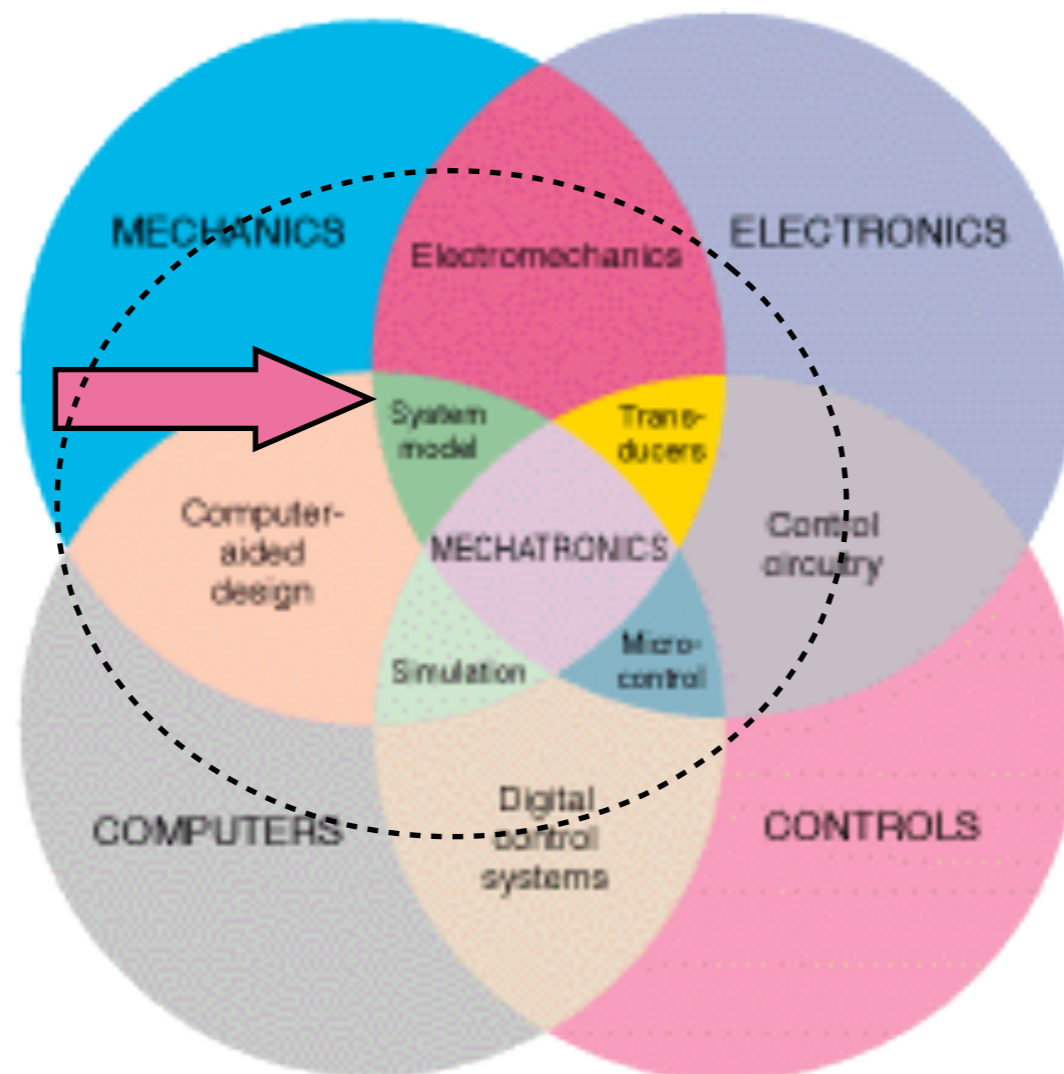


PMR 5237 Modelagem e Design de Sistemas Discretos em Redes de Petri

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 discretizavel

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

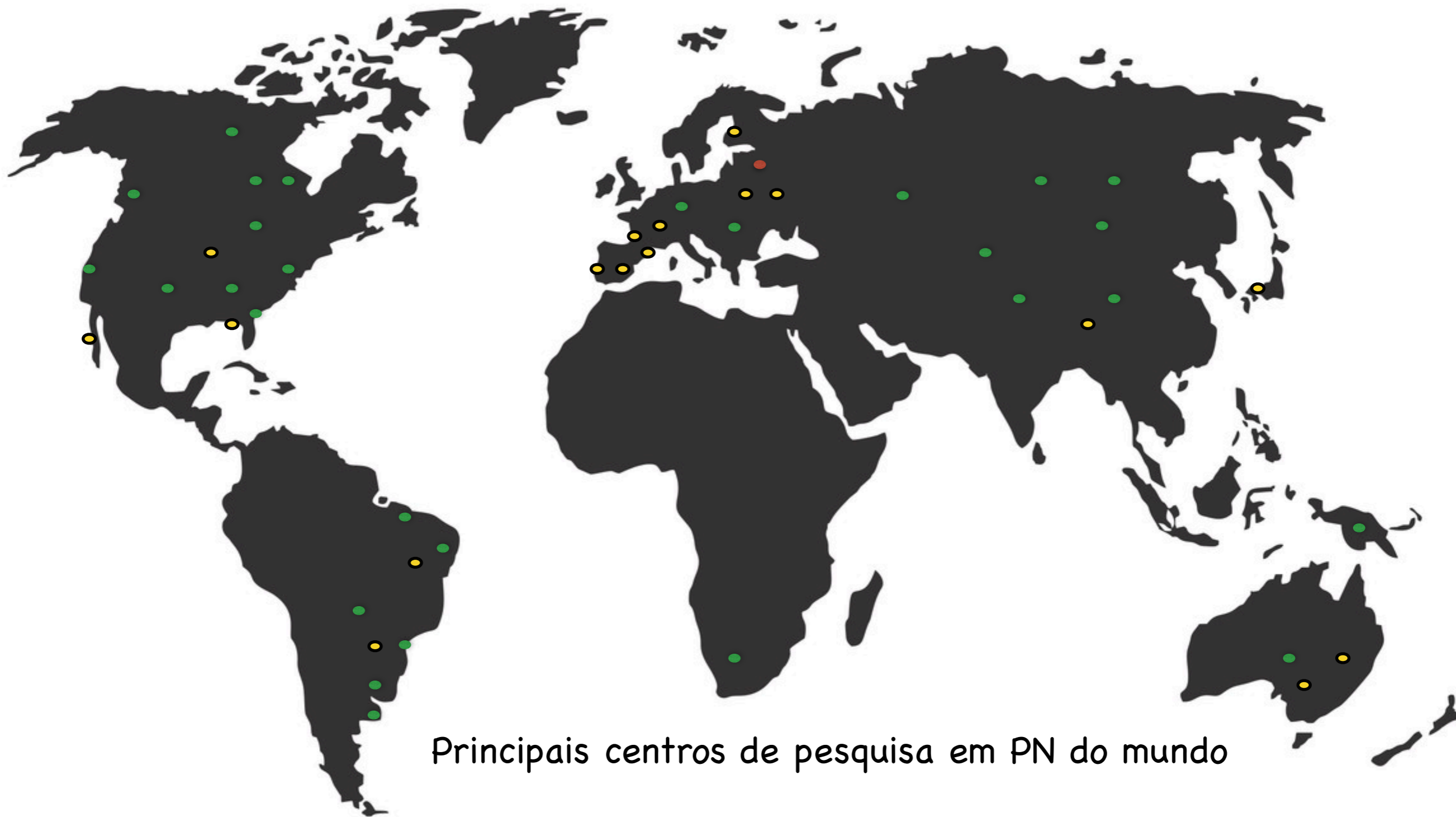


Schemas

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

TADÃO MURATA (1938-2021)

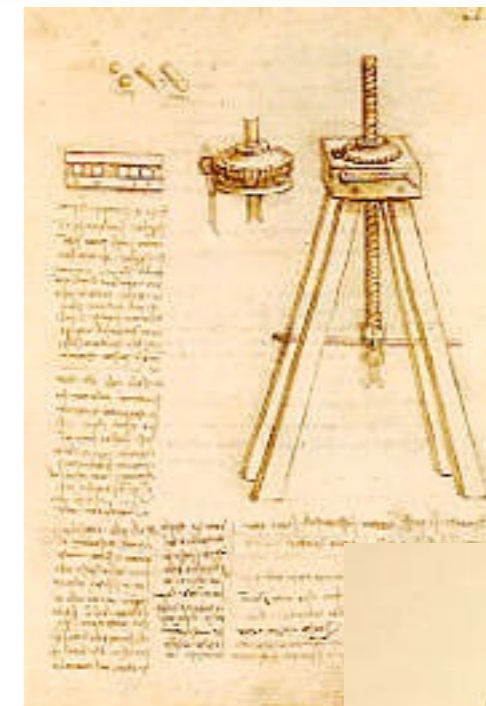




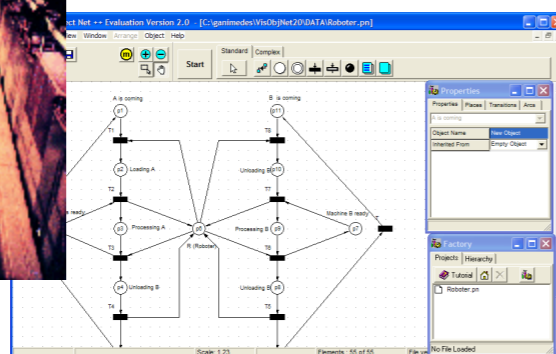
Principais centros de pesquisa em PN do mundo

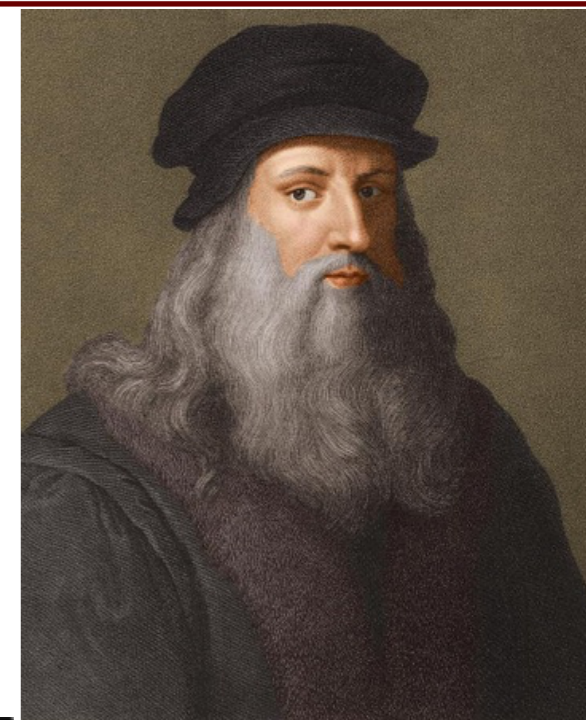
A “arte” de modelar

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.

