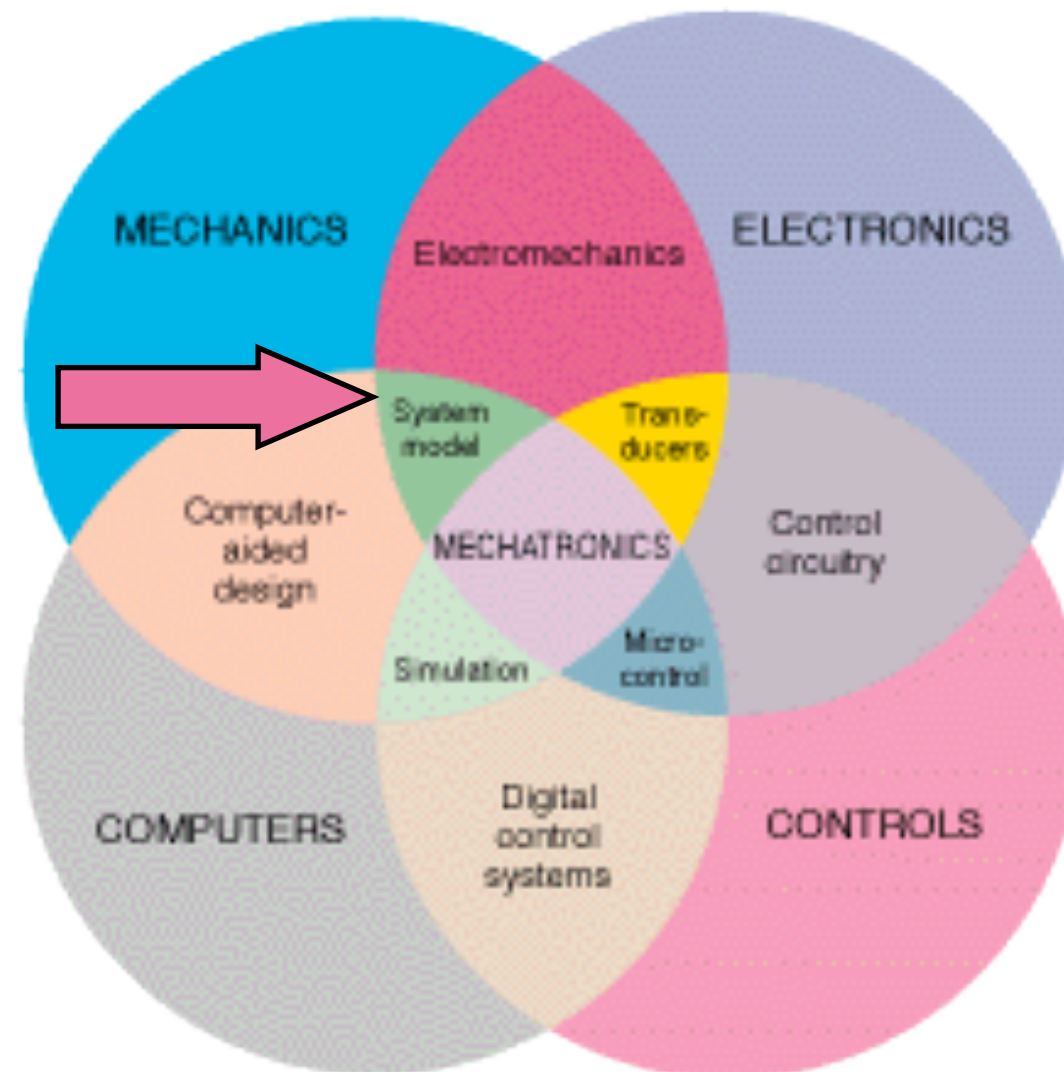


PMR 5237 Modelagem e Design de Sistemas Discretos em Redes de Petri

Aula I: Introdução

Prof. José Reinaldo Silva
reinaldo@usp.br

Contexto da Disciplina



Entre 70 a 80% dos sistemas de automação são discretos



Entre 20 a 30% dos sistemas não-discretos uma parte considerável é discretizável

Histórico das RdP

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

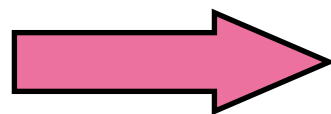


Carl Adam Petri nasceu em Leipzig, em 12 de julho de 1926 e morreu aos 83 anos em 2 de Julho de 2010, deixando um dos maiores legados teórico-científicos do século XX, desenvolvido em pouco mais de 40 anos.

modelagem de processos

2001 UML 2.0 modelagem de requisitos

modelagem de workflow



Schemas

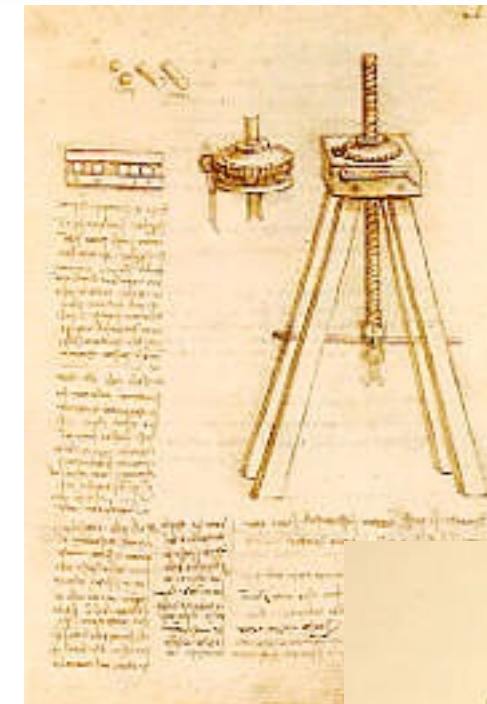
A schema is a cognitive framework or concept that helps organize and interpret information. Schemas can be useful because they allow us to take shortcuts in interpreting a vast amount of information



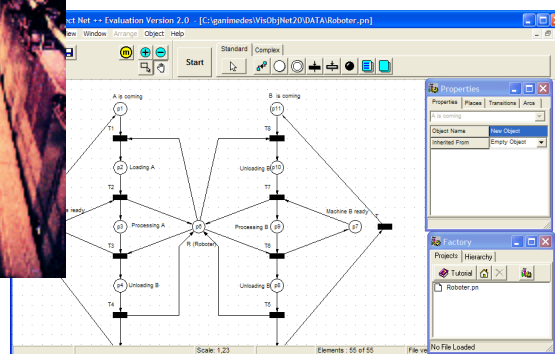
Principais centros de pesquisa em PN do mundo

A “arte” de modelar

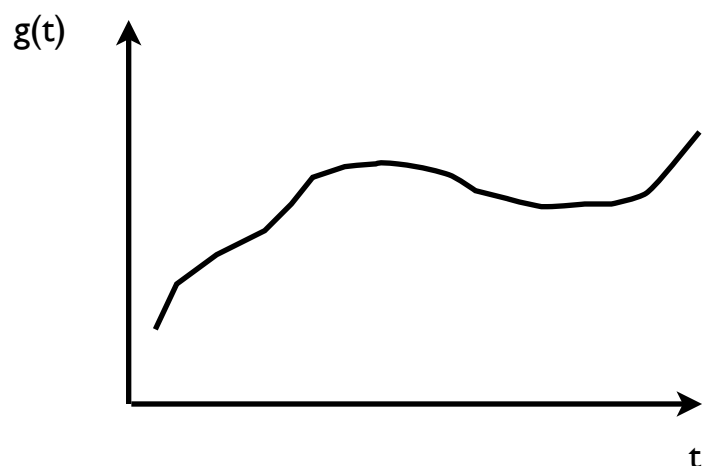
Modelar é produzir uma representação esquemática de um objeto ou sistema capturando suas propriedades mais importantes e a relação entre as partes dinâmicas, geralmente de modo a compactar as informações, aumentar o nível de abstração. Em Engenharia, a maior parte das vezes um modelo é o ponto de partida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



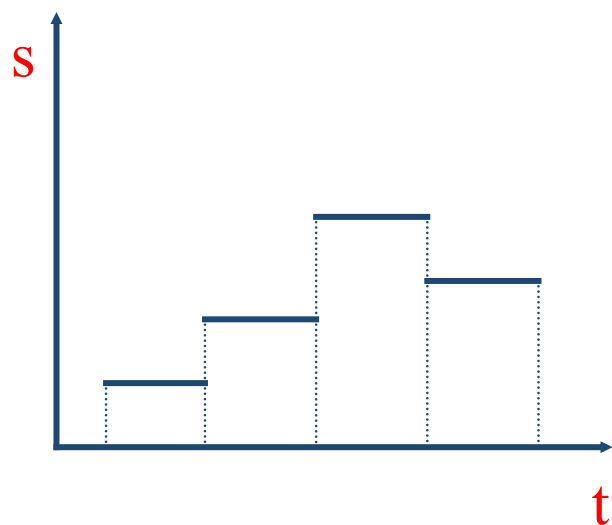
Modelo de Leonardo da Vinci



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. Esta tem pontos de equilíbrio que caracterizam o estado e evoluem no tempo denotando o comportamento do sistema. A dinâmica é caracterizada pela sucessão dos “eventos”.

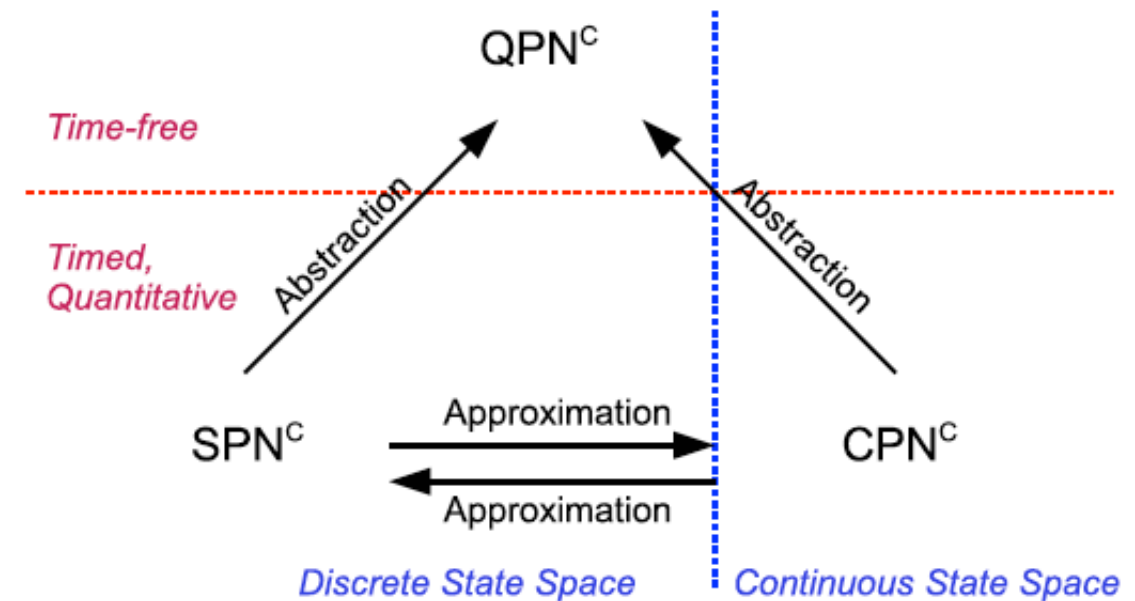


Em alguns casos a *função de estado* é contínua nas variáveis de estado e também no tempo caracterizando trajetórias também contínuas. Em outros casos a função de estado é discreta e evolui aos saltos caracterizando um sistema discreto.

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.

Novas aplicações - mais abrangentes e mais sofisticadas - podem ser associadas, por exemplo, a sistemas complexos de manufatura, “*hazardous systems*”, sistemas de automação inteligente, verificação formal e sistemas biológicos.

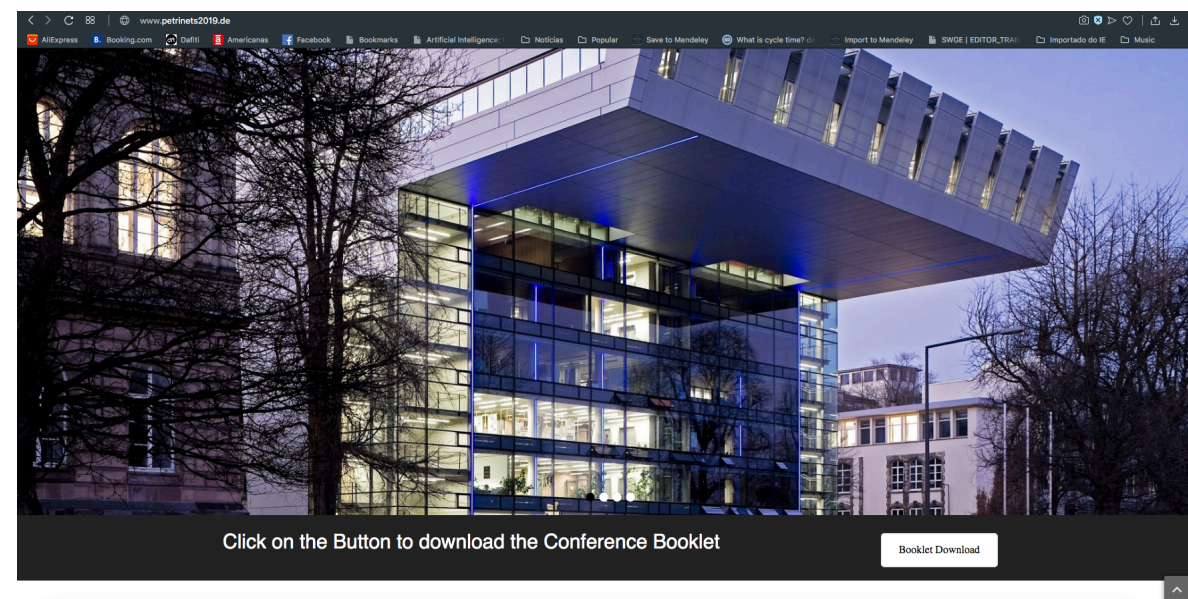


Liu, F. and Heiner, M.; Colored Petri Nets to Model and Simulate Biological Systems, Recent Advances in Petri Nets and Concurrency, S. Donatelli, J. Kleijn, R.J. Machado, J.M. Fernandes (eds.), CEUR Workshop Proceedings, volume 827, ISSN 1613-0073, Jan/2012, pp. 71-85.