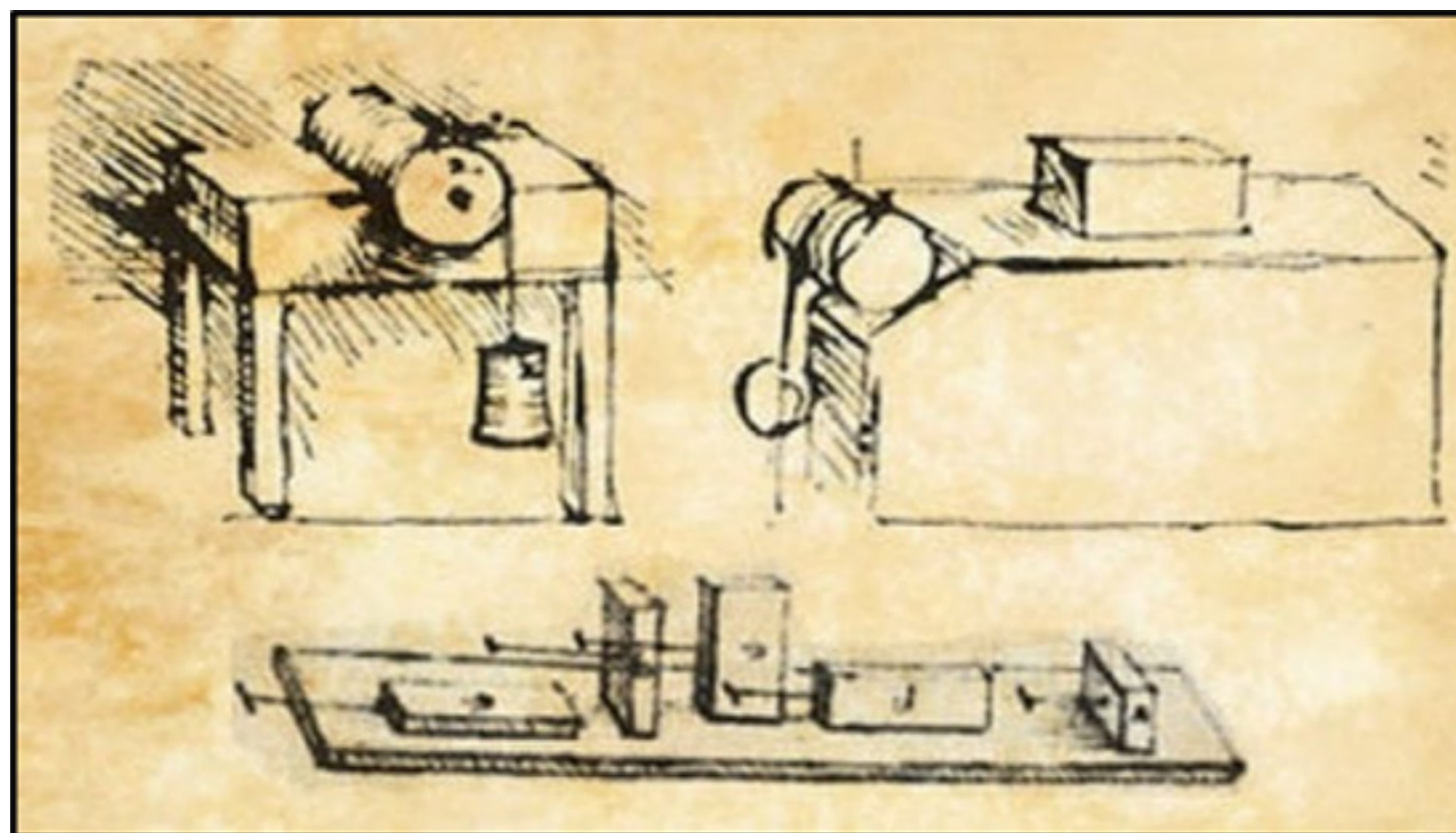


Modelo de Leonardo da Vinci

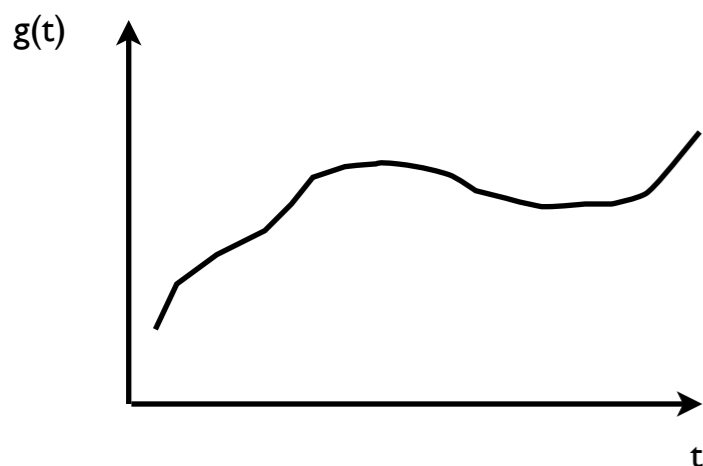




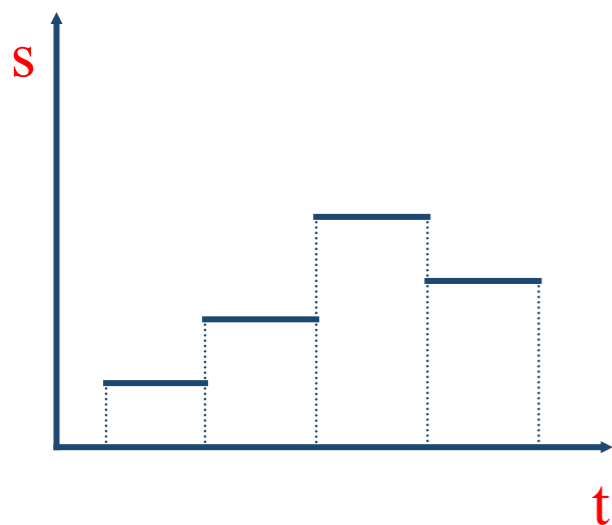
Por volta de 1470 (~1460-1480) Leonardo da Vinci e dois outros “gravadores de metal” noruegueses, Albrech Durer e Martin Schongauer introduziram na sua prática esboços com a descrição prévia do que iriam fazer.



Paradigmas para modelagem: Estado/Transição



Genericamente, a descrição do comportamento de sistemas é feita identificando uma função que descreve a evolução das “variáveis de estado” que caracterizam o sistema. Esta “função de estado” pode ter pontos de equilíbrio que caracterizam o estado e evoluem no tempo denotando o comportamento do sistema. A dinâmica ou mudança de estados é caracterizada pela sucessão dos “eventos”.

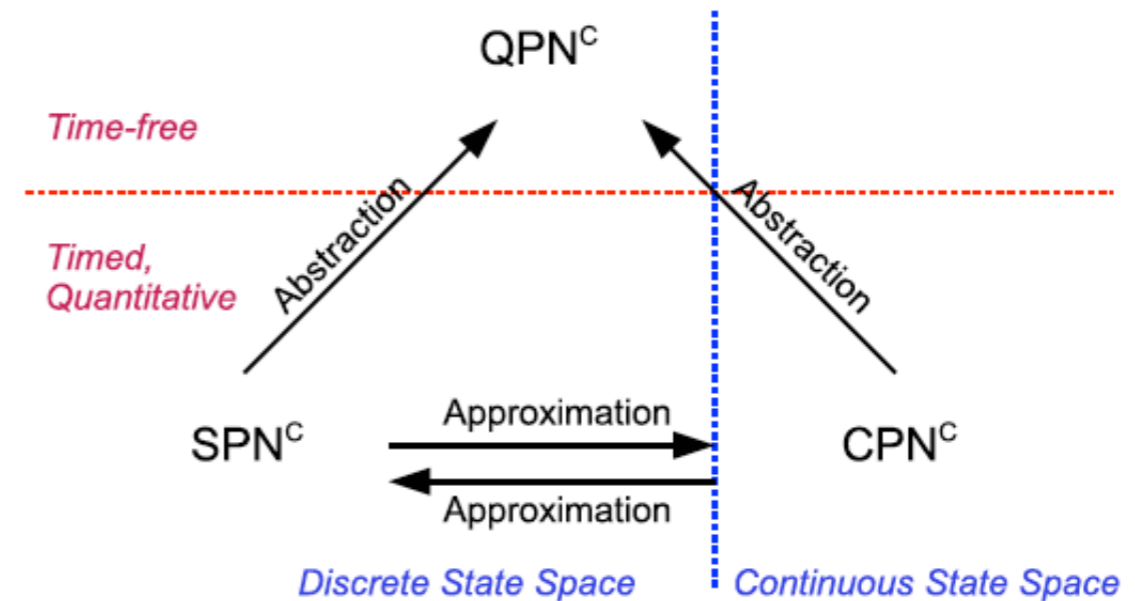


Em alguns casos a *função de estado* é contínua nas variáveis de estado e também no tempo caracterizando trajetórias também contínuas. Em outros casos a função de estado é discreta e evolui aos saltos caracterizando um sistema discreto.

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 42th Int. Conf. on Application and Theory of Petri Nets and Concurrency (realizada em conjunto com a 21th 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 2021.

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

PETRI NETS 2021 VIRTUAL (24TH JUNE 2021)

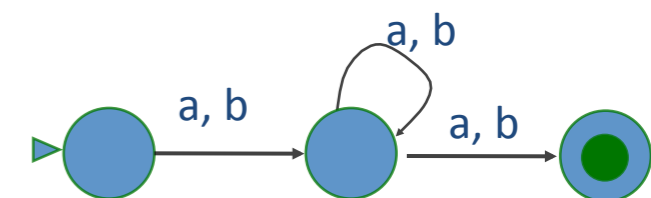


<https://lipn.univ-paris13.fr/petrinets2021/>

Formalização do processo de modelagem

A formalização da modelagem de qualquer sistema dinâmico (ou de qualquer processo discreto, seja qual for a aplicação) pode ser feito em Redes de Petri.

Para sistemas discretos, um formalismo alternativo genérico pode ser usado: a representação usando 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

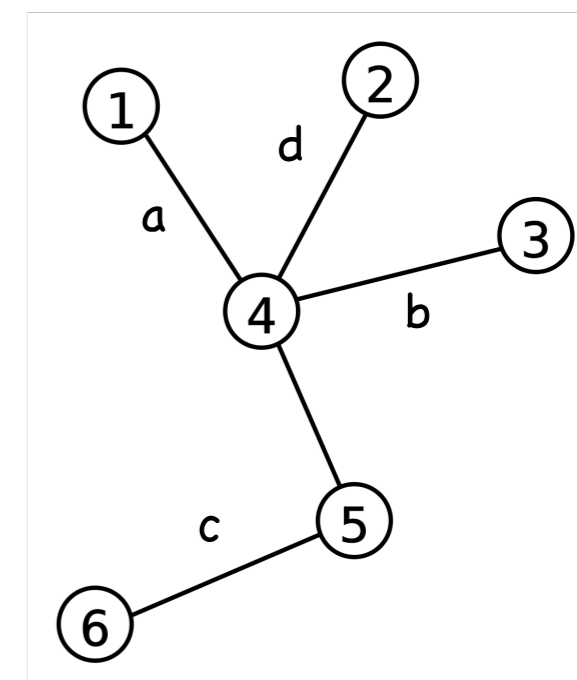
- ◆ Finite automata are finite collections of states with transition rules that take you from one state to another.
- ◆ Original application was sequential switching circuits, where the “state” was the settings of internal bits.
- ◆ Today, several kinds of software can be modeled by FA.

CSI 54 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

Representing FA

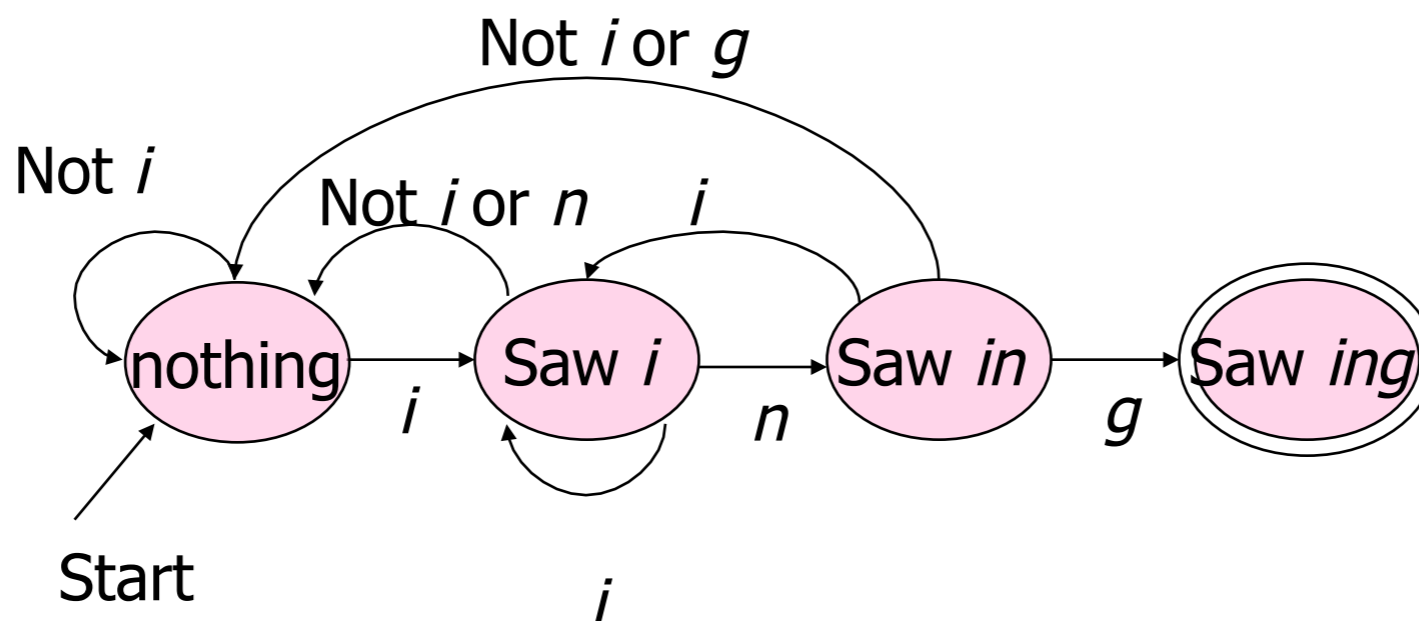
- ◆ Simplest representation is often a graph.
 - ▶ Nodes = states.
 - ▶ Arcs indicate state transitions.
 - ▶ Labels on arcs tell what causes the transition.



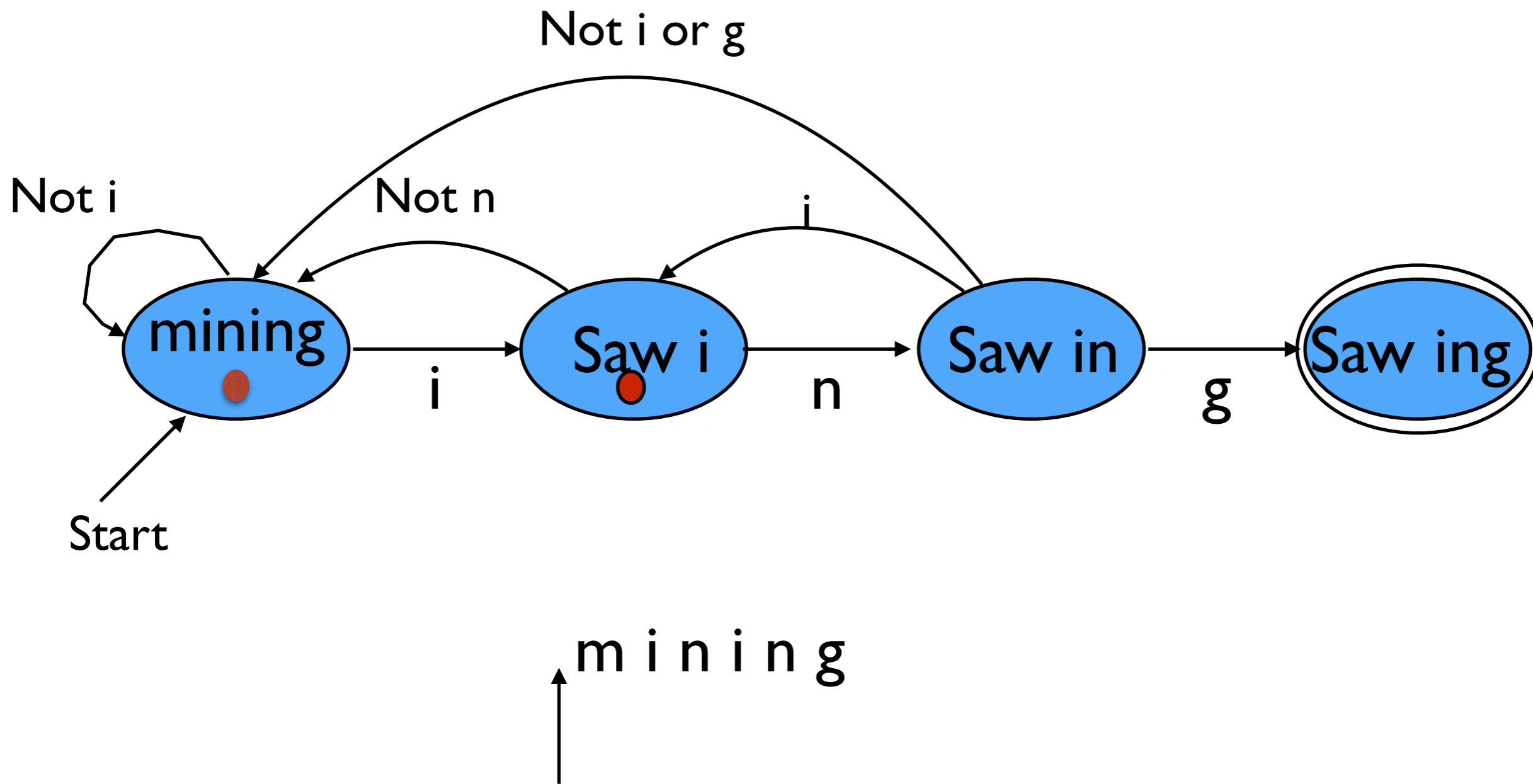
CSI 54 Introduction to Automata and Complexity Theory

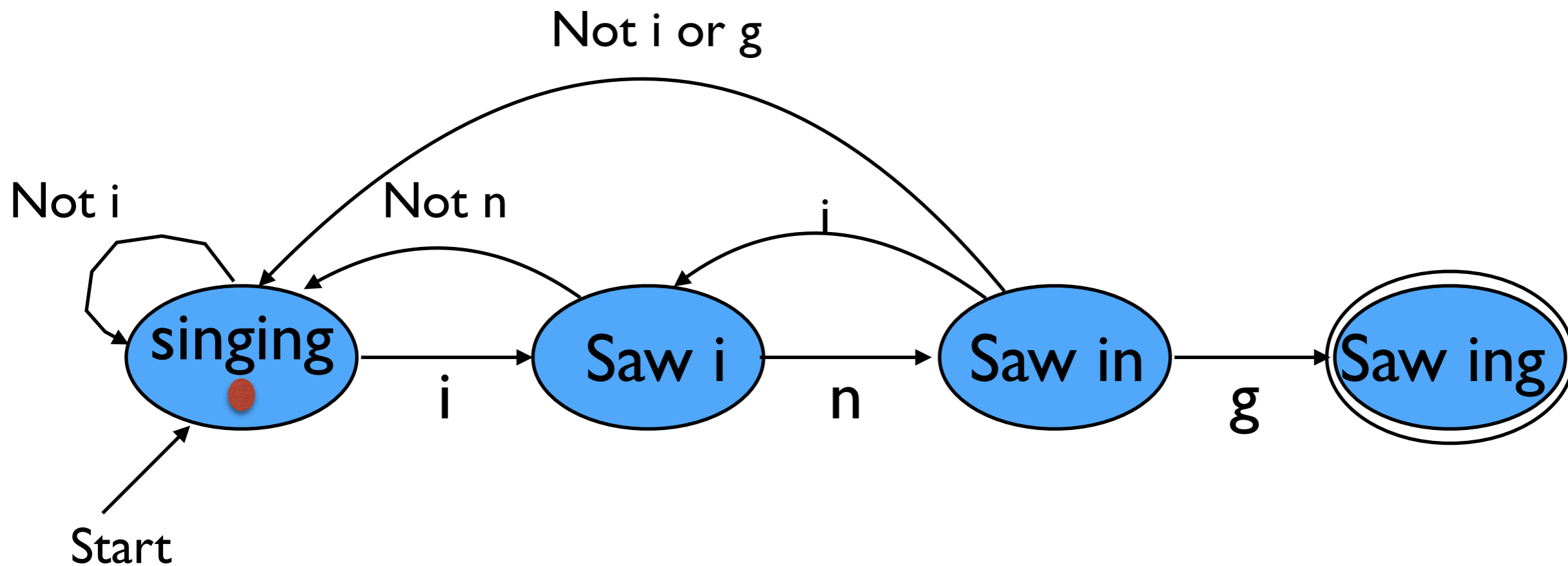
Jeffrey D. Ullman, Stanford University

Example: Recognizing Strings Ending in "ing"



Experimente agora fazer o parsing da palavra "mining".





S i n g i n g
 O que acontece se fizermos o parsing da palavra "singing"?

CSI 54 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

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

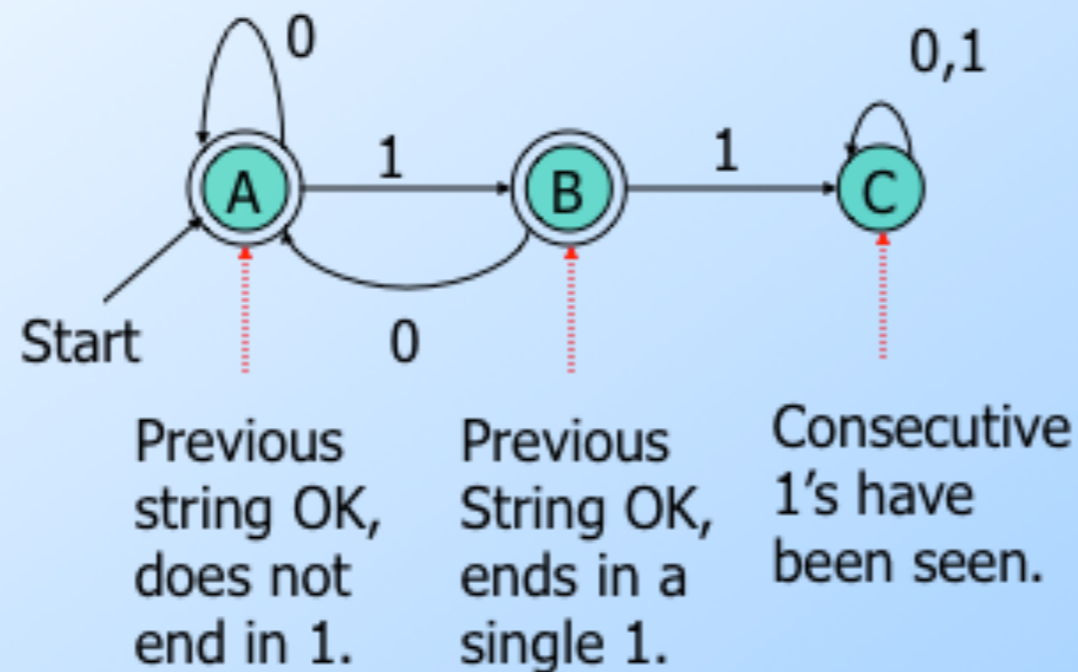
CSI 54 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

Também chamado Transition Graph

Example: Graph of a DFA

Accepts all strings without two consecutive 1's.

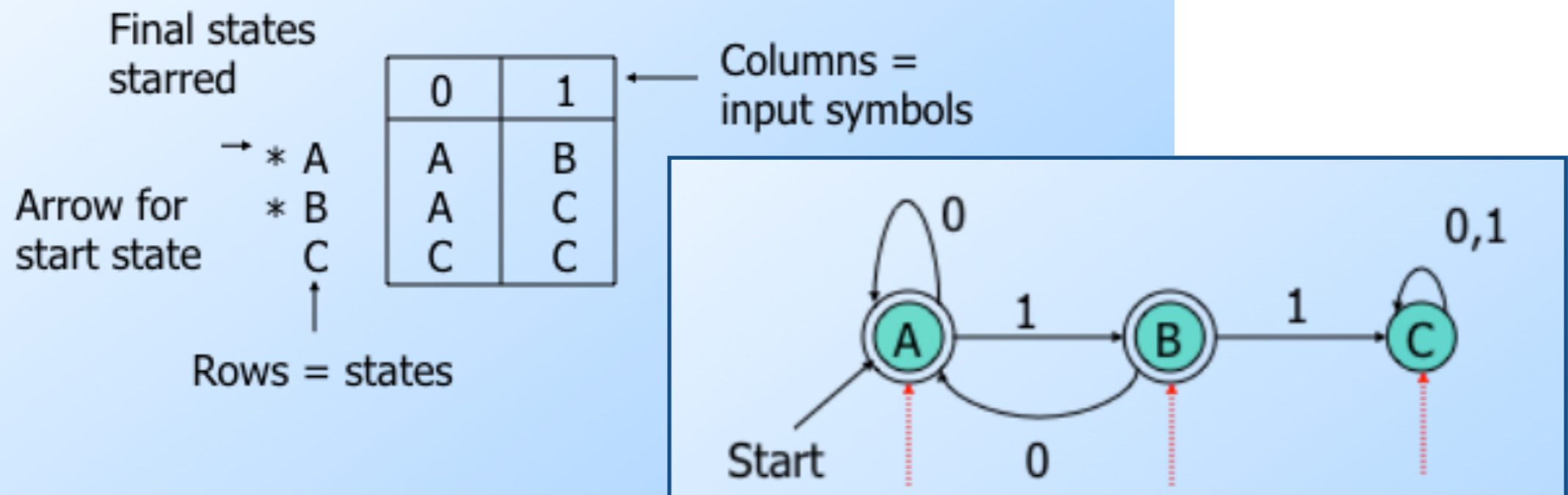


9

CSI 54 Introduction to Automata and Complexity Theory

Jeffrey D. Ullman, Stanford University

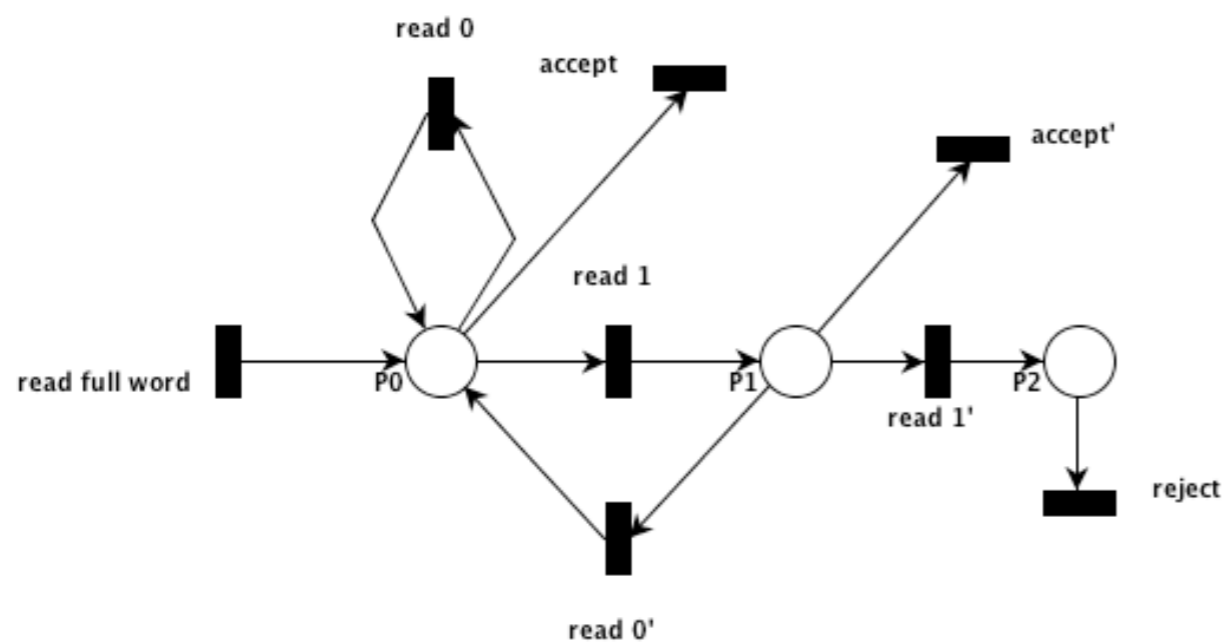
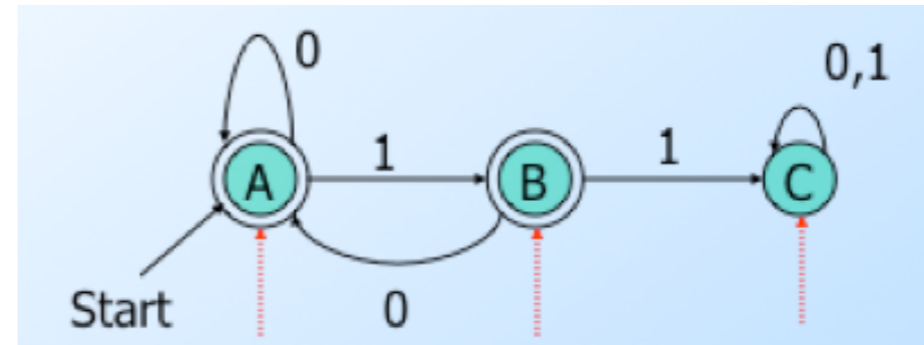
Alternative Representation: Transition Table



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

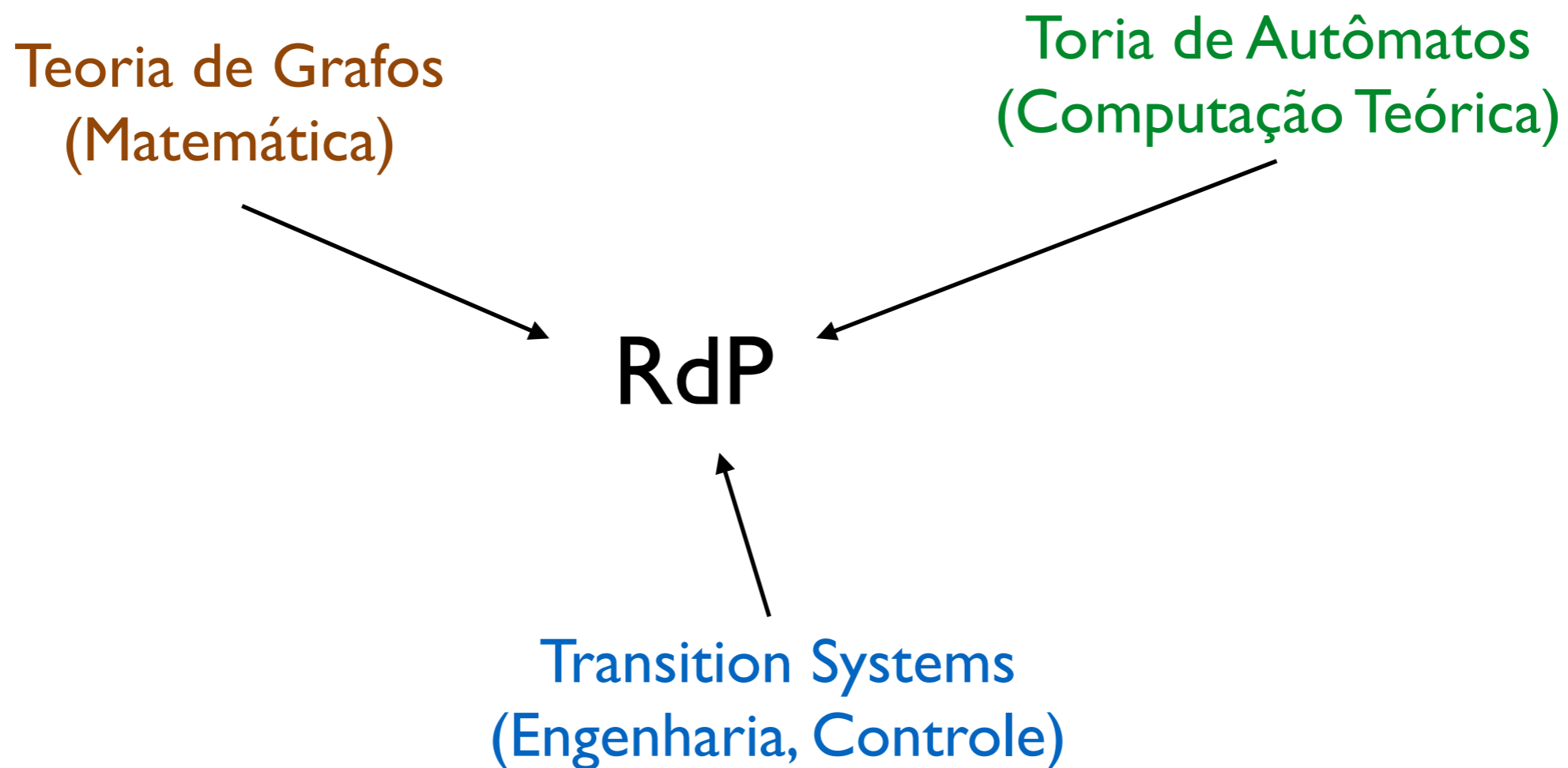
10

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?