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 - por Gerald Estrin, UCLA e Stephen Yau, UIUC. Hoje o congresso mais importante de Redes de Petri no mundo, a 40th Int. Conf. on Application and Theory of Petri Nets and Concurrency (realizada em conjunto com a 19th Int. Conf. on Application of Concurrency to Systems Design), que tem um workshop (desde 2008) dedicado especificamente a Petri Nets on Software Engineering, o PNSE. Ao lado está a página do evento em 2019.

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

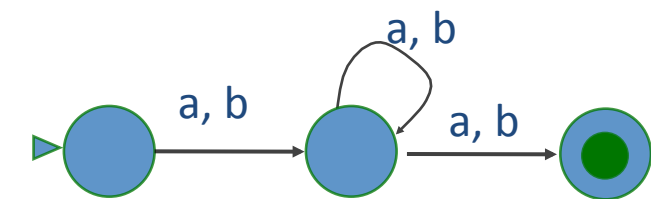


www.petrinets2019.de

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 em Redes de Petri.

Para sistemas discretos, um formalismo alternativo e também muito genérico pode ser usado: a Teoria de Autômatos. Esta representação também se enquadra no paradigma estado transição e pode descrever processos não concorrentes.



CS154 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

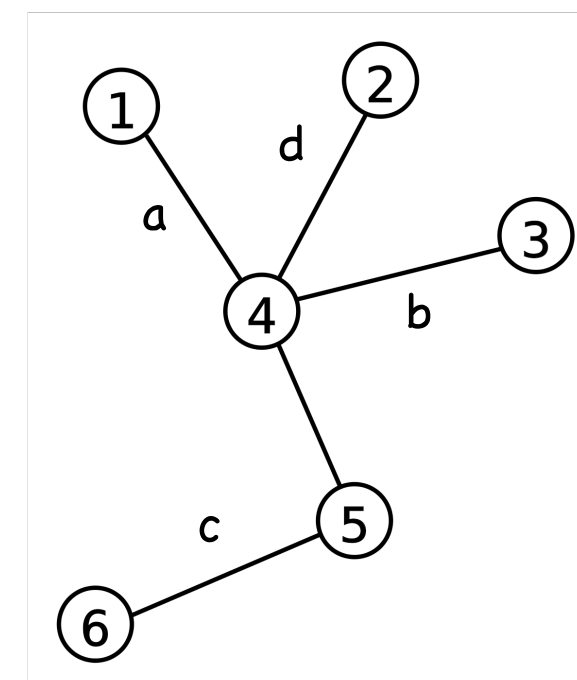
- ◆ Finite automata are finite collections of states with transition rules that take you from one state to another.
- ◆ Original application was sequential switching circuits, where the “state” was the settings of internal bits.
- ◆ Today, several kinds of software can be modeled by FA.

CS154 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

Representing FA

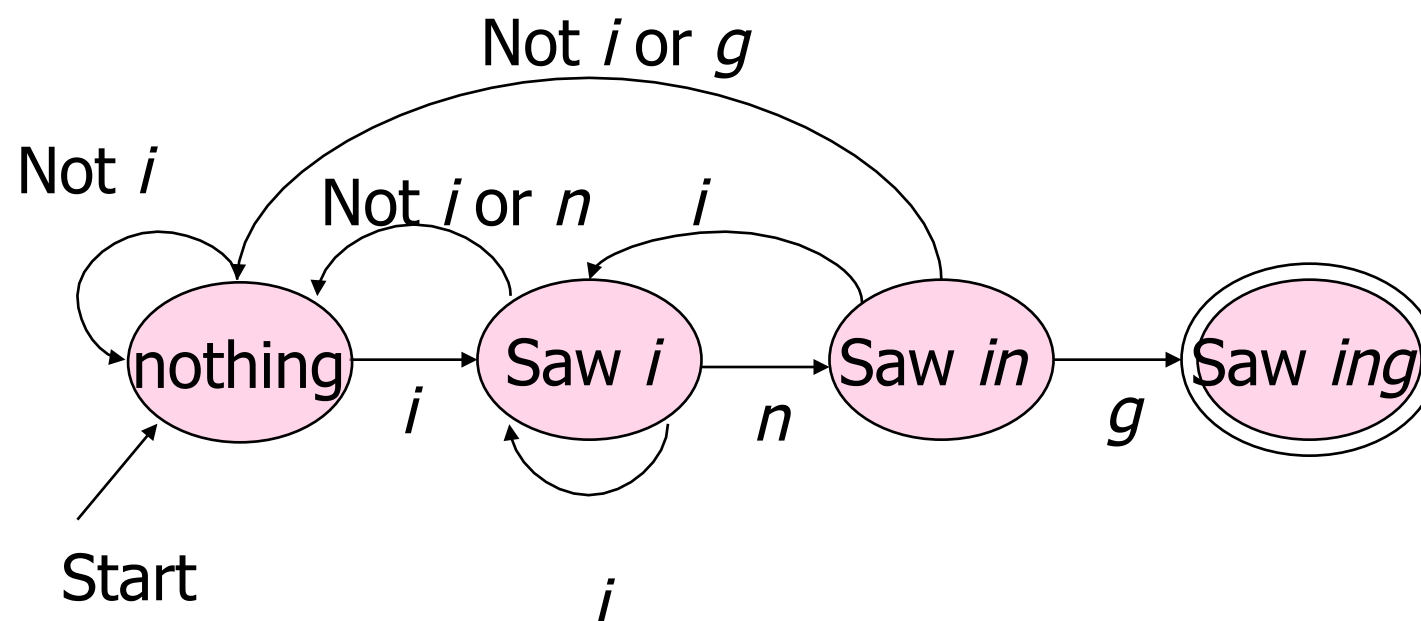
- ◆ Simplest representation is often a graph.
 - ▶ Nodes = states.
 - ▶ Arcs indicate state transitions.
 - ▶ Labels on arcs tell what causes the transition.



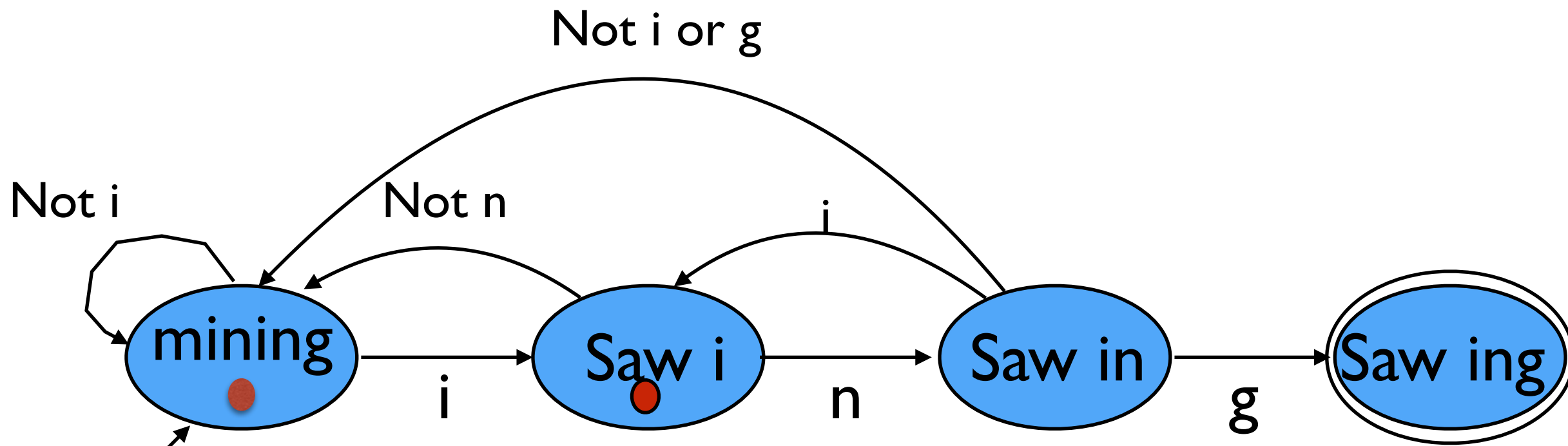
CS154 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

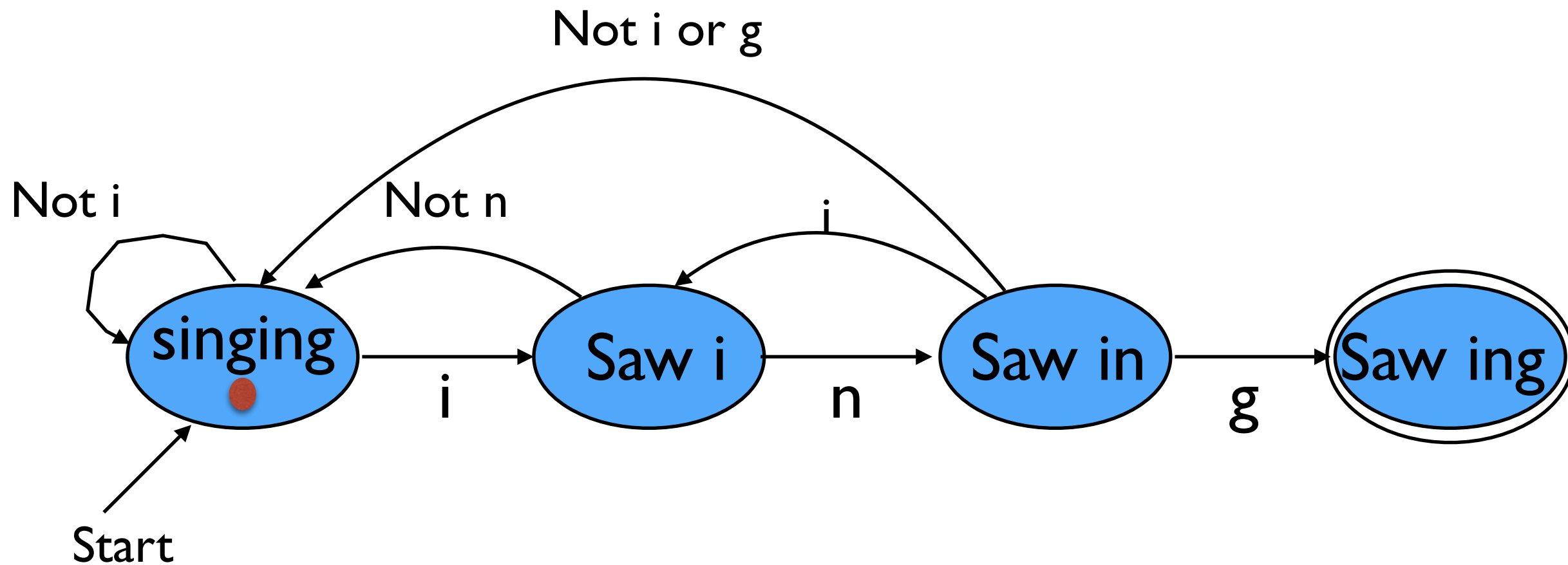
Example: Recognizing Strings Ending in "ing"



Experimente agora fazer o parsing da palavra "mining".



mining



S i n g i n g

O que acontece se fizermos o parsing da palavra "singing"?

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* (Σ , typically).
 3. A *transition function* (δ , 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

CS154 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

CS154 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

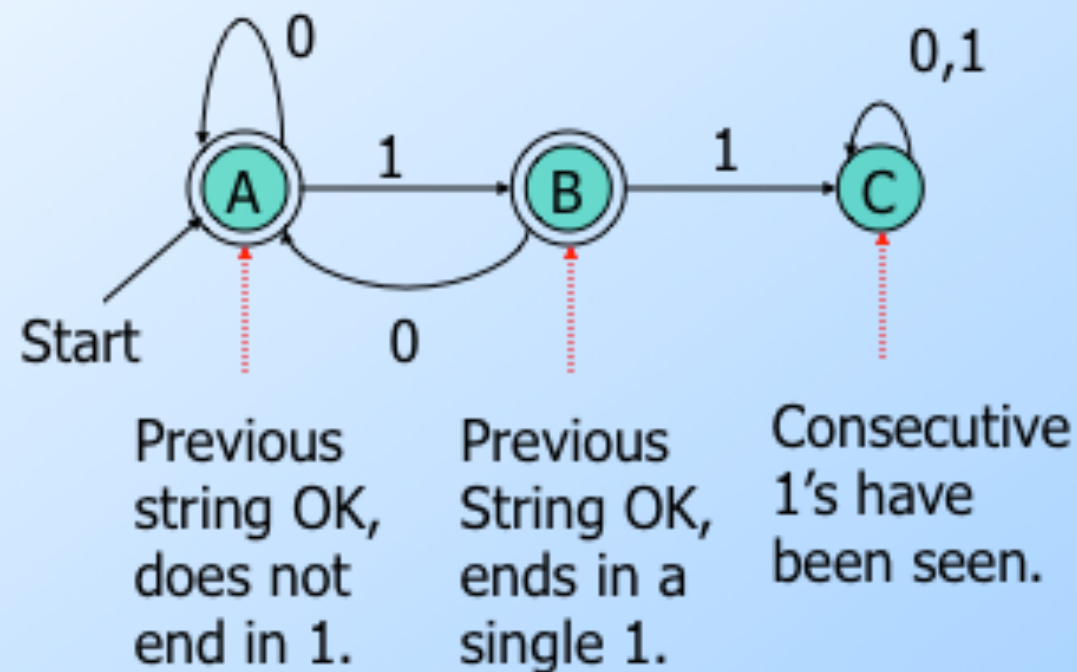
CS154 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

Também chamado Transition Graph

Example: Graph of a DFA

Accepts all strings without two consecutive 1's.

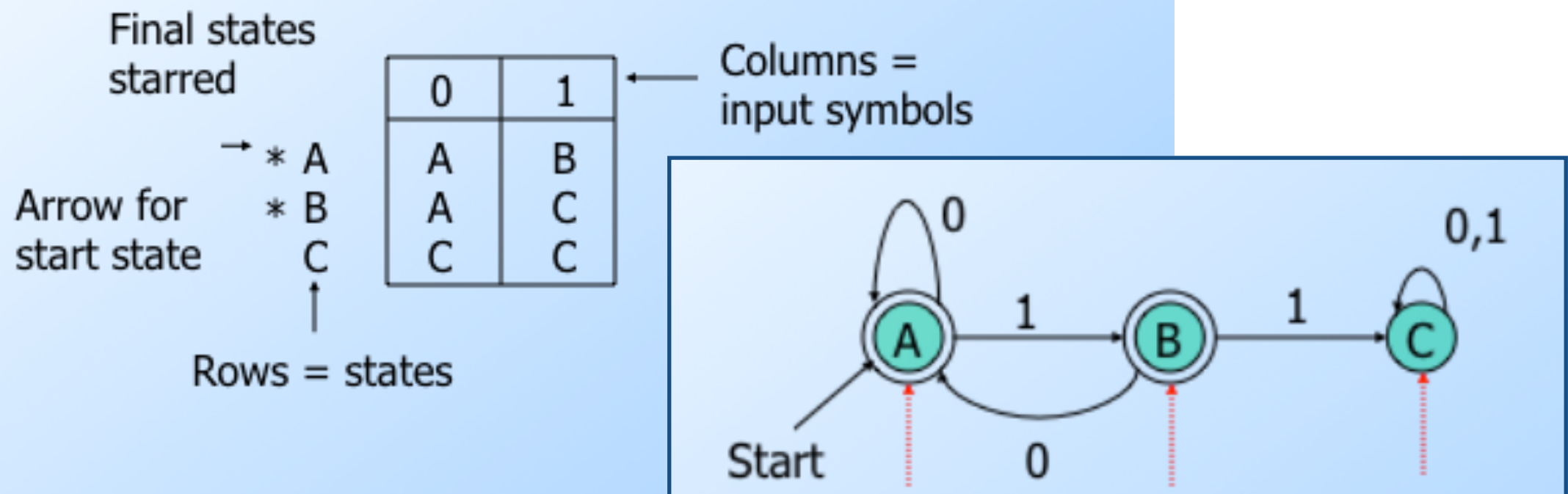


9

CS154 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

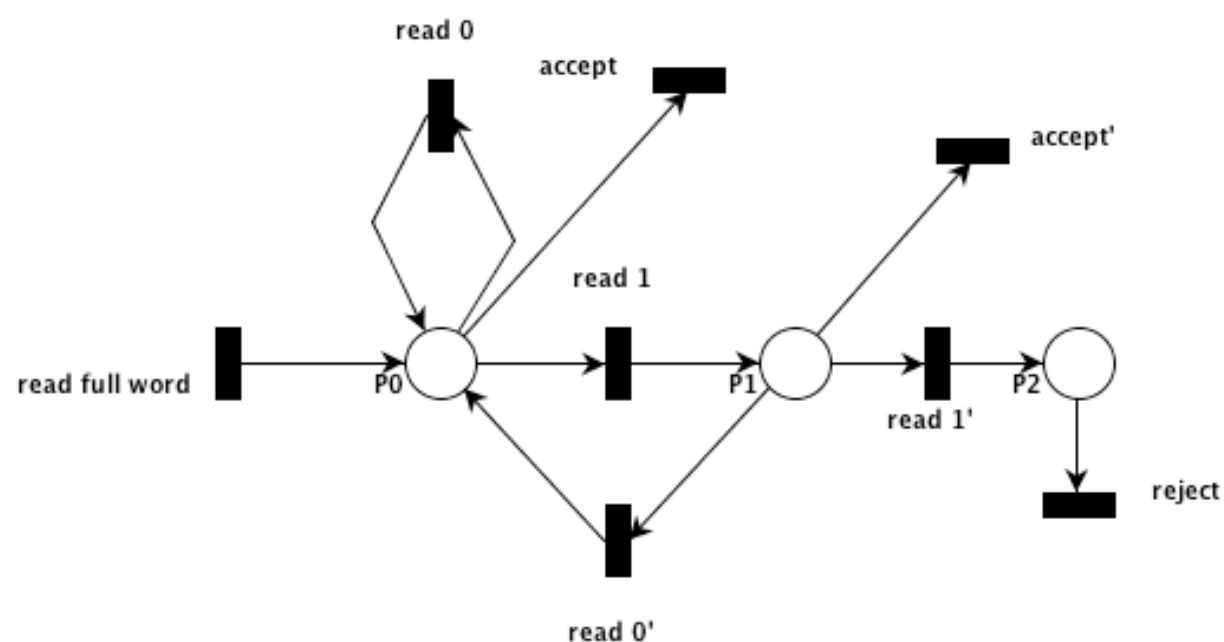
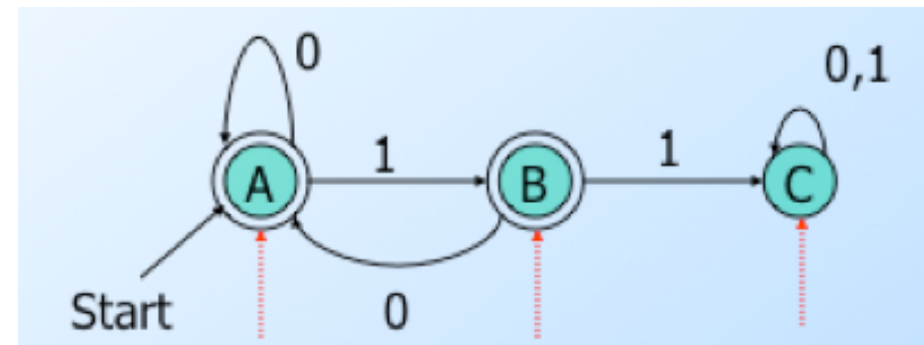
Alternative Representation: Transition Table



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

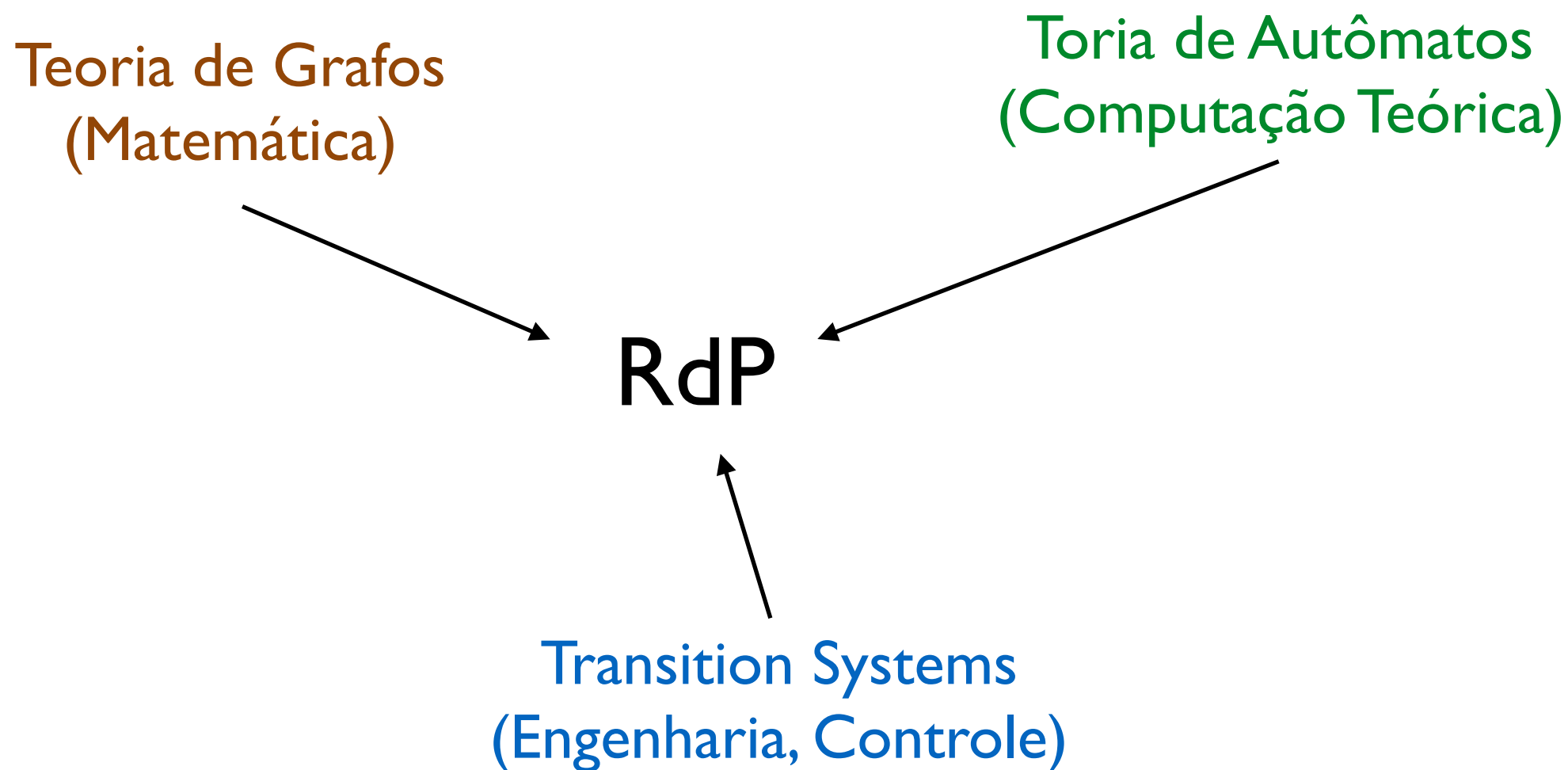
10

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?

Mapa Teórico das Redes de Petri



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 . Estes conjuntos satisfazem à relação,

Disjunção entre estados e transições

- $S \cap T = \emptyset$.

Portanto estados e transições são ambos elementos distribuídos:

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

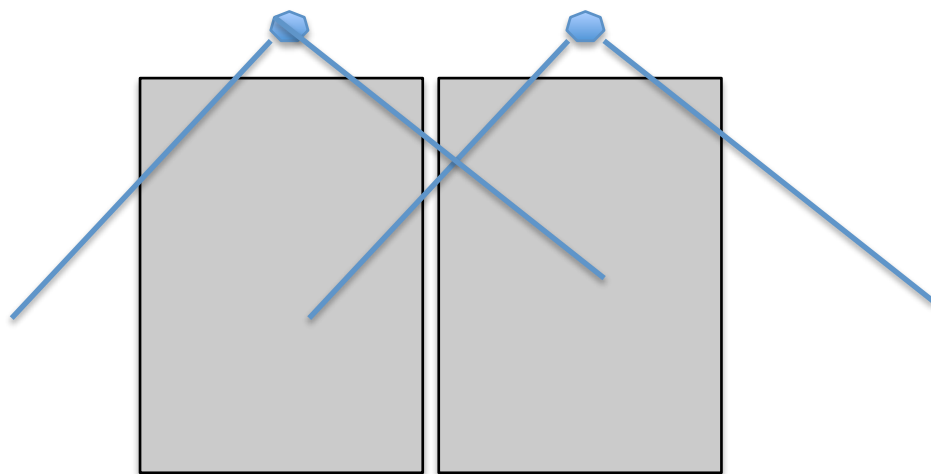
- $\{s_1, s_2, \dots, s_n\} \implies \textit{case}.$

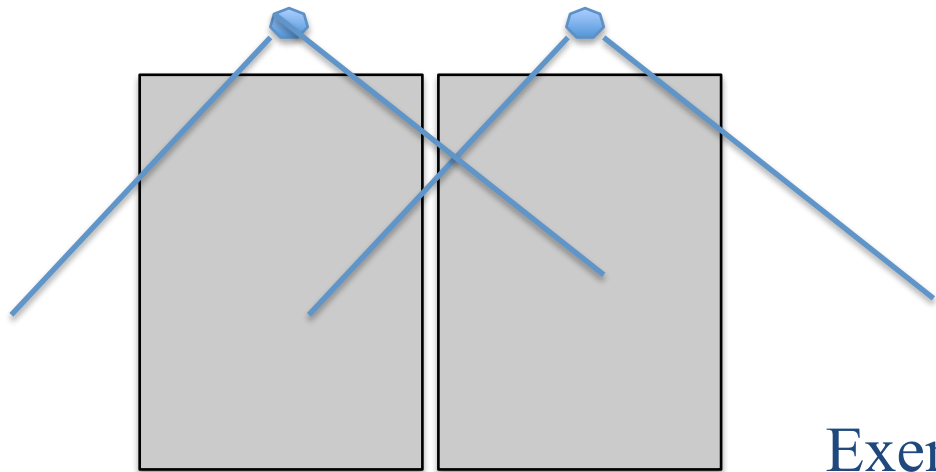
Uma transição distribuída é dada por um conjunto de transições válidas simultaneamente, isto é,

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

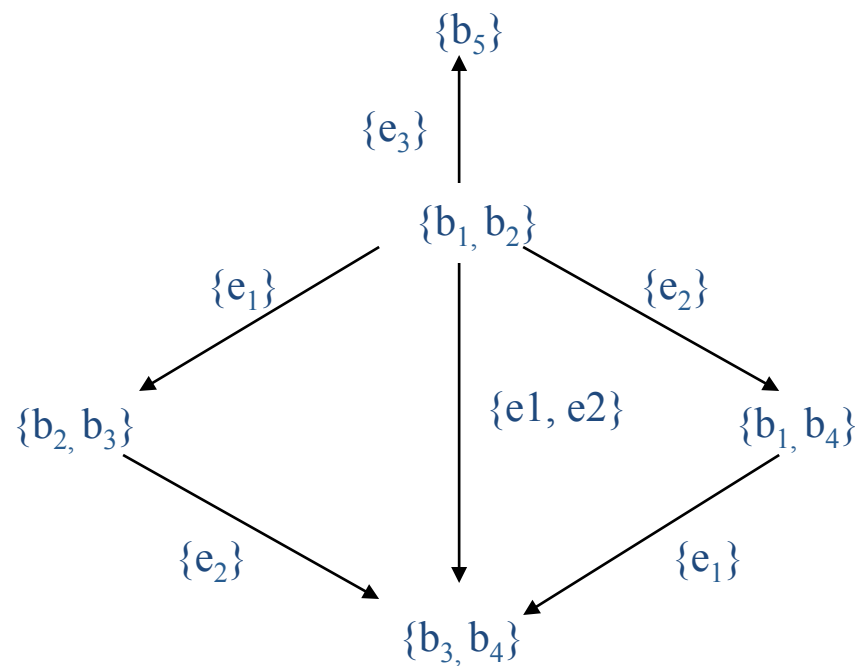
Um exemplo simples do que estamos falando, aplicado a sistemas de automação é a representação do comportamento de uma porta automática, que aciona motores para abertura e fechamento dependendo de sinal de sensores.