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

- $S \cap T = \phi$.

Estados e transições são ambos elementos distribuídos:

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

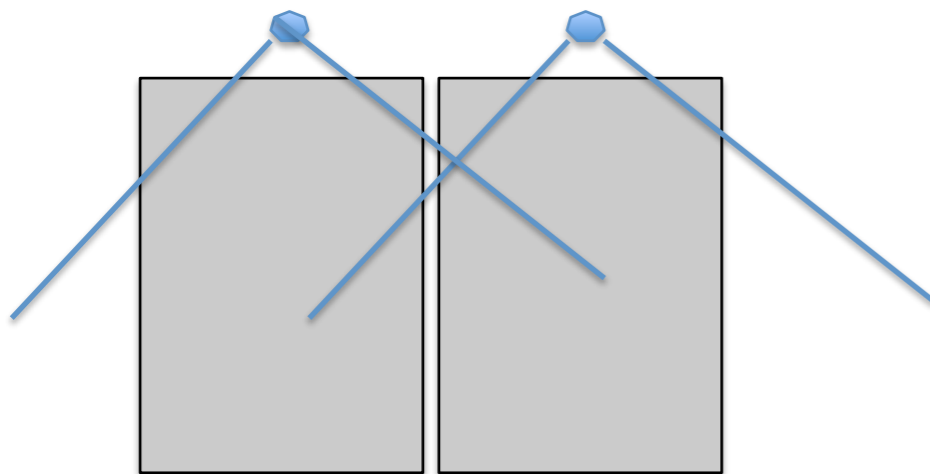
- $\{s_1, s_2, \dots, s_n\} \implies \textit{case}.$

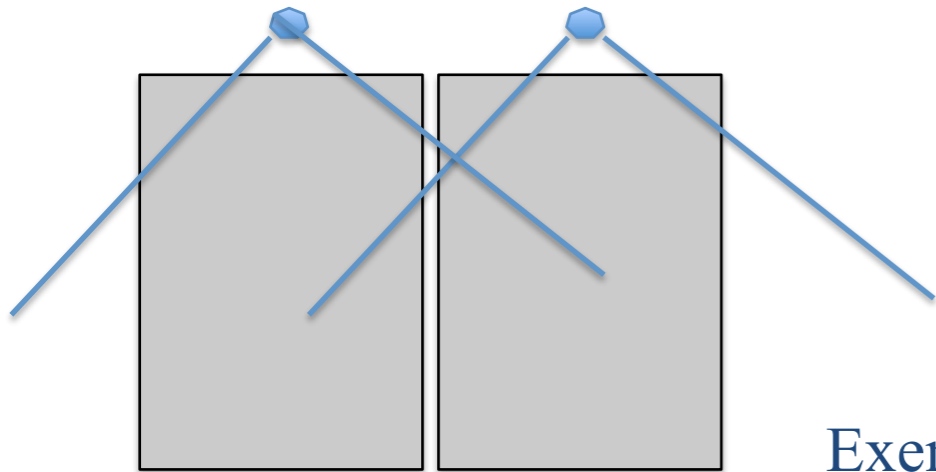
Uma transição distribuída é dada por um conjunto de transições válidas simultaneamente, isto é,

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

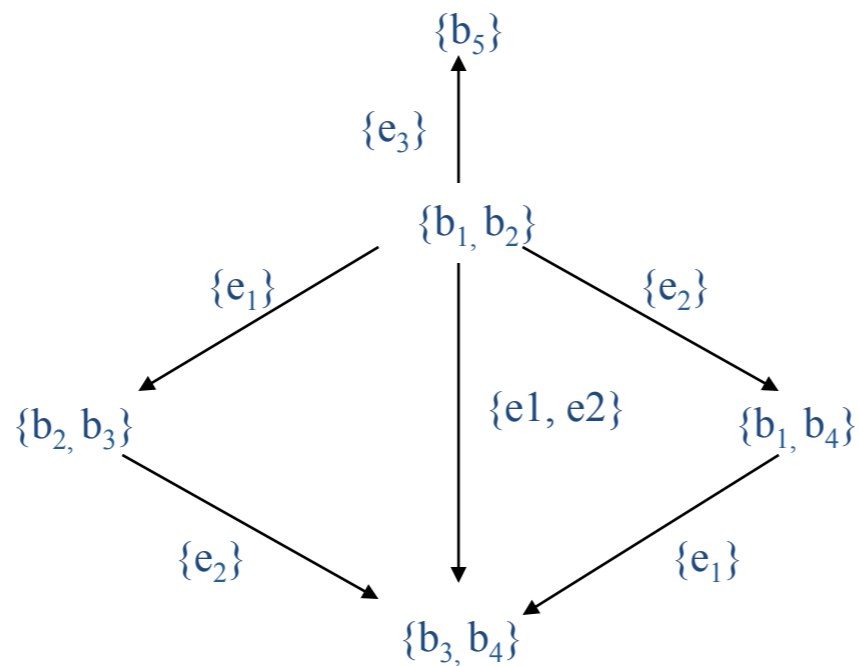
Um exemplo simples, aplicado a sistemas de automação, é a representação do comportamento de uma porta automática, que aciona motores para abertura e fechamento dependendo de sinal de sensores que detectam a aproximação de usuários.

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

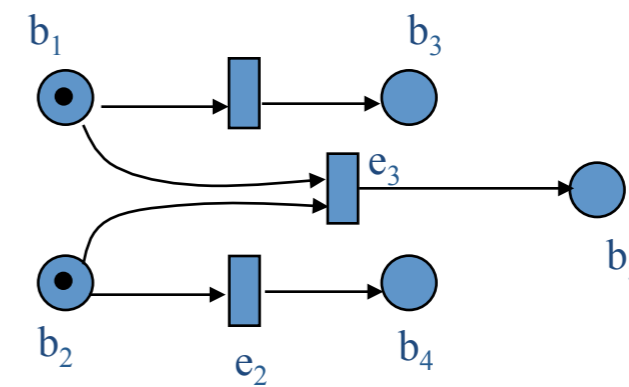




Exemplo (Thiagarajan) :



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 :

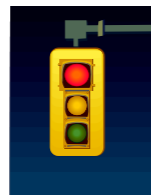
- o princípio da dualidade
- o princípio da localidade
- o princípio da concorrência
- o princípio da representação gráfica
- o princípio da representação algébrica

O princípio da Dualidade

Existem dois tipos de elementos em uma RdP: os elementos convencionalmente "passivos" ou P-elementos, que representam os estados, e os elementos "ativos" ou T-elementos, que representam as transições ou mudanças de estado.

Estes conjuntos são disjuntos. Não existe uma interseção (ou elemento) que possa pertencer às duas classes de elemento, ou, $P \cap T = \phi$.

Exemplo: a largada na fórmula 1



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

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

t_1

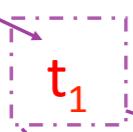
$M_2 = [p_1=F, p_2=V, p_3=F, p_4=T, p_5=F, p_6=T, p_7=F, p_8=F, p_9=F, p_{10}=T, p_{11}=F, p_{12}=F]$

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;

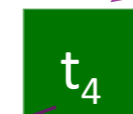
t_1 = carro A: envia sinal de prontidão
 t_2 = carro A: acelera
 t_3 = operador: manda sinal de largada
 t_4 = carro B: envia sinal de prontidão
 t_5 = carro B: acelera

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;

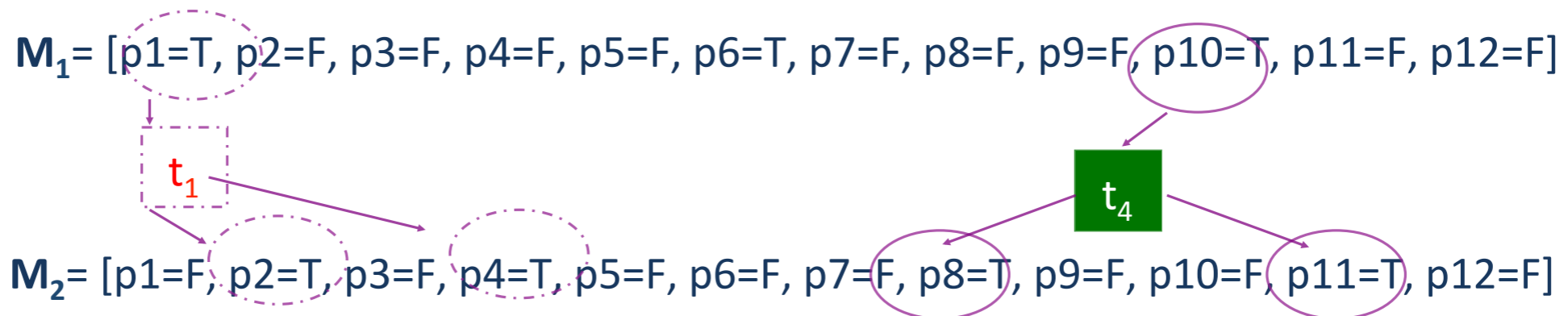
- t_1 = carro A: envia sinal de prontidão
- t_2 = carro A: acelera
- t_3 = operador: manda sinal de largada
- t_4 = carro B: envia sinal de prontidão
- t_5 = carro B: acelera

O princípio da localidade

O princípio da localidade é associado às transições

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

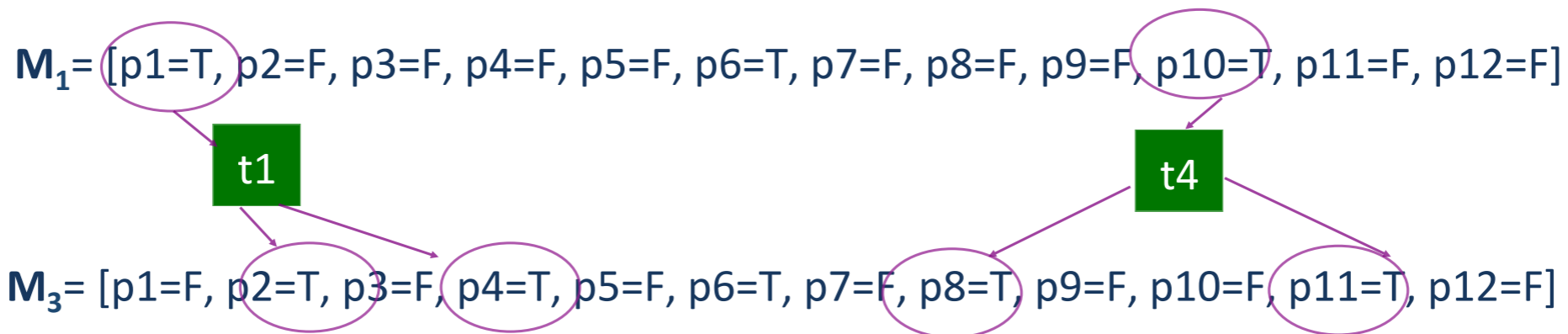
Pelo princípio da localidade o comportamento das transições depende unicamente da sua localidade.



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

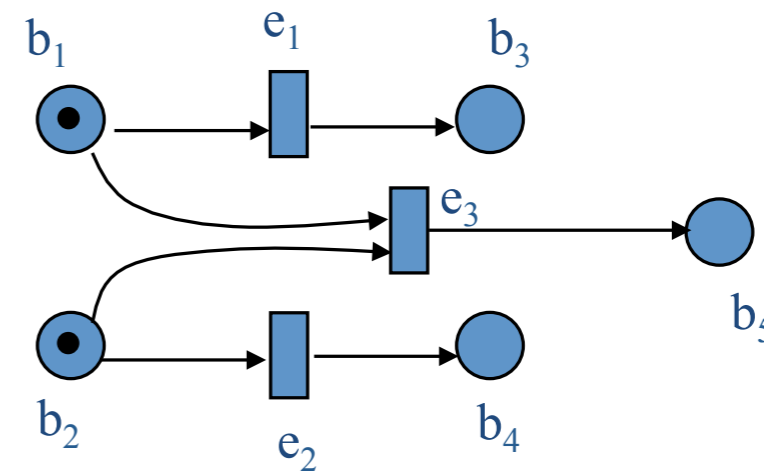


- Estados e transições são classes distintas que devem ser intercaladas;
- Estados e transições são entidades distribuídas;
- A extensão das mudanças causadas por uma transição se limitam a sua própria localidade;
- Uma transição é dita "habilitada" em um dado estado sse as mudanças a ela associadas puderem ocorrer neste estado, na extensão da sua localidade.

Exemplo:

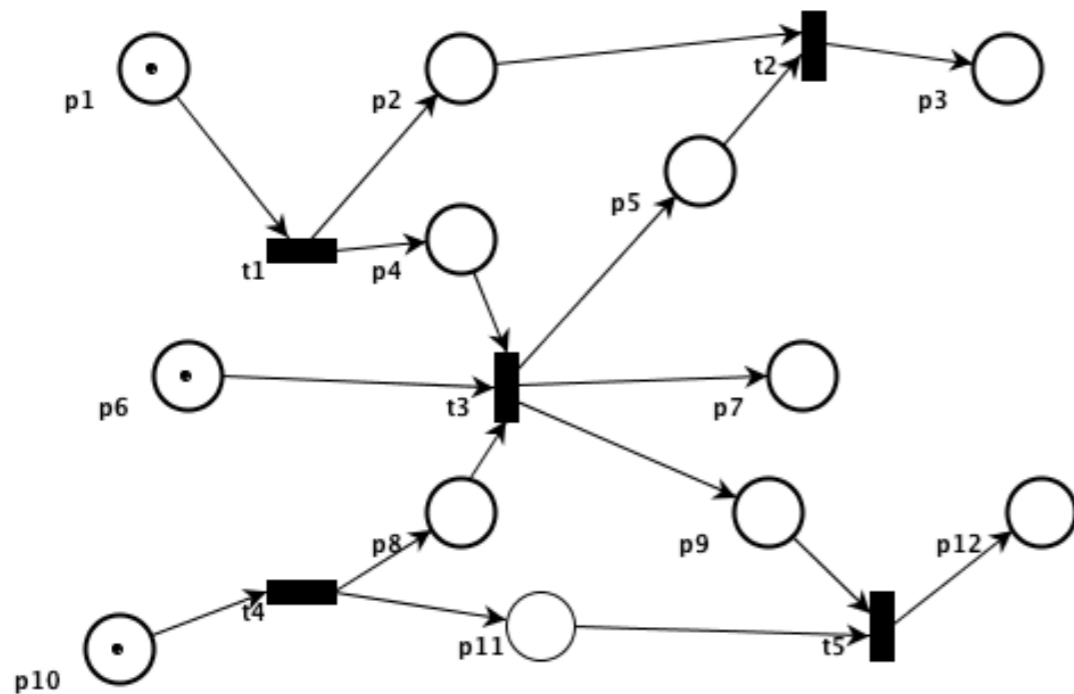
Usando o mesmo exemplo da porta automática podemos construir a representação algébrica (matriz de incidência) desta rede.

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

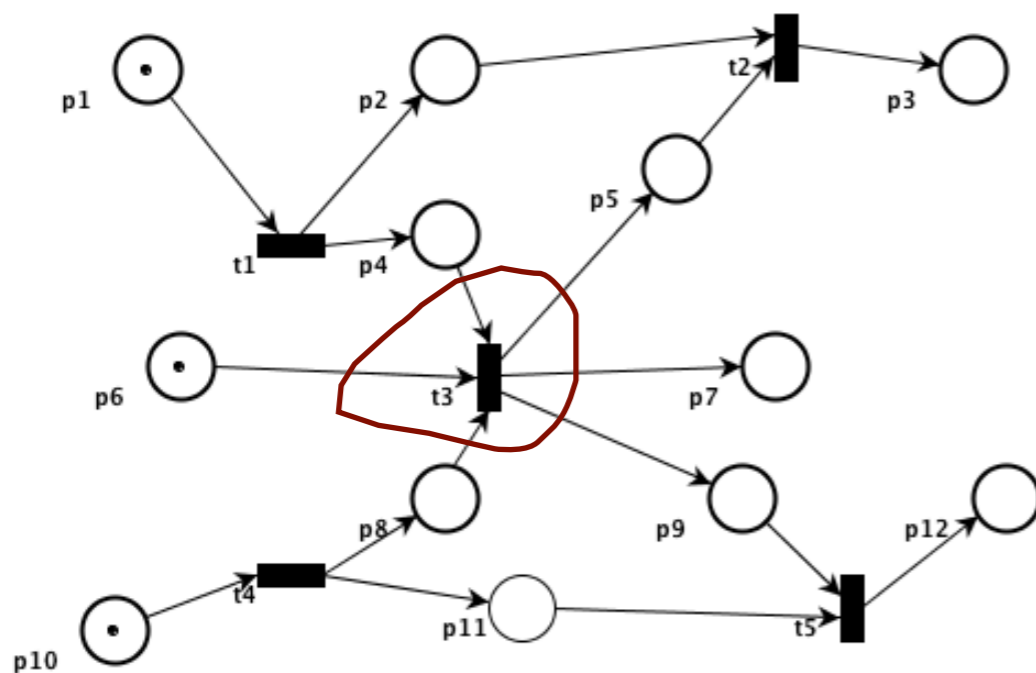
Use current net Filename:

Results

Combined incidence matrix I

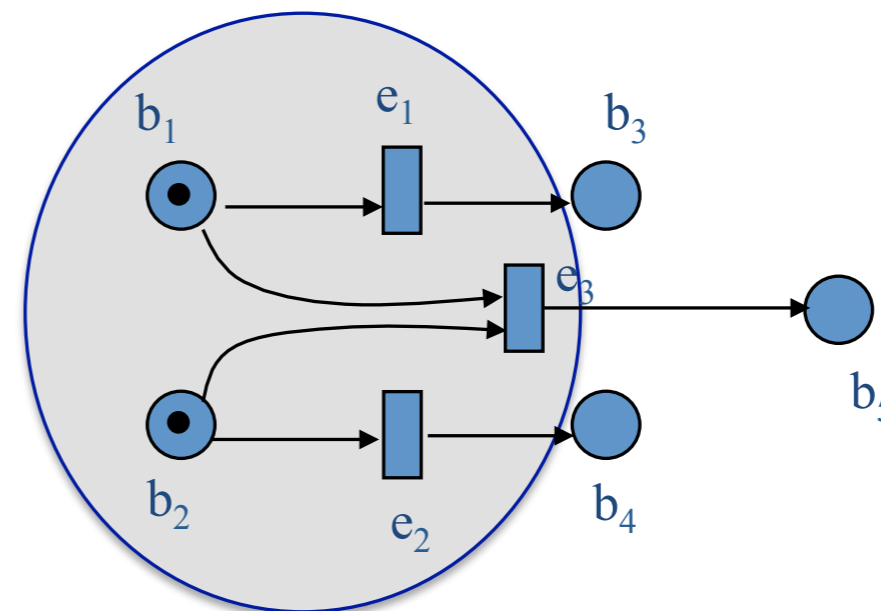
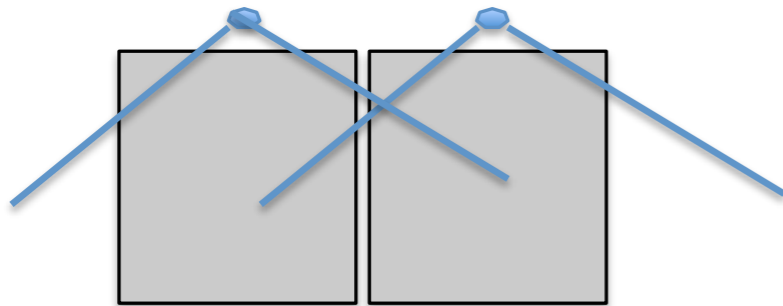
| | t1 | t2 | t3 | t4 | t5 |
|-----|----|----|----|----|----|
| p1 | -1 | 0 | 0 | 0 | 0 |
| p2 | 1 | -1 | 0 | 0 | 0 |
| p6 | 0 | 0 | -1 | 0 | 0 |
| p7 | 0 | 0 | 1 | 0 | 0 |
| p4 | 1 | 0 | -1 | 0 | 0 |
| p3 | 0 | 1 | 0 | 0 | 0 |
| p5 | 0 | -1 | 1 | 0 | 0 |
| p10 | 0 | 0 | 0 | -1 | 0 |
| p11 | 0 | 0 | 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.

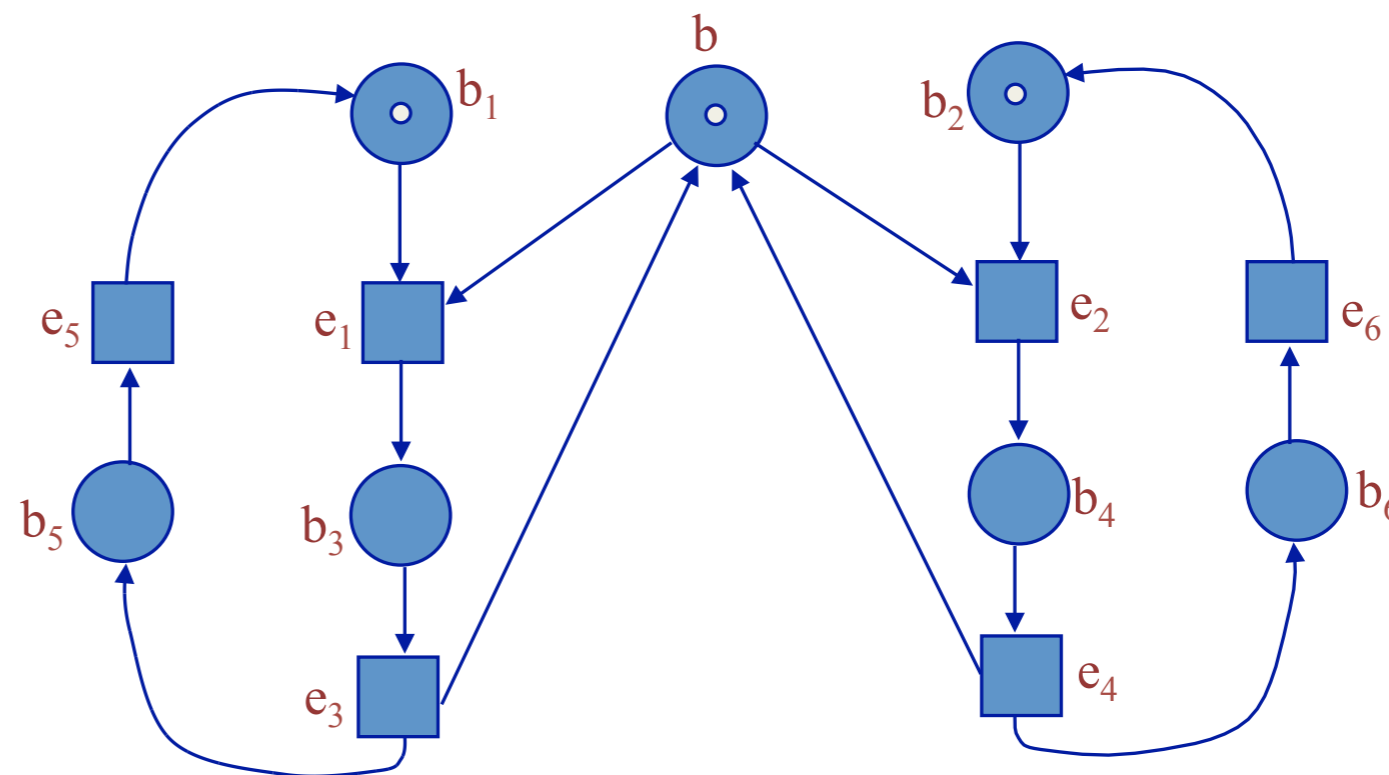
Configurações especiais: conflito



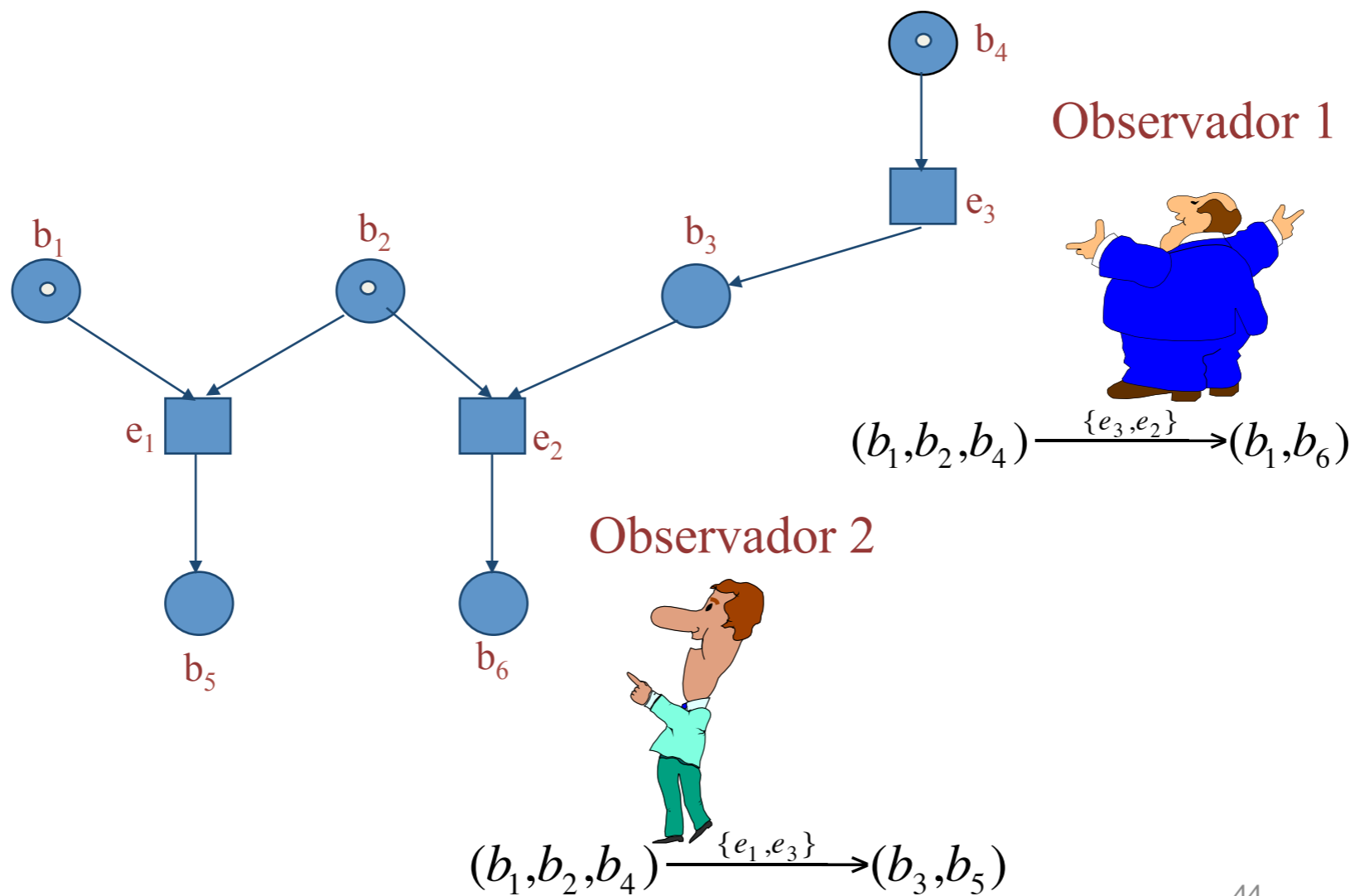
O Conflito denota uma situação de exclusão mútua (mutex), onde a ocorrência de uma transição inibe outras. No caso do exemplo as transições $\{e_1, e_2\}$ são mutuamente exclusivas à transição $\{e_3\}$.

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.