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.

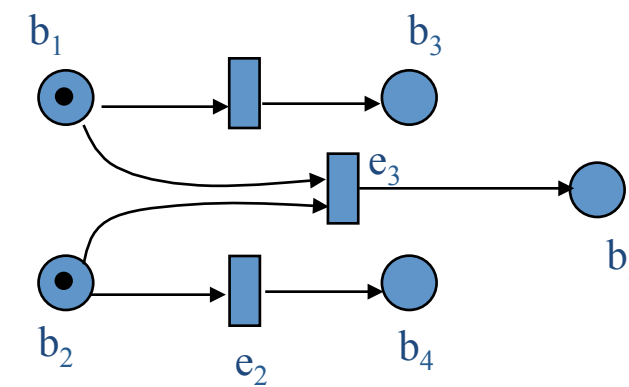




Exemplo (Thiagarajan) :



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 :

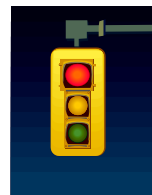
- o princípio da dualidade
- o princípio da localidade
- o princípio da concorrência
- o princípio da representação gráfica
- o princípio da representação algébrica

O princípio da Dualidade

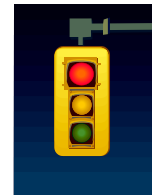
Existem dois tipos de elementos na rede: os elementos ditos passivos ou P-elementos (S-elementos), que representam estados, e os elementos ditos ativos ou T-elementos que representam mudança de estado ou transição.

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;

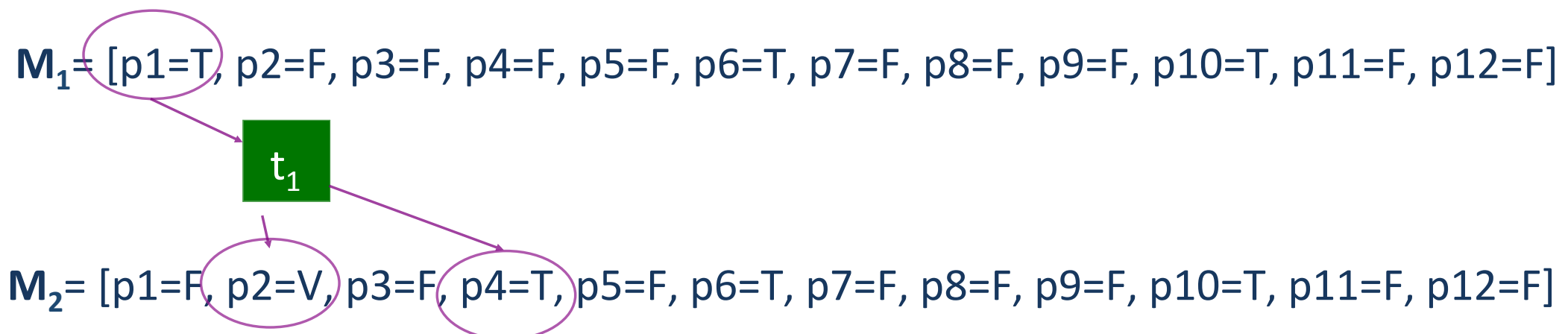
p_9 = sinal de largada para o carro B enviado;

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

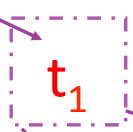


p_1 = carro A: preparando-se para começar;
 p_2 = carro A: esperando o sinal de largada;
 p_3 = carro A: correndo;
 p_4 = sinal de prontidão do carro A enviado;
 p_5 = sinal de largada para o carro A enviado;
 p_6 = operador: esperando sinal de prontidão dos pilotos;
 p_7 = operador: sinal de largada 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;

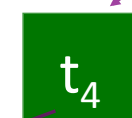
t_1 = carro A: envia sinal de prontidão
 t_2 = carro A: acelera
 t_3 = operador: manda sinal de largada
 t_4 = carro B: envia sinal de prontidão
 t_5 = carro B: acelera

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;

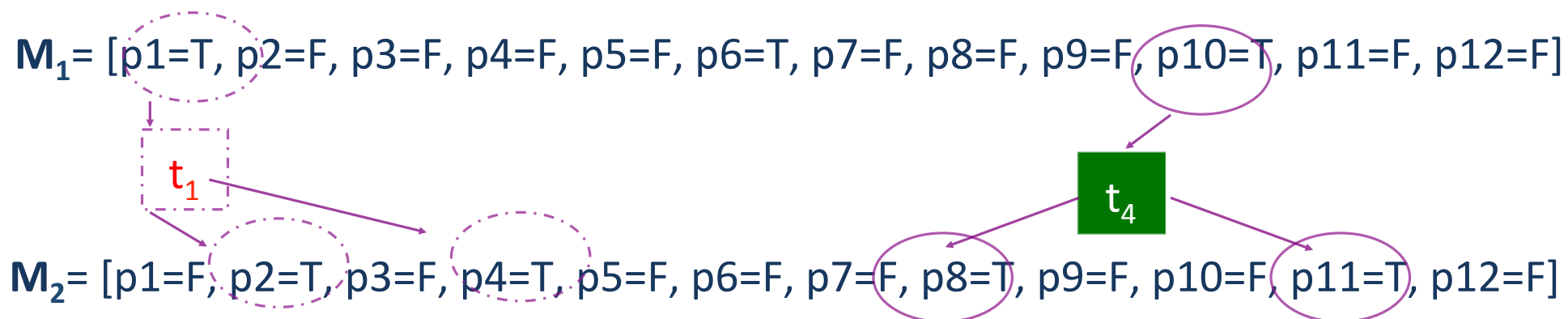
- t_1 = carro A: envia sinal de prontidão
- t_2 = carro A: acelera
- t_3 = operador: manda sinal de largada
- t_4 = carro B: envia sinal de prontidão
- t_5 = carro B: acelera

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

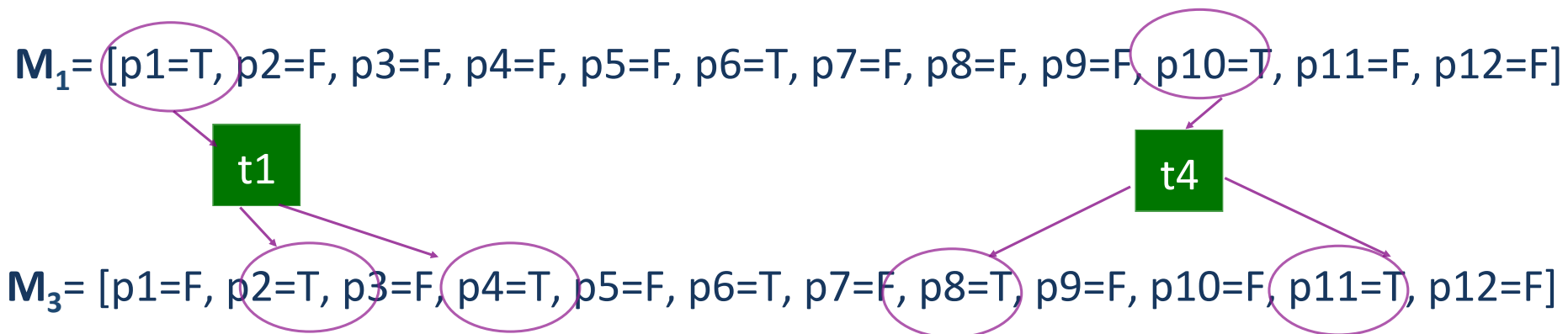
Pelo princípio da localidade o comportamento das transições depende unicamente da sua localidade.



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 ditas concorrentes se e somente se possuem localidades disjuntas (são independentes), isto é:

$$Indep(t_1, t_2) \Leftrightarrow Loc(t_1) \cap Loc(t_2) = \phi.$$

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

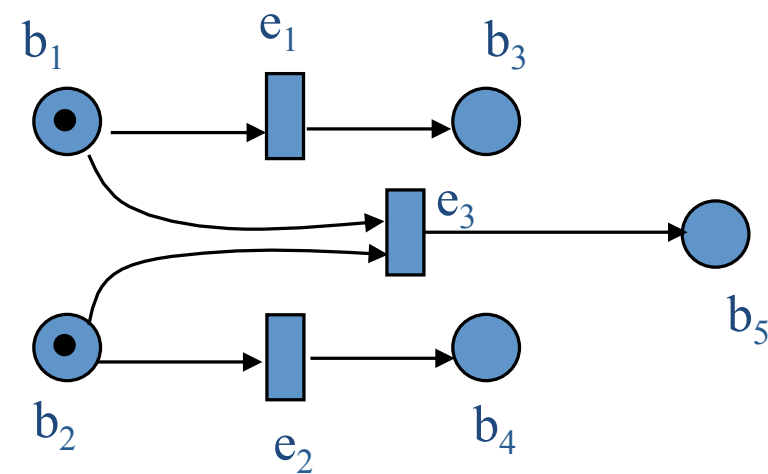


- Estados e transições são noções distintas porém intercaladas;
- Ambos, estados e transições são entidades distribuídas;
- A extensão das mudanças causadas por uma transição é restrita e não depende do estado em que esta ocorre;
- Uma transição está *habilitada* em um estado sse as mudanças associadas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

Exemplo:

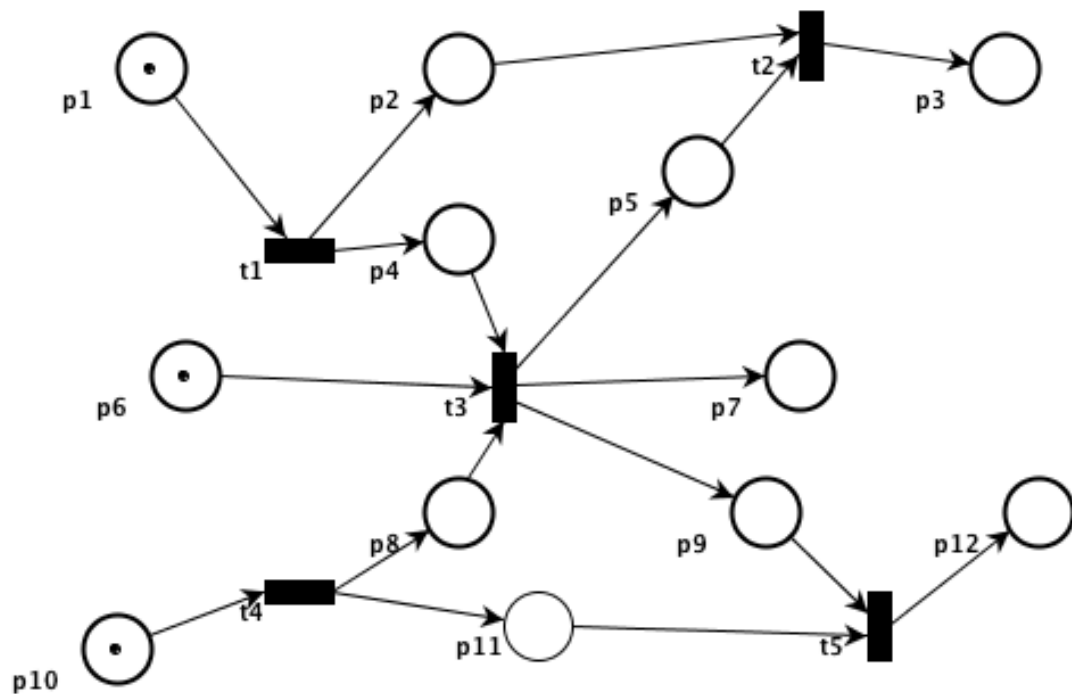
Usando o mesmo exemplo da porta automática podemos construir a representação algébrica (matriz de incidência) desta rede.