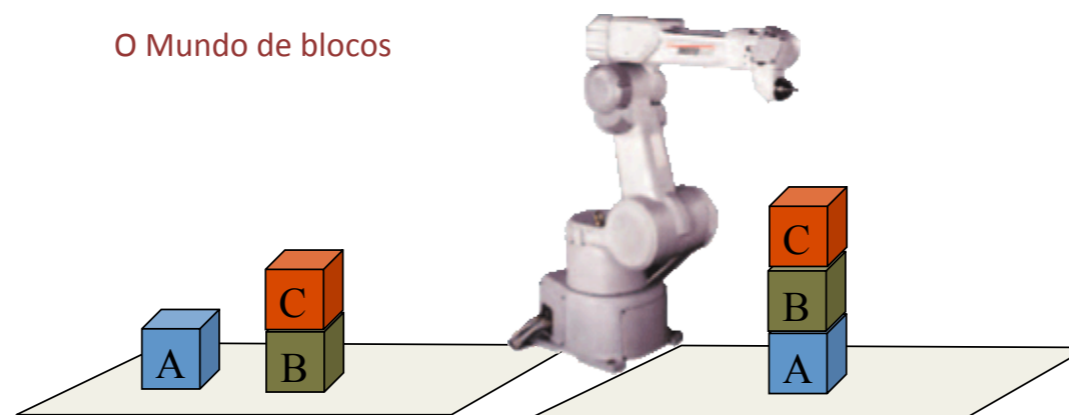


Configurações especiais: contato ou confusão



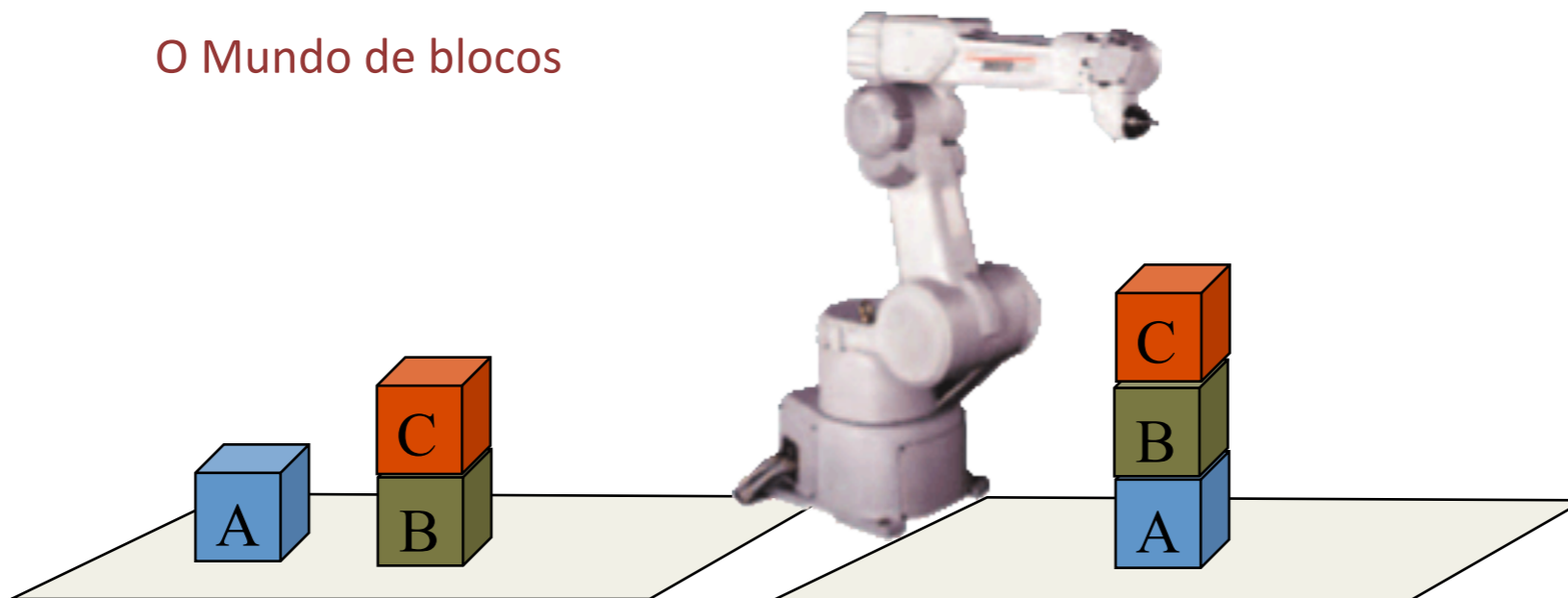
IA Planning: o STRIPS

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.

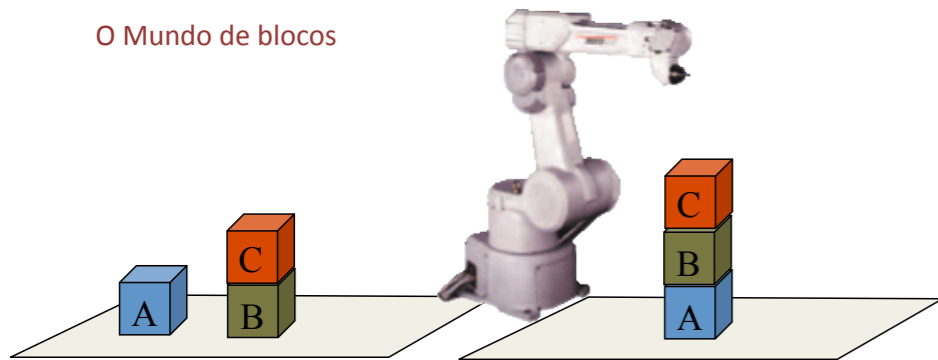


O exemplo mais simples e intuitivo do sistema STRIPS é o chamado “mundo de blocos” que consiste em mudar a configuração de um conjunto de blocos usando robôs.

O Mundo de blocos

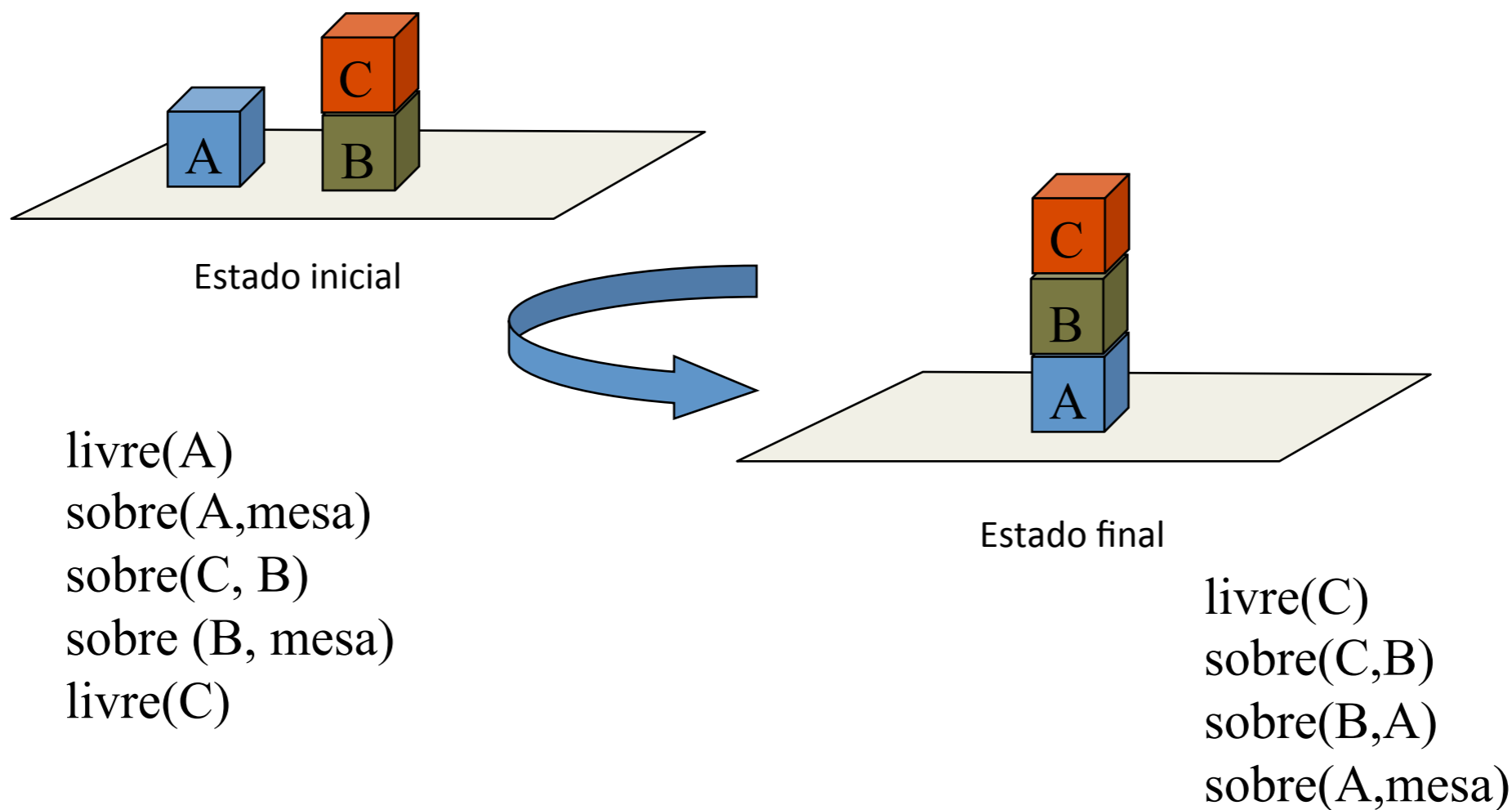


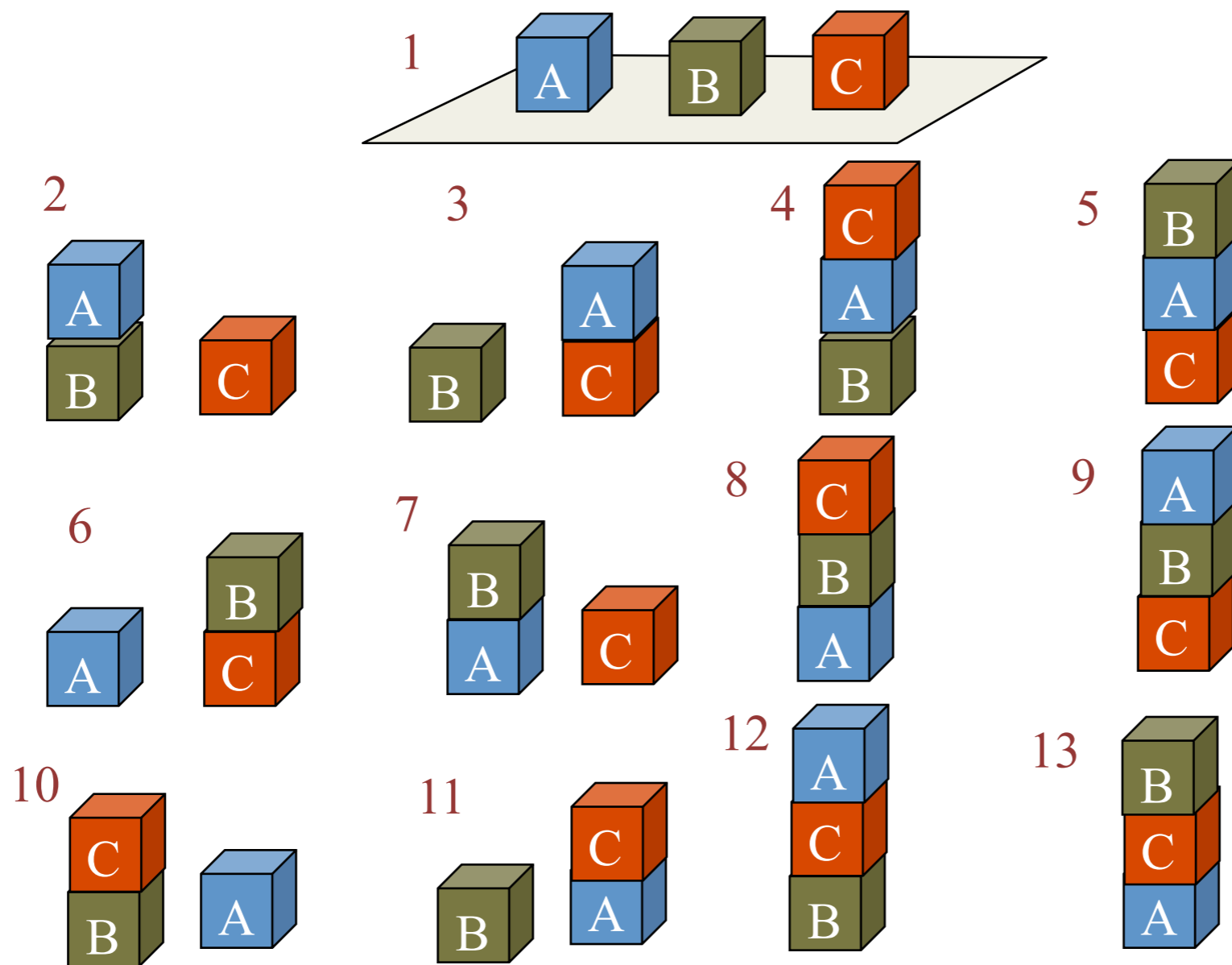
O Mundo de blocos

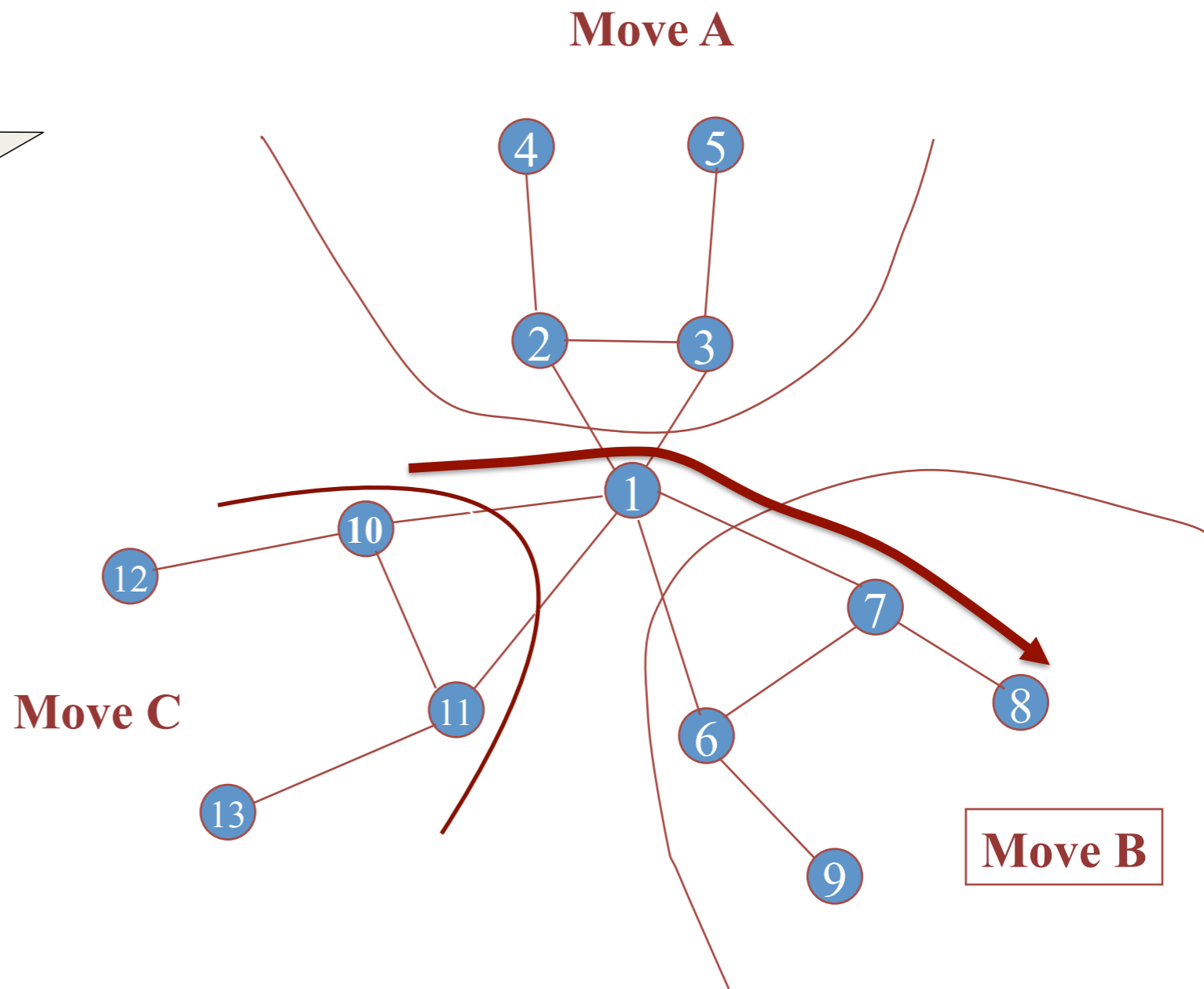
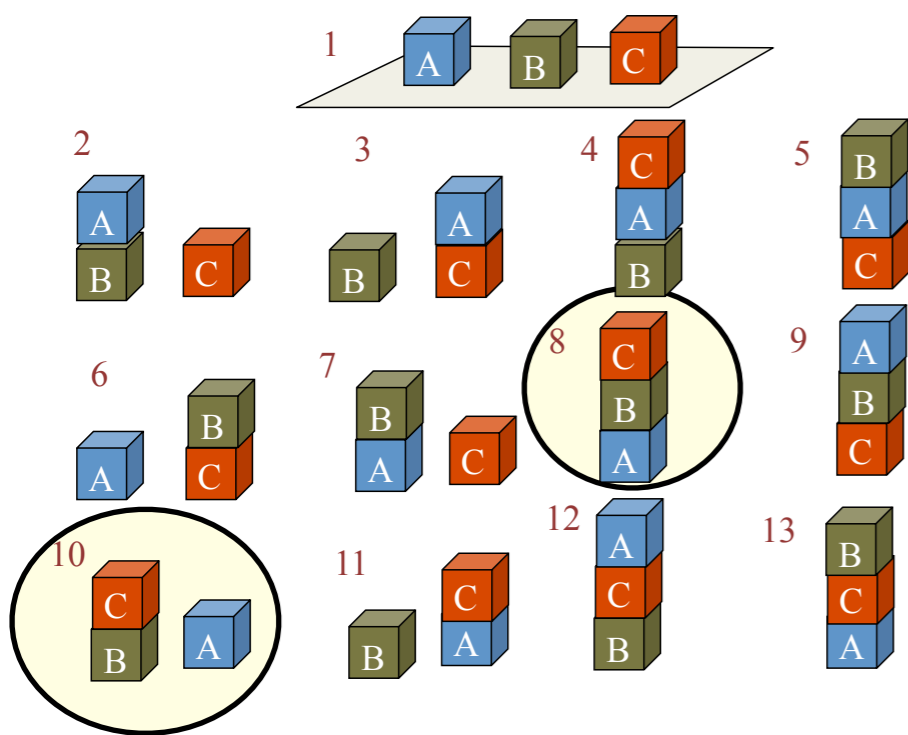
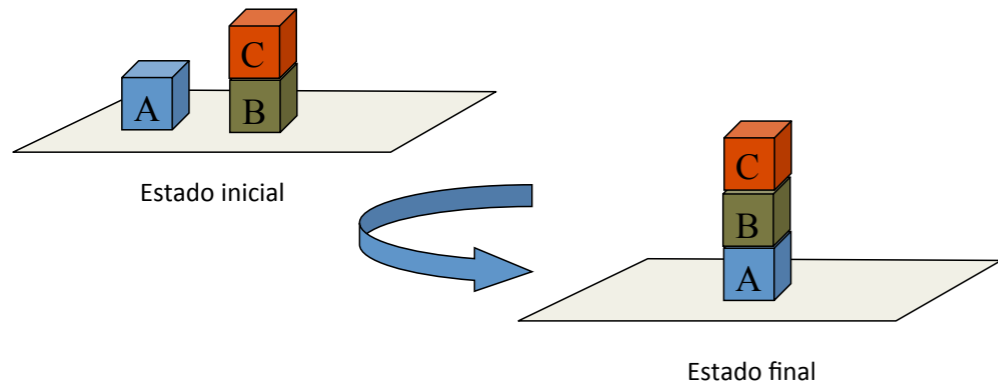


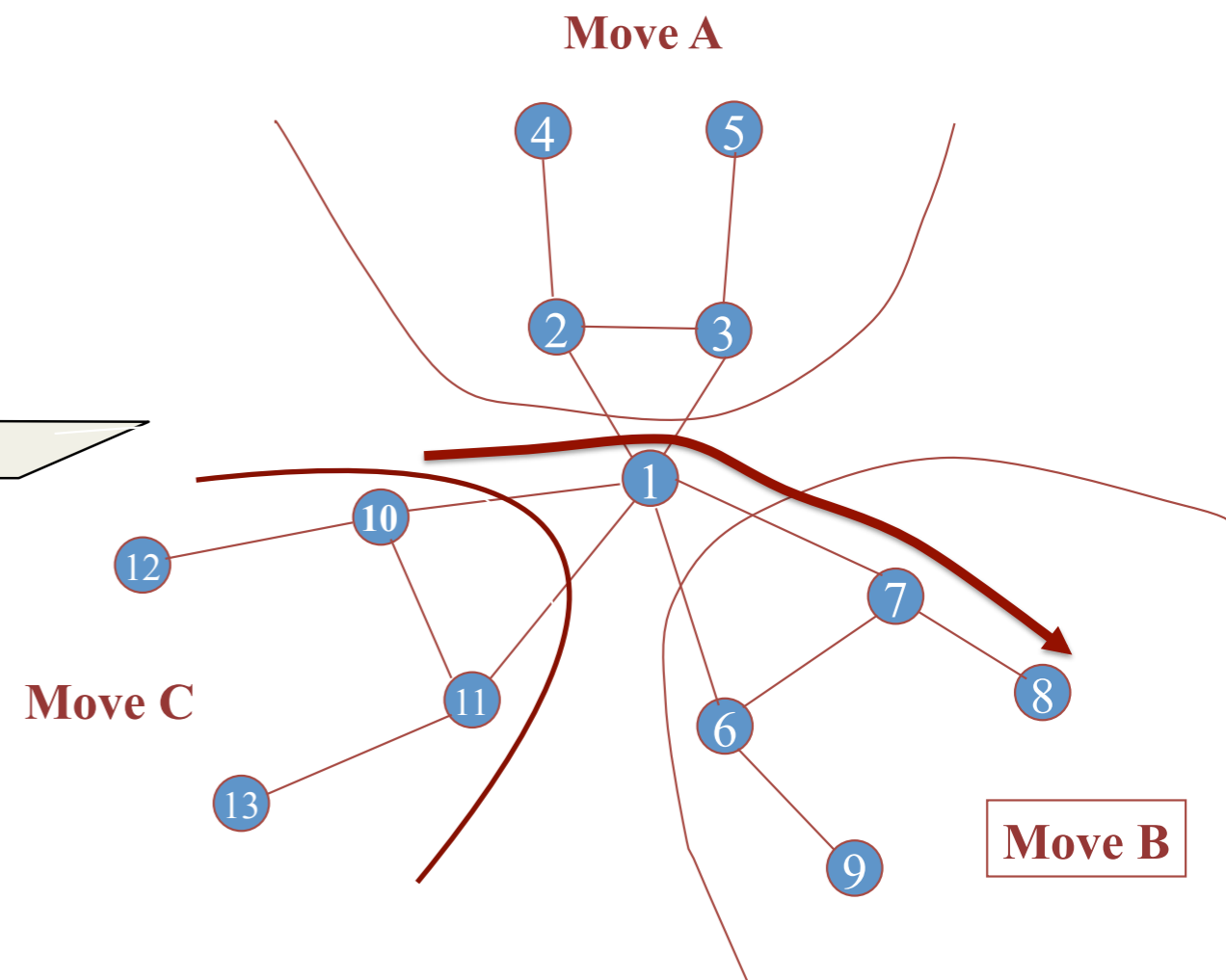
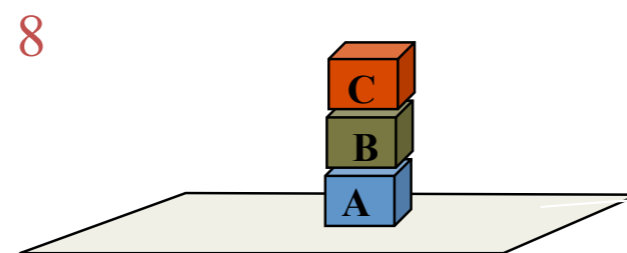
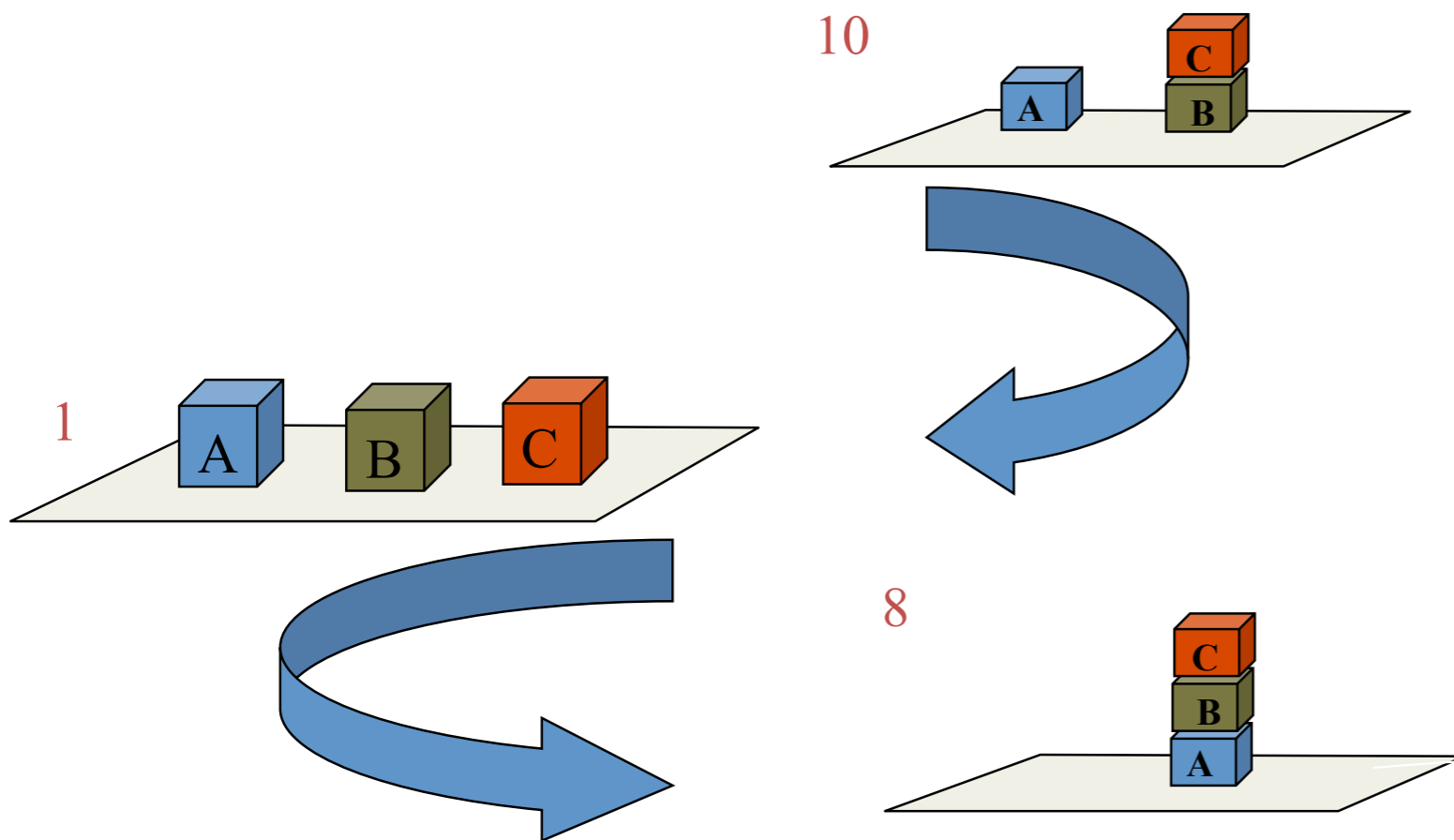
O objetivo é analisar a dinâmica deste sistema. Vamos portanto separar a “planta” do “controle”, onde a planta são os possíveis estados do “mundo de blocos” e os movimentos necessários para passar de um estado para o outro (análise do espaço de estados). Existe um agente dinâmico capaz de operar as mudanças na planta, isto é, capaz de realizar transições: o robô. Por enquanto 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 2021

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.

Wu, N., Zhou, M.; System Modeling and Control with Resource-Oriented Petri nets, CRC Press, Taylor & Francis, 2010.