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 Fórmula I



Incidence & Marking

Source net

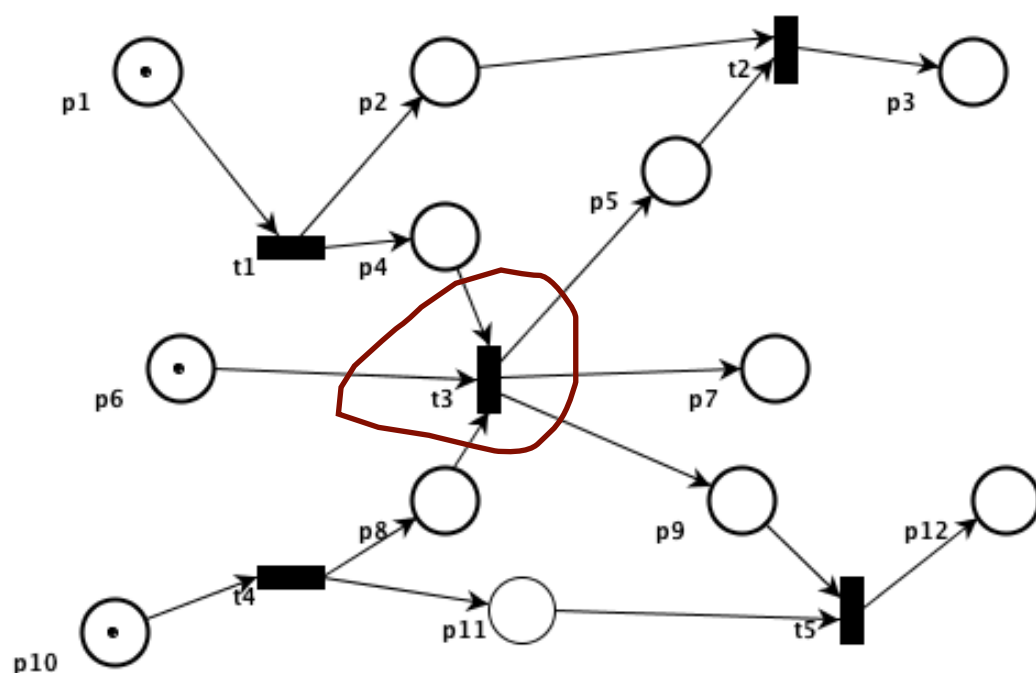
Use current net Filename:

Results

Combined incidence matrix I

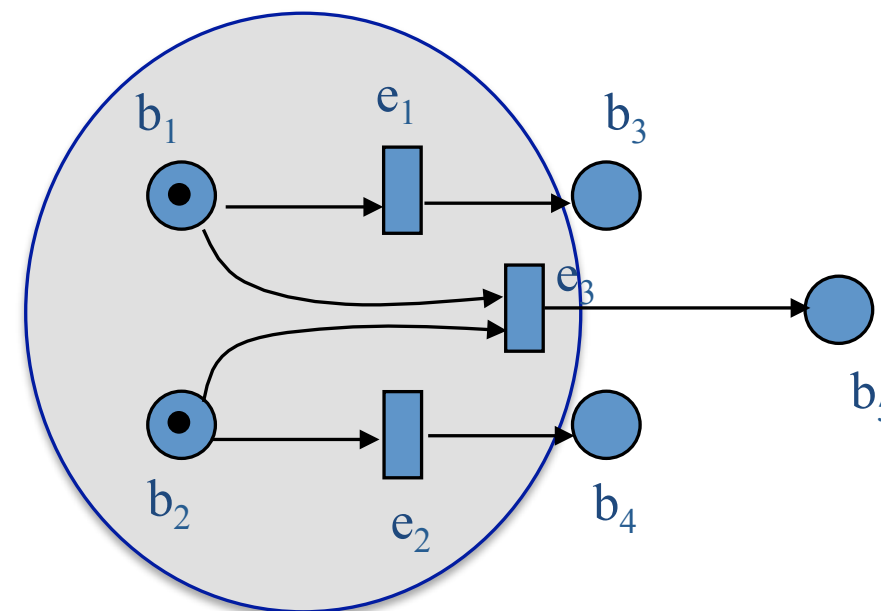
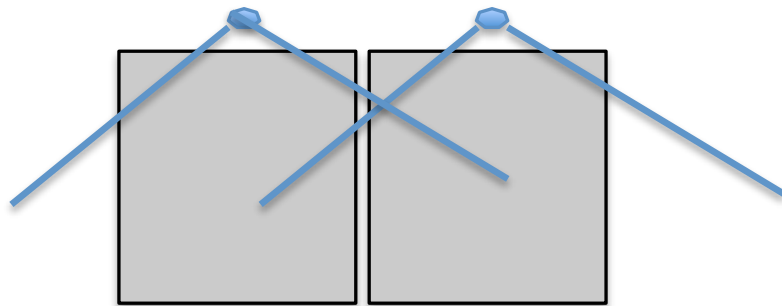
	t1	t2	t3	t4	t5
p1	-1	0	0	0	0
p2	1	-1	0	0	0
p6	0	0	-1	0	0
p7	0	0	1	0	0
p4	1	0	-1	0	0
p3	0	1	0	0	0
p5	0	-1	1	0	0
p10	0	0	0	-1	0
p11	0	0	0	1	-1
p8	0	0	-1	1	0
p9	0	0	1	0	-1
p12	0	0	0	0	1

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.

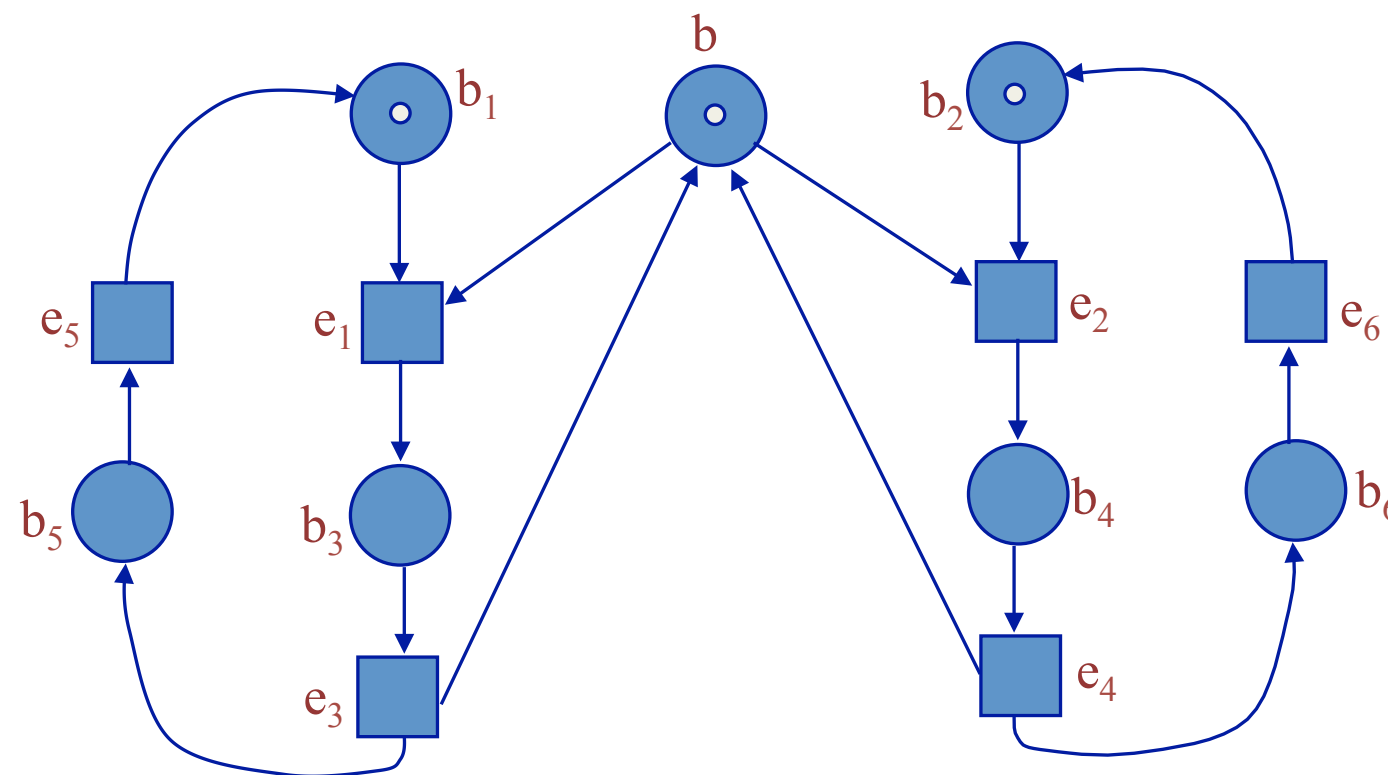
Configurações especiais: conflito



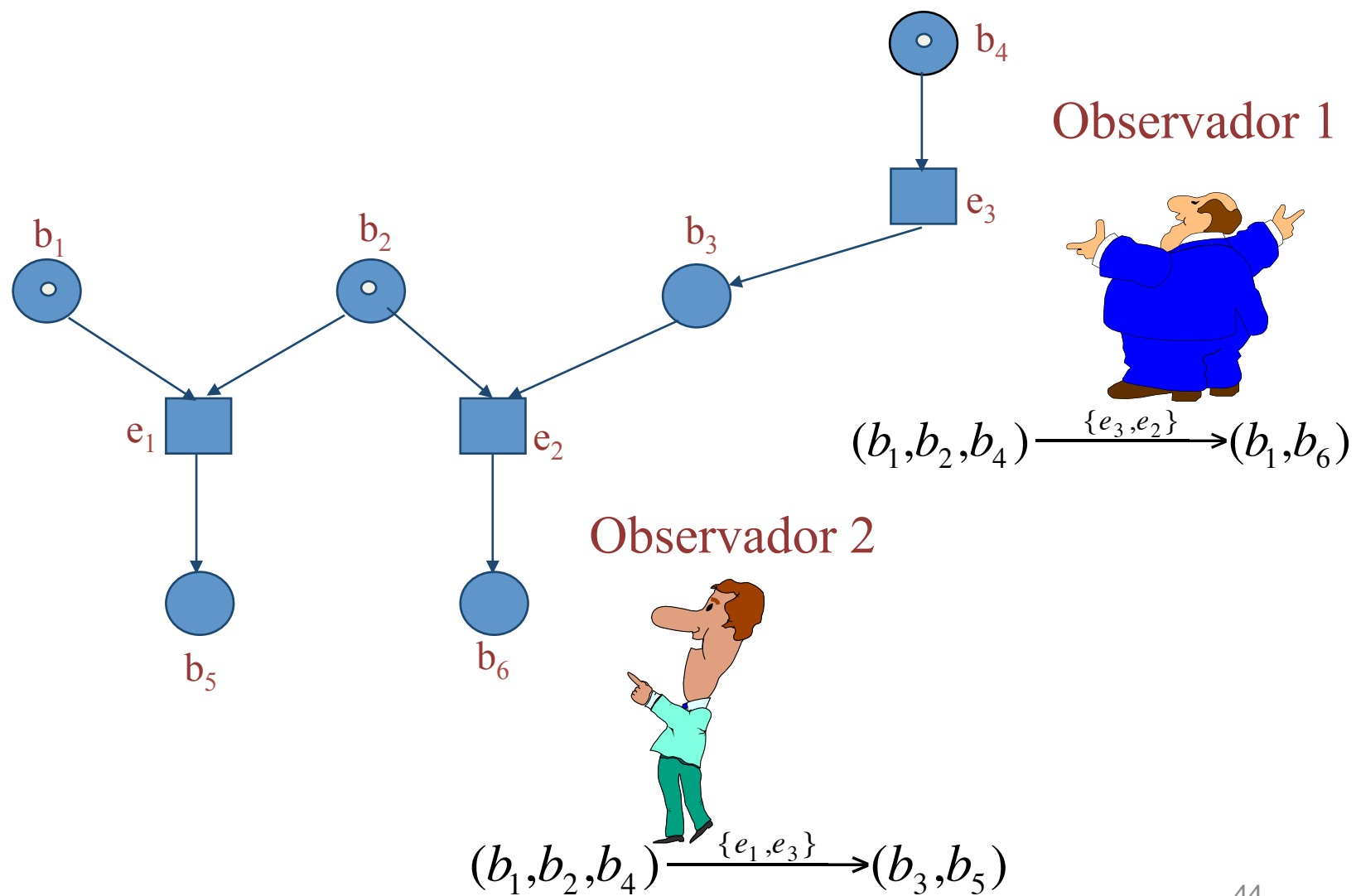
O Conflito denota uma situação de exclusão mútua (mutex), onde a ocorrência de uma transição inibe outras. No caso do exemplo as transições $\{e_1, e_2\}$ são mutuamente exclusivas à transição $\{e_3\}$.

Semáforo: o mutex clássico

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

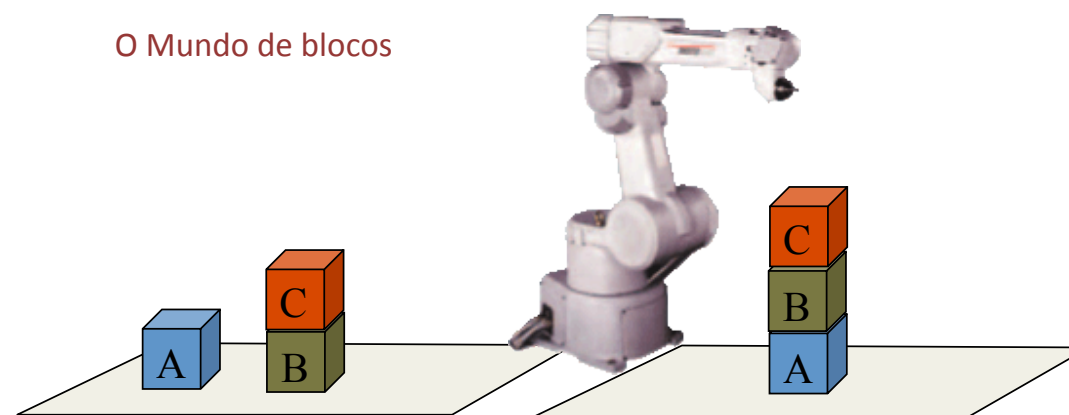


Configurações especiais: contato ou confusão



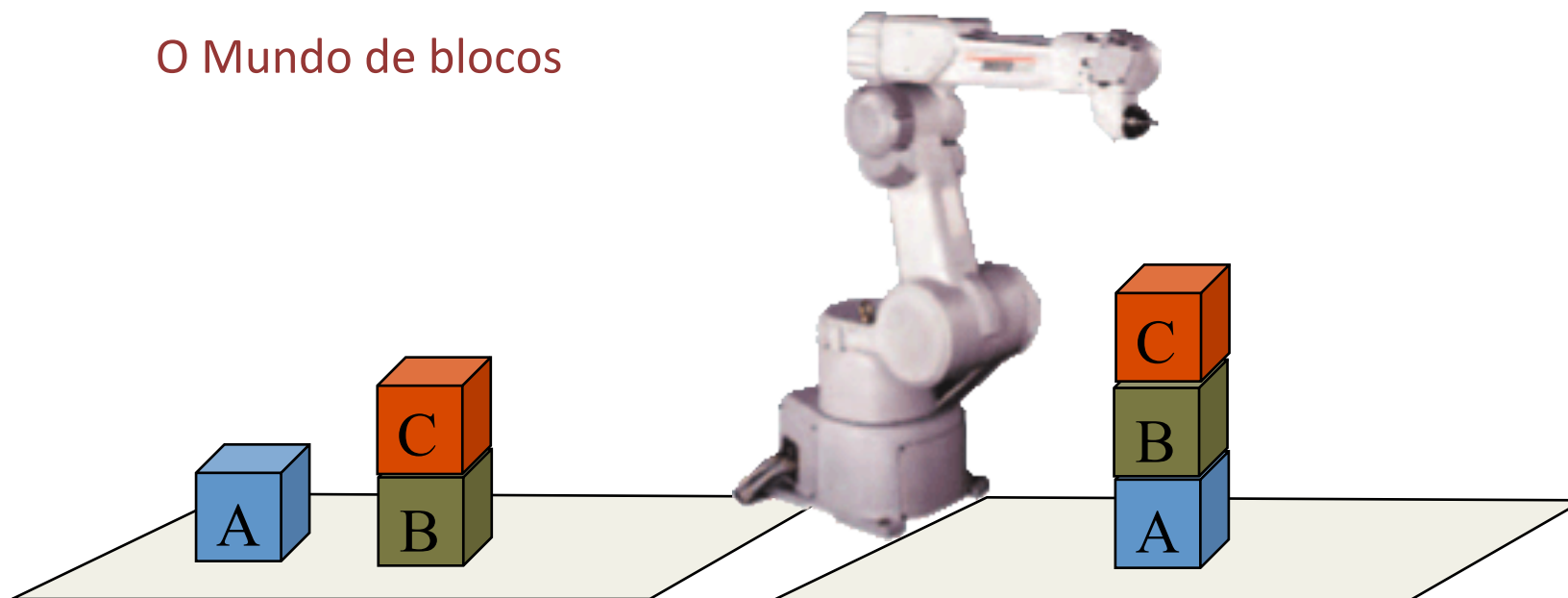
IA Planning: o STRIPS

O sistema STRIPS é a estratégia de resolução de problemas mais usada em planning. Note-se que é uma estratégia baseada no método estado-transição e por isso é passível de ser analisada em Redes de Petri. O problema modelo mais conhecido resolvido com o sistema STRIPS é o problema do mundo de blocos.

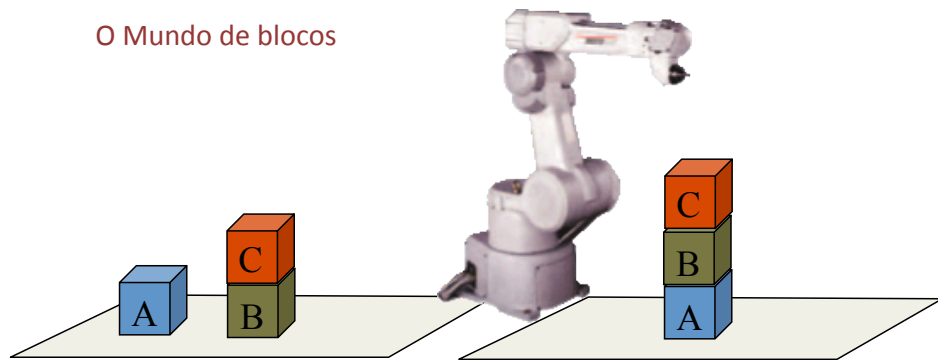


O exemplo mais simples e intuitivo do sistema STRIPS é o chamado “mundo de blocos” que consiste em mudar blocos de configuração usando robôs.