PMR 5237

3o. período de 2021

L - média das listas de exercício

n_l - número de listas feitas

n_t - número total de listas

M_l - média dos milestones

n_m - número de milestones feitos

n_r - número total de milestones

N_f - nota do artigo final

$$M_f = (3 \cdot (n_l/n_t)L + 3 \cdot (n_m/n_r)M_l + 4 \cdot T_f) / 10$$

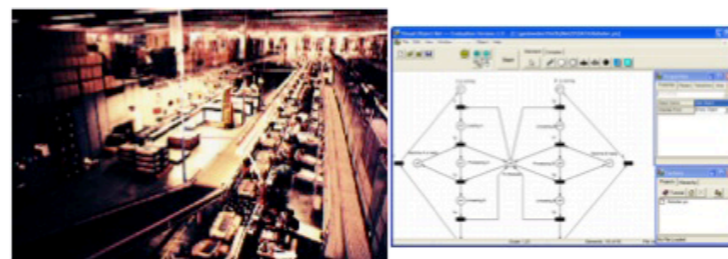
Suporte a disciplinas : e-disciplinas

edisciplinas.usp.br

PMR5237-Modelagem e Design de Sistemas Discretos em Redes de Petri

Início / Meus Ambientes / 2021 / EP / PMR / PMR5237--2021

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, quando foram propostas pela primeira vez por Carl Adam Petri, as redes de Petri vêm ampliando o seu escopo de aplicação. Paralelamente, o formalismo também foi ampliado e aperfeiçoado, e várias extensões surgiram para atender a domínios de aplicação específicos. 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 foi também um requisito para sua difusão, especialmente no meio industrial. Entretanto, aplicações industriais demandam uma certa uniformidade para dar suporte a produtos (CLPs, SDCDs, por exemplo). Por isso, um comitê de pesquisadores passou a discutir a padronização das Redes de Petri, culminando com a proposição de uma norma, que por um lado atendesse à grande difusão das RdP inclusive para a área de software, análise de requisitos, planejamento automático, verificação formal, model checking, etc., e por outro atendesse à demanda do meio industrial.

O nosso curso será norteado pela norma ISO/IEC 15.909-1 que define os modelos clássicos de Redes de Petri (sem adjetivos adicionais, mas seria o modelo mais amplo das redes Place/Transition), e das redes de Alto Nível

Corresponder ✕

+ Inserir condição

Limpar filtros

Aplicar filtros

10 participantes encontrados

Nome 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 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

| <input type="checkbox"/> | Nome / Sobrenome ^ | Endereço de email | Número USP | Papéis | Grupos | Último acesso ao curso | Estado |
|--------------------------|--------------------------------------|-----------------------|------------|-----------|---------------------|------------------------|--------|
| <input type="checkbox"/> | - | - | - | - | - | - | - |
| <input type="checkbox"/> | Arthur Bugarin de Mello Marques | arthurbugarin@usp.br | 10335207 | Estudante | T- PMR5237- 4 | 2 dias 10 horas | Ativo |
| <input type="checkbox"/> | Felipe Czubieniak Rubello | felipe.rubello@usp.br | 9833364 | Estudante | T- PMR5237- 4 | 5 dias 9 horas | Ativo |
| <input type="checkbox"/> | Lucas Freitas Marcusso | lucas.marcusso@usp.br | 8993858 | Estudante | T- PMR5237- 4 | 10 dias 10 horas | Ativo |
| <input type="checkbox"/> | Yaney Gomez Correa | ygc8104@usp.br | 11889889 | Estudante | T- PMR5237- 4 | 57 dias 6 horas | Ativo |
| <input type="checkbox"/> | Eduardo Pereira Ayres Netto | eduardo.netto@usp.br | 9345620 | Estudante | T- PMR5237- 4 | 19 dias 9 horas | Ativo |
| <input type="checkbox"/> | Guilherme Pereira Marchioro Bertelli | gbertelli@usp.br | 10970972 | Estudante | T- PMR5237- 4 | 1 dia 23 horas | Ativo |

Pesquisa Avançada ?

Últimos avisos

Acrescentar um novo tópico...
(Nenhum aviso publicado.)

Calendário

◀ setembro 2021 ▶

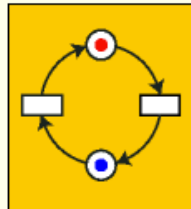
| Seg | Ter | Qua | Qui | Sex | Sáb | Dom |
|-----|-----|-----|-----|-----|-----|-----|
| | | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | | | |

- Ocultar eventos - site
- Ocultar eventos - categoria
- Ocultar eventos - curso
- Ocultar eventos - grupo
- Ocultar eventos - usuário
- Ocultar eventos - outro ⬆

Referência na Internet: Petri Nets World

www2.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

- [January 23, 2021](#): FMICS 2021 Call for Papers
- [January 23, 2021](#): Special Issue of International Journal of Parallel, Emergent and Distributed Systems: Petri/Sleptsov Net Based Technology of Programming for Parallel, Emergent and Distributed Systems
- [January 20, 2021](#): CFP: International Workshop on Petri Nets and Software Engineering , PNSE'21, Paris, France, June, 2021
- [January 19, 2021](#): Assistant Professor in Formal Methods @ DTU Compute
- [January 19, 2021](#): Two open Ph.D. positions in Probabilistic Verification Methodologies for Synthetic Biology and Nanotechnology
- [January 18, 2021](#): [Vardi-list] Postdoc Position - Analysis Techniques for DNNs
- [January 18, 2021](#): Petri Nets 2021: deadline extension
- [January 18, 2021](#): MODEL CHECKING CONTEST 2021 - (2/2) - CALL FOR TOOLS
- [January 16, 2021](#): Call for Graduate Symposium: Diagrams 2021
- [January 15, 2021](#): CFP ATAED 2021
- [January 14, 2021](#): MODEL CHECKING CONTEST 2021 - (1/2) - CALL FOR MODELS
- [January 12, 2021](#): World-Logic-Day Lecture: Diego Calvanese
- [January 11, 2021](#): RAMiCS 2021 - CFP
- [January 9, 2021](#): 2021 Alonzo Church Award - Call for Nominations
- [January 8, 2021](#): Call for tutorials: Diagrams 2021
- [January 4, 2021](#): CFP: Diagrams 2021
- [December 24, 2020](#): QEST 2021: Call for Papers
- [December 15, 2020](#): [CfP] Petri Nets 2021 in Paris + special track on ACSD (ICATPN 2021, June 20-25, 2021), 2nd call for papers

PMR5237--2021

- Participantes
- Emblemas
- Notas
- Download center
- Introdução (Aula 1)
- Modelando processos simples
- Modelando sistemas produtivos
- Redes Place/Transition
- Redes de Alto Nível
- Redes Coloridas
- Redes HLPN
- Propriedades das Redes Clássicas e de Alto Nível
- Modelagem Análise e verificação com Redes de Petri
- Análise de invariantes e

domínios de aplicação específicos. Assim, a chegada do século XXI marca um período de grandes profusões e de alguma contusão, sobre o formalismo base das redes de Petri, o que foi também um requisito para sua difusão, especialmente no meio industrial. Entretanto, aplicações industriais demandam uma certa uniformidade para dar suporte a produtos (CLPs, SDCDs, por exemplo). Por isso, um comitê de pesquisadores passou a discutir a padronização das Redes de Petri, culminando com a proposição de uma norma, que por um lado atendesse à grande difusão das RdP inclusive para a área de software, análise de requisitos, planejamento automático, verificação formal, model checking, etc., e por outro atendesse à demanda do meio industrial.

O nosso curso será norteado pela norma ISO/IEC 15.909-1 que define os modelos clássicos de Redes de Petri (sem adjetivos adicionais, mas seria o modelo mais amplo das redes Place/Transition), e das redes de Alto Nível

ISO/IEC 15.909-2 (que tem como modelo mais difundido as redes coloridas). Trataremos brevemente do modelo de transferência - que permite que modelos possam migrar entre plataformas e software de apoio) em XML, que é o PNML (Petri Nets Markup Language), e incluiremos algumas extensões - ISO/IEC 15.909-3 - como os gates inibidores, os lugares de conexão, e o modelo hierárquico.

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

Avisos

Fórum de notícias

Leitura da Semana

[A Gentle Introduction to Graph Theory](#)

Uma breve introdução para os que quiserem ter mais informação sobre a teoria de grafos.

[Acesso ao Petri Nets World](#)

Ferramentas de software



PIPE (Windows, Linux, Mac)

~~HPSIM (Windows) Renew~~

CPN Tools

ePNK, GHENeSys (?)

PIPE2

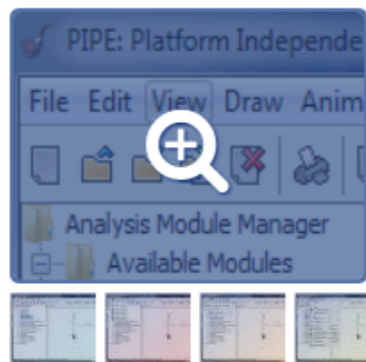
DOWNLOAD NOW

5,674 downloads · Updated: February 2, 2015 · GPL

4.3/5 2

- PIPE2 4.3.0
- ADD TO WATCHLIST
- SEND US AN UPDATE

5 SCREENSHOTS:



RUNS ON:

Windows All

FILE SIZE:

28.5 MB

MAIN CATEGORY:

Others

DEVELOPER:

Advertisement

CorelCAD

Get the New CorelCAD 2021

DOWNLOAD

REVIEW FREE DOWNLOAD SPECIFICATIONS 100% FREE REPORT MALWARE

A suitable environment packed with various drawing tools and analysis modules to create thorough Petri nets and run simulations to test your projects

Statistics and representations of ideas help us greatly because we can better visualize each step and the whole process to check a project for errors before actual implementation. Based on action and consequence, a Petri net can be used to simulate various business, industry or academia situations. In this regard, PIPE2 gives you the possibility to easily design and run Petri nets.

Intuitive design and various examples

When it comes to visuals, the application is based on a simple, but highly-intuitive interface. Most of the main window represents your interactive workspace where you insert needed elements to run a simulation. Most creation and analysis tools are

Advertisement

Execute o Windows no Mac

Parallels® Desktop 17 para Mac

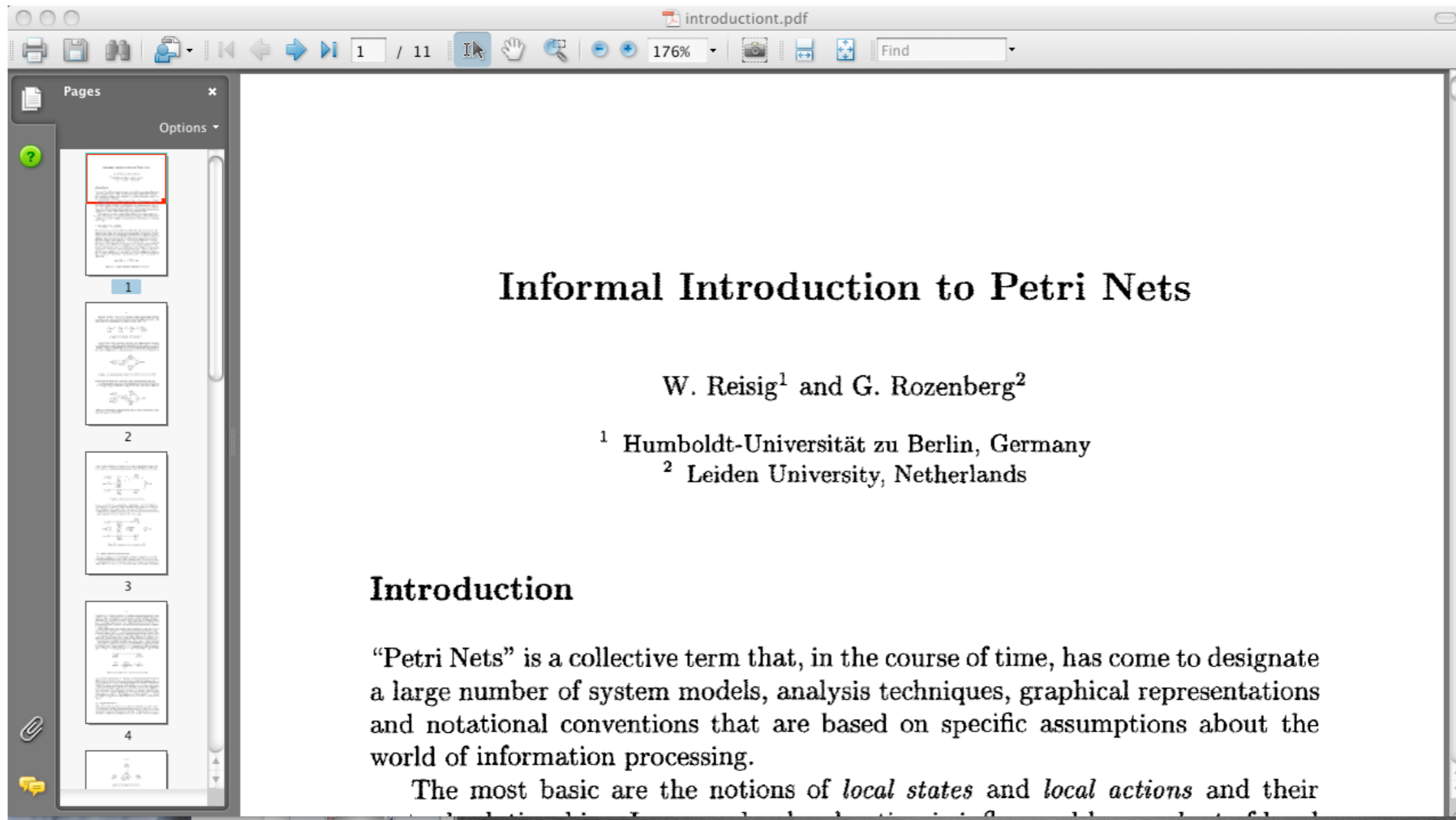
Execute o Windows no Mac

Execute o Windows no Mac com chip M1 da

We use cookies to improve your experience. By using the site you agree to the use of cookies. For more information and options regarding cookies and personal data see our [Cookies Policy](#) and [Privacy Policy](#)

Accept cookies & close

Leitura da semana



introductiont.pdf

1 / 11 176% Find

Pages

Options

1

2

3

4

Informal Introduction to Petri Nets

W. Reisig¹ and G. Rozenberg²

¹ Humboldt-Universität zu Berlin, Germany
² Leiden University, Netherlands

Introduction

“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


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



[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 1 usando o PIPE2: Acesse a página da Softpedia e baixe o sistema PIPE2 v4.3 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;



Até ~~sempre~~ próxima aula!