O Mundo de blocos

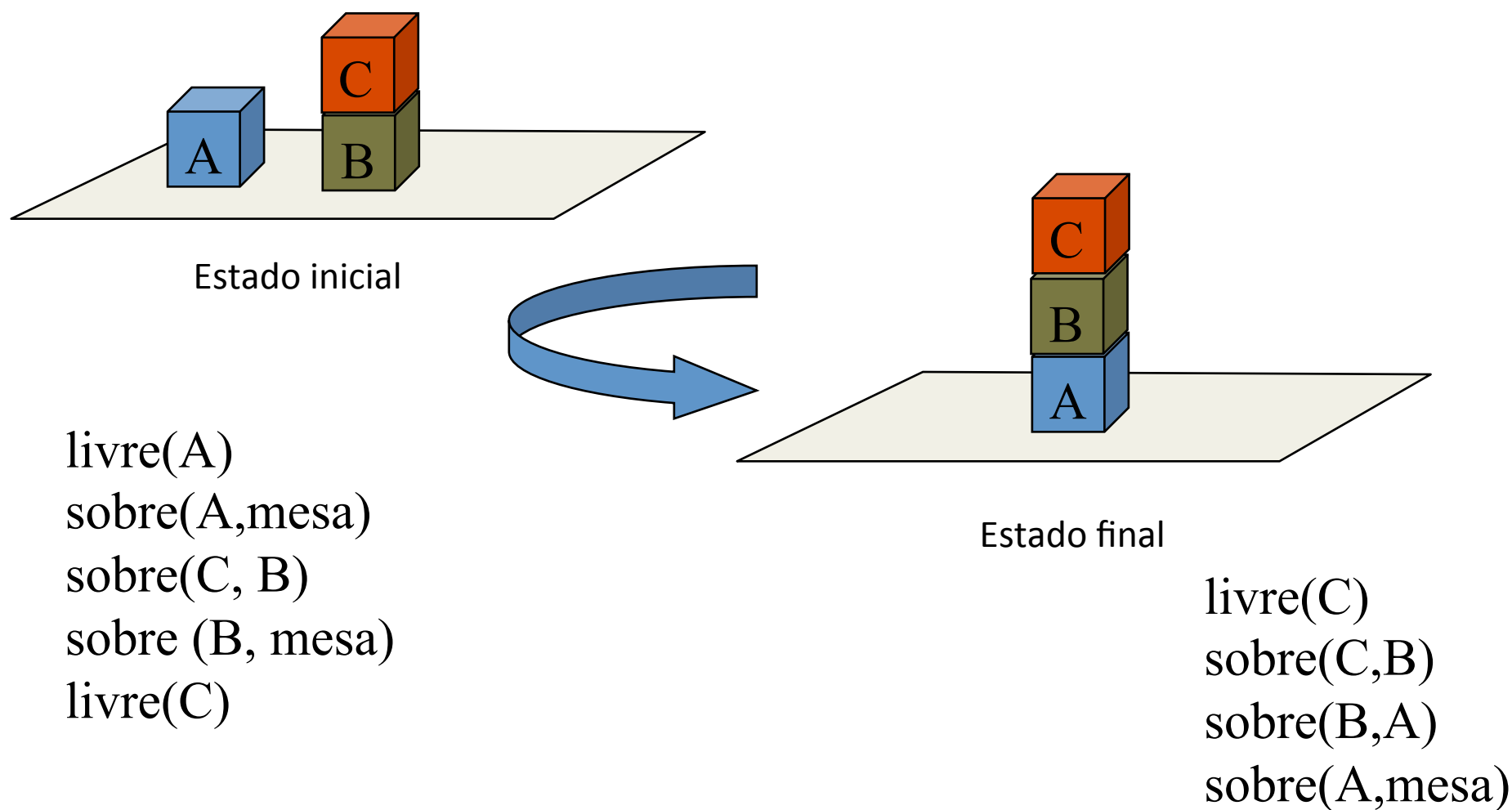


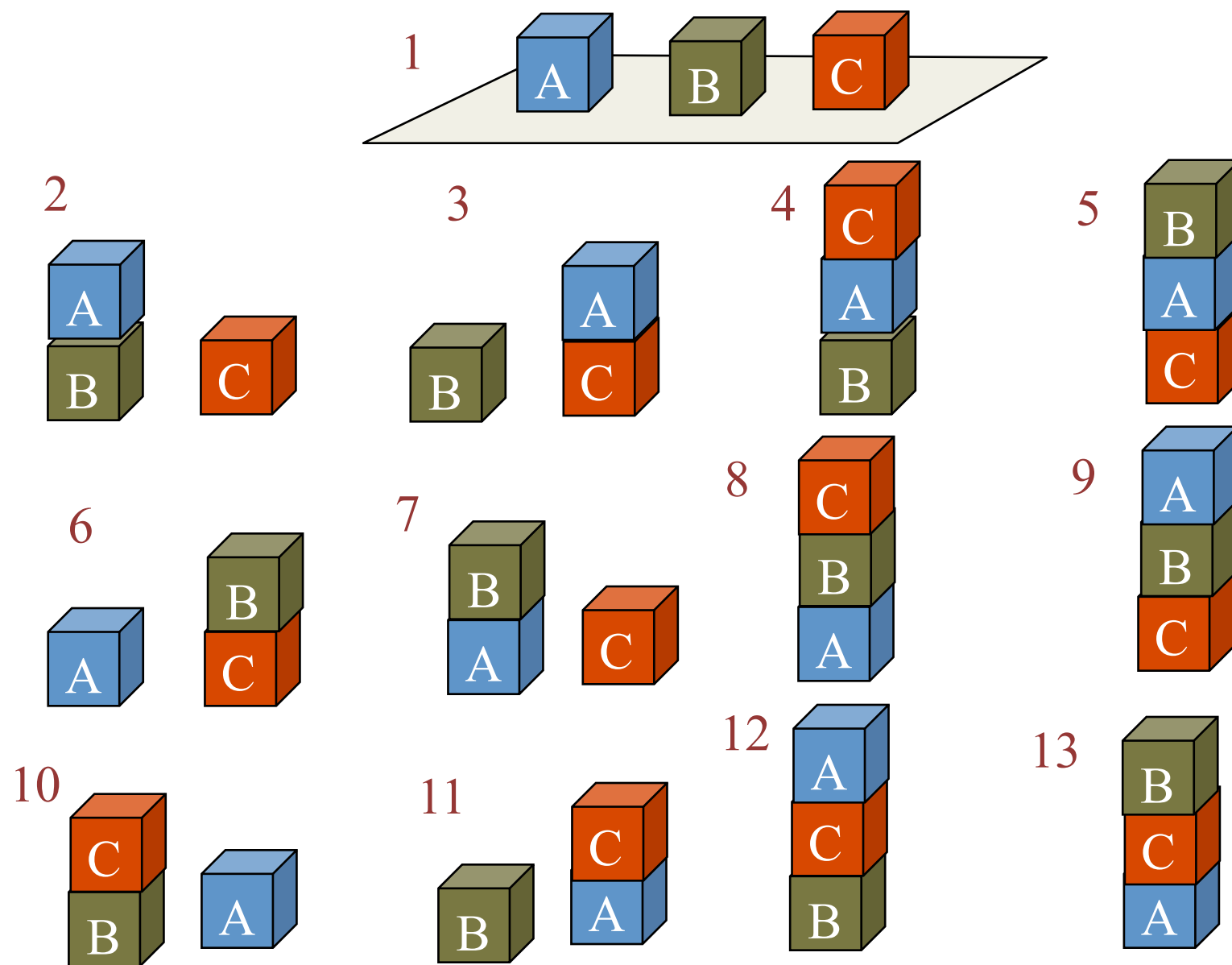
O Mundo de blocos

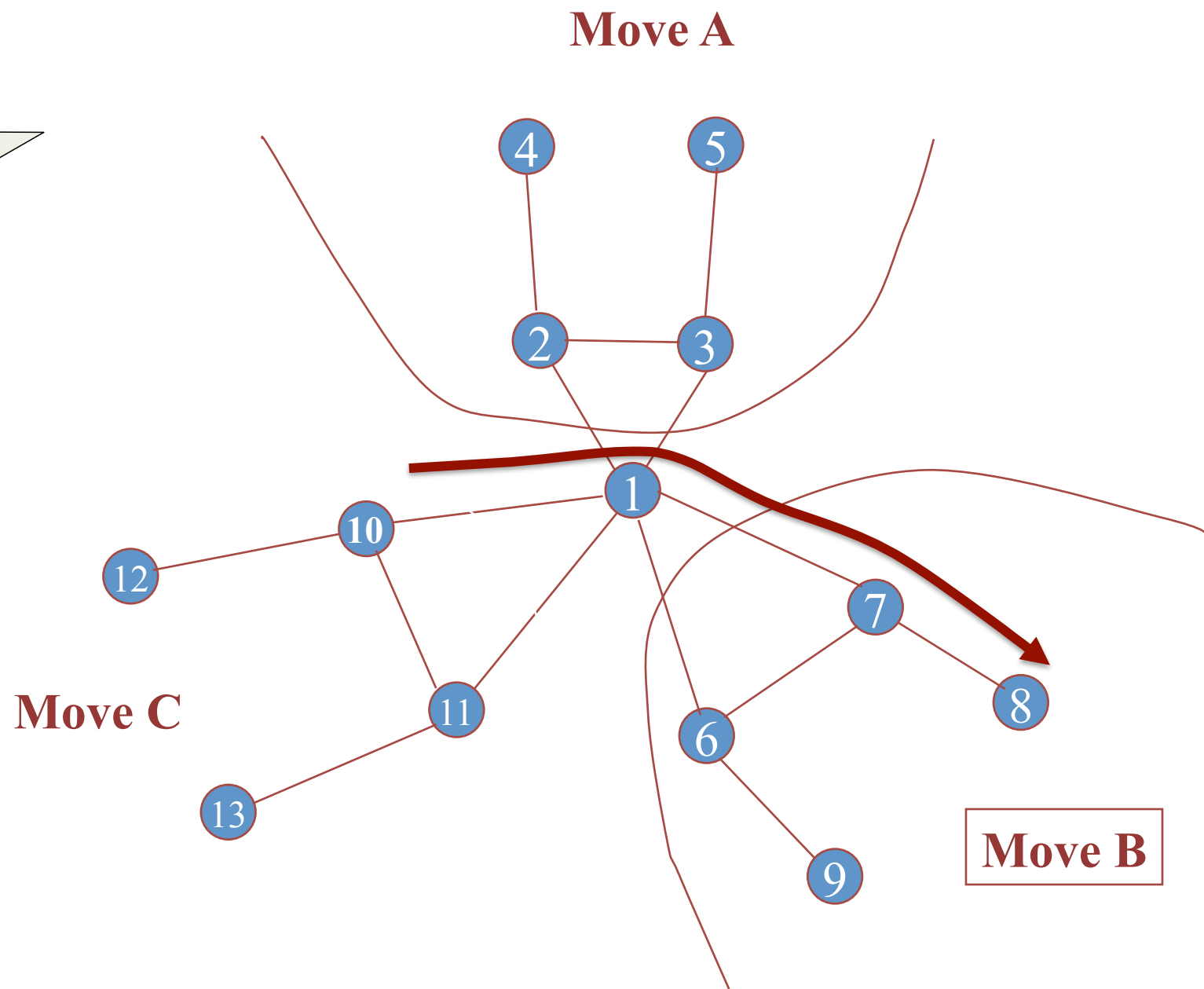
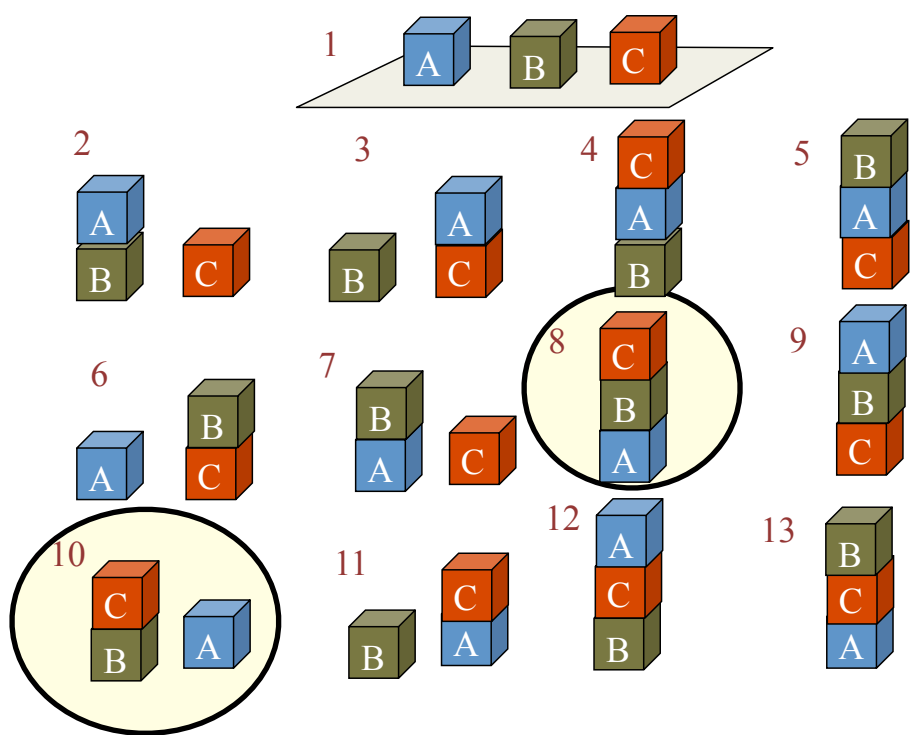
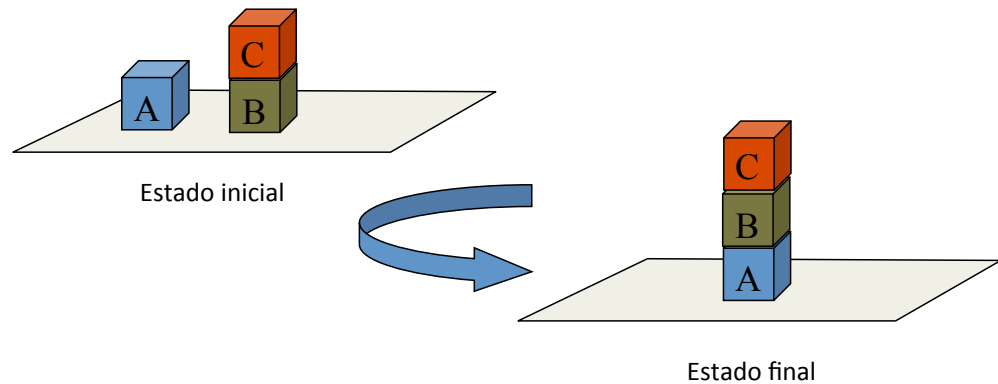


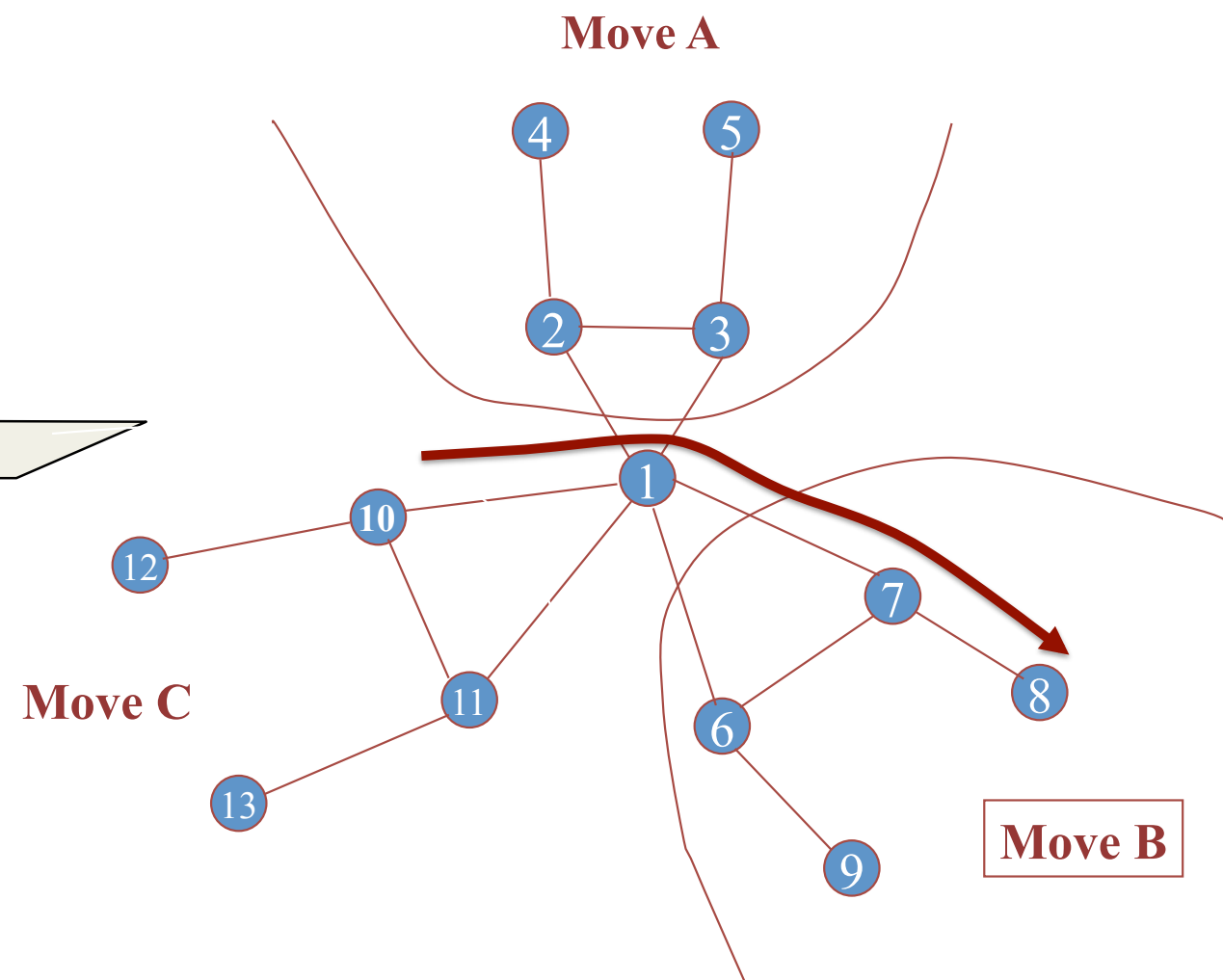
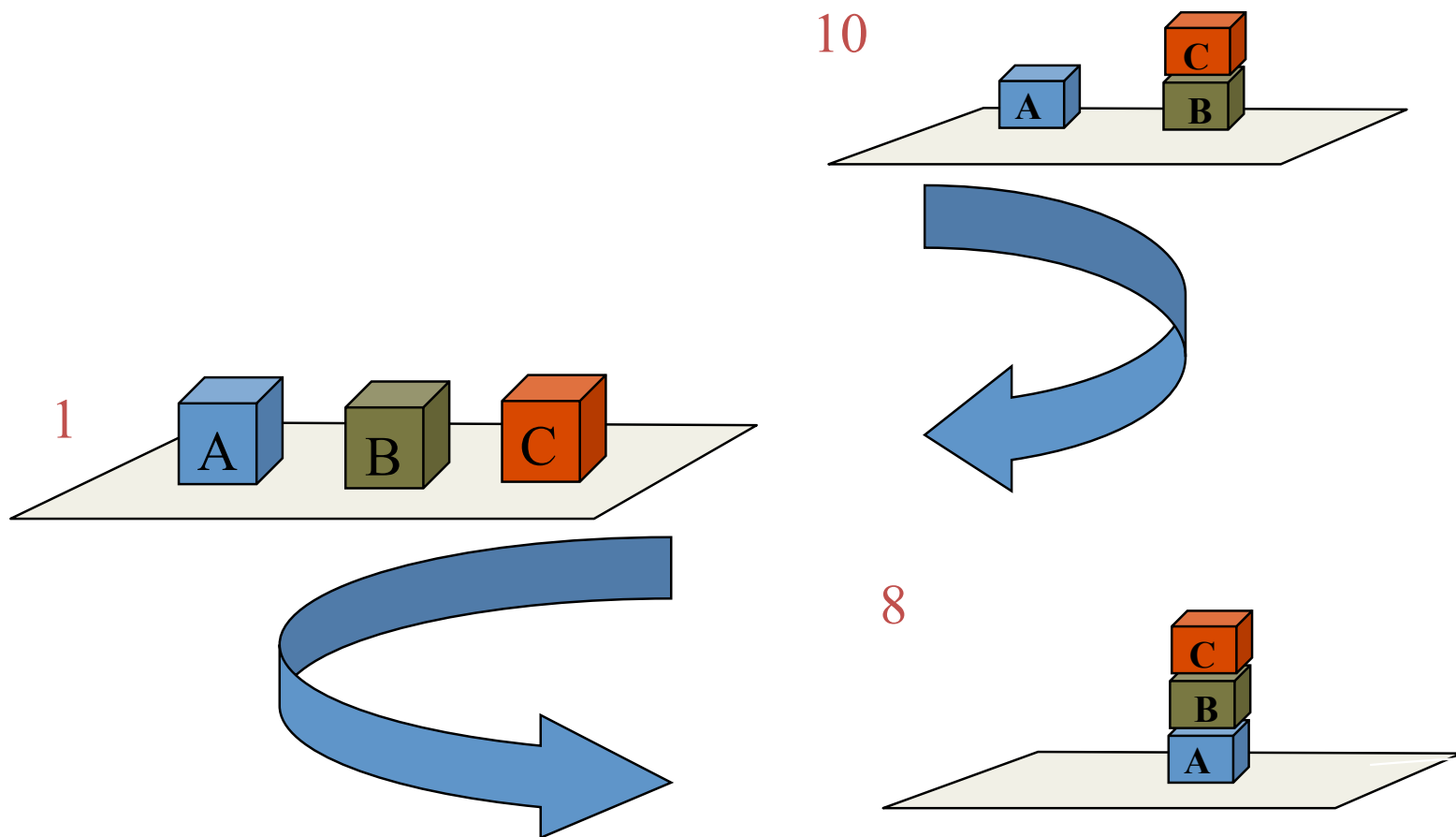
O exercício consiste em analisar a dinâmica deste sistema. Vamos portanto separar a “planta” do “controle”, onde a planta são os possíveis estados do “mundo de blocos” e que movimento é preciso fazer para passar de um estado para o outro. Existe um agente dinâmico capaz de operar a planta, isto é, capaz de, dentro de algumas condições realizar uma transição. Este agente é o robô. No momento vamos apenas modelar a “planta”.

Um plano é a solução de um problema composto de um estado inicial, um estado final, e uma sequência de ações (ou um passo) que transforma o estado inicial no estado final, ou, em outras palavras que os coloca na mesma localidade









PMR 5237 em 2019

Introdução à modelagem de sistemas discretos, padrão ISO/IEC 15.909. Redes Lugar/Transição; propriedades das RdP. Redes de alto nível (HLPN) e redes coloridas. Análise formal das propriedades estruturais e comportamentais. Verificação formal. Validação. Aplicações: sistemas de manufatura, análise de requisitos, design de sistemas.

Bibliografia

Girault, C., Valk, R.; Petri Nets for Systems Engineering: A Guide to Modeling, Verification and Applications, Springer, 2003.

Reisig, W.; Understanding Petri Nets: modeling techniques, Analysis methods, case studies, Springer, 2013.

Murata, T.; Petri nets: properties, analysis and applications. Proceedings of IEEE, 77(4), pp. 541-580, 1989.

Silva, M.; Half a century after Carl Adam Petri's Ph.D. thesis: A perspective on the field, Annual Review in Control, Elsevier, 37, pp. 191-219, 2013.

Wang, J. (ed.); Handbook of Finite State Based Models and Applications, CRC Press, Taylor & Francis, 2013.

PMR 5237

1o. ciclo de 2013

L - média das listas de exercício

n_l - número de listas feitas

n_t - número total de listas

Mon - média das monografias

n_{mo} - número de monografias entregues

n_{tm} - total de monografias

T_f - média do trabalho final (artigo)

n_m - número de milestones feitos

n_r - número total de milestones

$$Mf = (3 \cdot (n_l/n_t)L + 2 \cdot (n_{mo}/n_{tm})Mon + 5 \cdot (n_m/n_r)T_f) / 10$$

Suporte a disciplinas : e-disciplinas

edisciplinas.usp.br

e-Disciplinas Disciplinas » Suporte » Português - Brasil (pt_br)

Jose Reinaldo Silva

PMR5237 - Modelagem e Design de Sistemas Discretos em Redes de Petri (2019)

[Início](#) ▶ [Meus Ambientes](#) ▶ [2019](#) ▶ [EP](#) ▶ [PMR](#) ▶ [PMR5237--2019](#)

Ativar edição

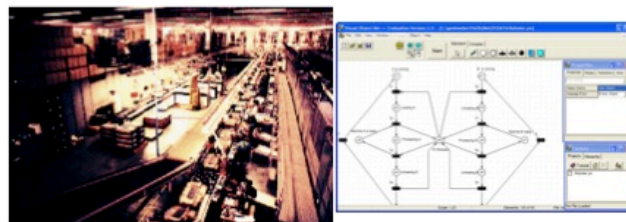
ADMINISTRAÇÃO

- ▼ Administração do ambiente
 - ⚙ Editar configurações
 - ✎ Ativar edição
 - ⚙ Conclusão de curso
 - ▶ Usuários
 - 🔍 Filtros
 - ▶ Relatórios
 - ⚙ Configuração das Notas
 - 📊 Resultado da aprendizagem
 - ▶ Emblemas
 - 📁 Backup
 - 📁 Restaurar
 - 📁 Importar
 - ▶ Banco de questões
 - 🗑 Lixeira

NAVEGAÇÃO

- [Início](#)
- 🏠 Painel
 - ▶ e-Disciplinas
- ▼ Meus Ambientes
 - ▼ 2019
 - ▼ EP
 - ▼ PMR
 - ▼ **PMR5237--2019**
 - ▶ Participantes

Introdução (Aula 1)



A figura mostra a antiga fábrica de impressoras RIMA cujo fluxo de material transportado por AGVs foi projetado e analisado pela Escola Politécnica em 1990.

Desde a sua criação em 1962 as redes de Petri só fizeram ampliar o seu escopo de aplicação. Paralelamente, o formalismo também foi ampliado e aperfeiçoado, e várias extensões surgiram para atender a domínios de aplicação localizados (embora muito importantes). Assim, a chegada do século XXI marca um período de grande profusão e até alguma confusão, sobre o formalismo base das redes de Petri, o que é também um requisito para a difusão das Redes de Petri, especialmente no meio industrial, que precisa de uma certa unidade para investir em produtos (CLPs, SDCDs, por exemplo) que possam fazer uso deste formalismo. Um comitê de pesquisadores passou a discutir a padronização formalismo base, culminando com a proposição de uma norma, premido pela grande difusão das RdP inclusive para a área de software, análise de requisitos, planejamento automático, verificação formal, model checking, etc., além do seu uso normal como esquema para a modelagem de sistemas discretos e simulação destes sistemas.

Nesta primeira aula veremos uma breve introdução, ainda intuitiva, sobre as redes de Petri, e especialmente sobre os conceitos da modelagem de sistemas discretos. Também serão abordados brevemente os paradigmas de modelagem e análise, sempre priorizando os sistemas dinâmicos discretos segundo o paradigma de modelagem estado/transição.

PESQUISAR NOS FÓRUMS

[Pesquisa Avançada ?](#)

ÚLTIMOS AVISOS

[Acrescentar um novo tópico...](#)
(Nenhum aviso publicado.)

PRÓXIMOS EVENTOS

Não há nenhum evento próximo
[Ir para o calendário...](#)

CALENDÁRIO

setembro 2019						
Seg	Ter	Qua	Qui	Sex	Sáb	Dom
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

PMR5237 - Modelagem e Design de Sistemas Discretos em Redes de Petri (2019)

Início » Meus Ambientes » 2019 » EP » PMR » PMR5237--2019 » Participantes

ADMINISTRAÇÃO

- Administração do ambiente
 - Editar configurações
 - Ativar edição
 - Conclusão de curso
- Usuários
 - Usuários inscritos
 - Métodos de inscrição
 - Grupos
 - Permissões
 - Outros usuários
- Filtros
 - Relatórios
 - Configuração das Notas
 - Resultado da aprendizagem
 - Emblemas
 - Backup
 - Restaurar
 - Importar
 - Banco de questões
 - Lixeira

Participantes

Não foram aplicados filtros

Pesquise por palavra-chave ou selecione um filtro

Inscriver usuários

Número de participantes: 3

Nome **Todos** A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 Sobrenome **Todos** A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Selecionar	Nome / Sobrenome	Endereço de email	Número USP	Papéis	Grupos	Último acesso ao curso	Estado
<input type="checkbox"/>	Jose Reinaldo Silva	reinaldo@usp.br	58759	Docente	T-PMR5237-3	13 segundos	Ativo
<input type="checkbox"/>	Oscar Stiven Morales Zapata	oscar.s.morales.z@usp.br	11310087	Estudante	T-PMR5237-3	4 horas 42 minutos	Ativo
<input type="checkbox"/>	Sonia Kaoru Shiba Justo	sonia.shiba@usp.br	5222528	Estudante	T-PMR5237-3	Nunca	Ativo

Selecionar tudo Desmarcar todas as seleções

Com usuários selecionados...

Escolher...

Inscriver usuários

PESQUISAR NOS FÓRUMS

Vai

[Pesquisa Avançada](#)

ÚLTIMOS AVISOS

[Acrescentar um novo tópico...](#)

(Nenhum aviso publicado.)

PRÓXIMOS EVENTOS

Não há nenhum evento próximo

[Ir para o calendário...](#)

CALENDÁRIO

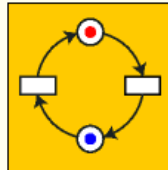
setembro 2019

Seg	Ter	Qua	Qui	Sex	Sáb	Dom
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Referência na Internet: Petri Nets World

www.informatik.uni-hamburg.de/TGI/PetriNets/Imprint/Disclaimer

[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.



PUBLICATIONS

- [Bibliographies](#) •
Search large publication databases on Petri Nets. In particular the *Petri Nets Bibliography* contains thousands of entries.
- [Periodicals](#) •
Information on the *Petri Net Newsletter*. Also contains a listing of similar journals, newsletters, and other regularly published material.

Recent News in the Petri Nets World

- [September 5, 2019](#): Realizing the promises of Predictive Maintenance: an exciting PhD position at University of Twente, the Netherlands
- [September 5, 2019](#): PhD on Trusted Decentralised Applications
- [September 4, 2019](#): Open Postdoc Position at CiTIUS - Research Centre in Intelligent Technologies, Santiago de Compostela, Spain
- [September 4, 2019](#): Open position Project Manager for H2020-MSCA-ITN-2019 "Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI)"
- [September 3, 2019](#): 2nd round Call-for-Nominations: SPEC Kaivalya Dixit Distinguished Dissertation Award 2019
- [September 3, 2019](#): FM'19 - 2nd Call for Participation
- [September 2, 2019](#): PADL 2020 Call for Papers
- [August 31, 2019](#): ETAPS 2020 2nd joint call for papers
- [August 29, 2019](#): A PhD position (Probabilistic Model Checking) in the University of South Florida
- [August 29, 2019](#): FMTea19 CALL for PARTICIPATION: Formal Methods Teaching, on October 7, 2019, in Porto, Portugal
- [August 27, 2019](#): TPTP v7.3.0 released
- [August 27, 2019](#): Tutorial on Formal Methods for Biomedicine @ FM World Congress, Porto, October 8, 2019
- [August 27, 2019](#): [Deadline Approaching, August 30] FPS'19, Foundations & Practice of Security, Springer-LNCS Proceedings, Toulouse, France
- [August 24, 2019](#): Second call for papers Software Verification and Testing (SVT) at SAC 2020
- [August 16, 2019](#): Cfp PhD Symposium at iFM'19 on Formal Methods: Algorithms, Tools and Applications
- [August 16, 2019](#): PhD position in Utah State University (Probabilistic Verification Methodology for Synthetic Biology and Nanotechnology)
- [August 11, 2019](#): CFP: EPEW 2019 - The 16th European Performance Engineering Workshop

Ferramentas de software

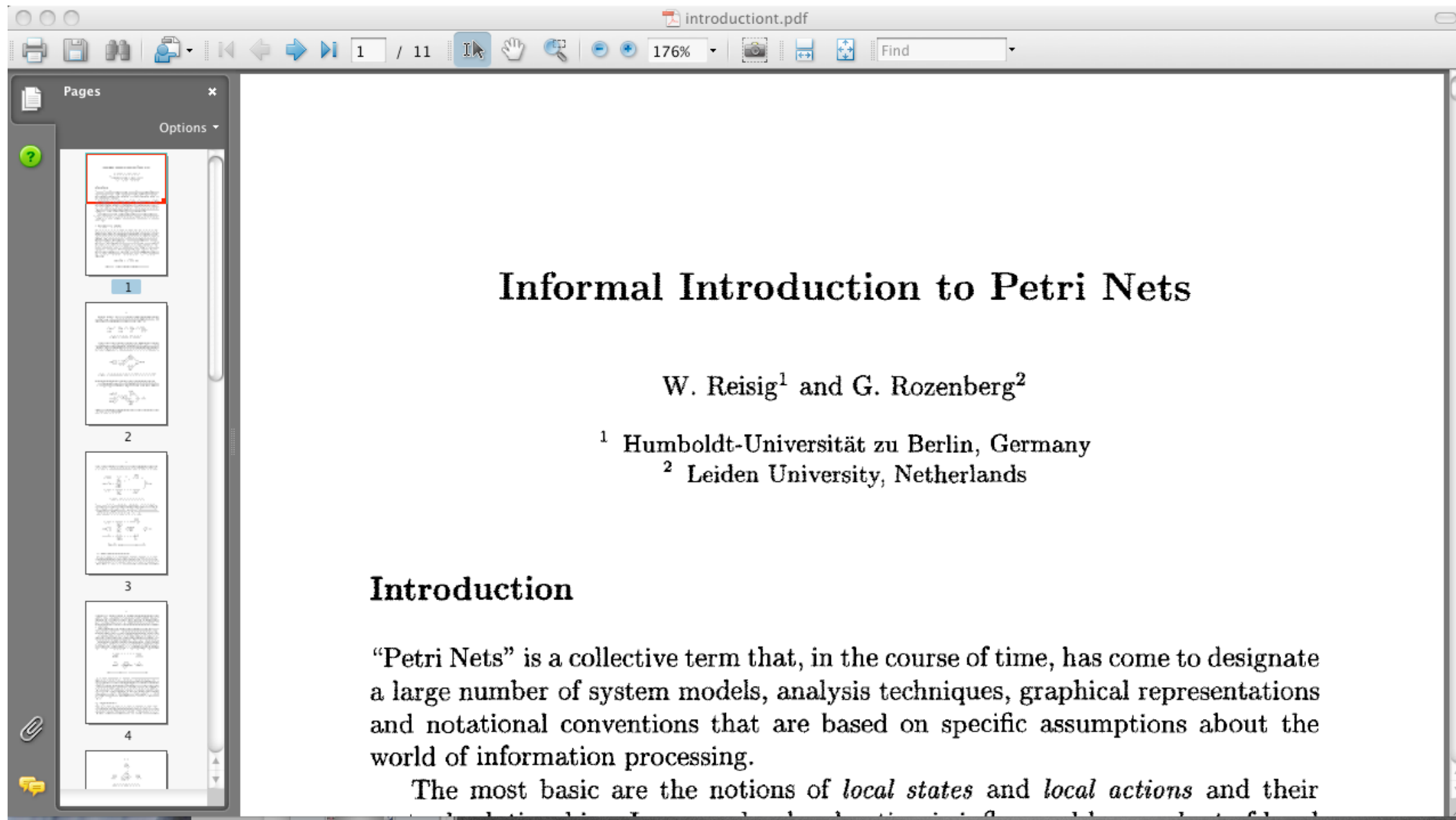
PIPE (Windows, Linux, Mac)

~~HPSIM (Windows)~~ Renew

CPN Tools

ePNK, GHENeSys (?)

Leitura da semana



The screenshot shows a PDF viewer window titled 'introductiont.pdf'. The main content area displays the title page of a document. The title is 'Informal Introduction to Petri Nets'. Below the title are the authors 'W. Reisig¹ and G. Rozenberg²'. The affiliations are listed as '¹ Humboldt-Universität zu Berlin, Germany' and '² 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 currently selected. The top toolbar includes navigation and search icons, and the status bar shows '1 / 11' pages and a zoom level of '176%'.

<https://medium.com/basecs/a-gentle-introduction-to-graph-theory-77969829ead8>

base_{cs}

Follow

Sign in

Get started

HOME FUNDAMENTALS DATA STRUCTURES ALGORITHMS THEORY IN PRACTICE 



Vaidehi Joshi

Follow

Writing words, writing code. Sometimes doing both at once.
Mar 20, 2017 · 11 min read

A Gentle Introduction To Graph Theory

So many things in the world would have never come into existence if there hadn't been a problem that needed solving. This truth applies to everything, but *boy, is it obvious* in the world of computer science.

Someone needed a way of keeping track of the order of things, so they played around with and created different data structures until they found the one that worked the best for the specific problem that they were trying to solve. Someone else needed a good way of storing data, so they played around with different number systems until they found one that worked best for the kind of information that they wanted to contain. People needed a good way of labeling and processing tasks, so they found a way to build upon the tools they had and created a way to juggle all the things that one single system needed to do, at any given time.

Of course, computer science isn't the *only* field to innovate and build upon



Never miss a story from **basecs**, when you sign up for Medium.

[Learn more](#)

[GET UPDATES](#)

Para a próxima aula:

1. Refazer o exercício da largada de fórmula I usando o PIPE2: Acesse o Petri Nets World e baixe o sistema PIPE2 para o seu sistema operacional.
2. Leia o artigo colocado no link “leitura da semana”, trata-se de uma introdução informal às redes de Petri, escrito por dois dos nome mais proeminentes da área no mundo;
3. Comece a ler o artigo do Manuel Silva para um histórico interessante sobre as redes de Petri

Próxima aula 17 de setembro

Fim