

**Prof. Dr. João José Neto**  
Escola Politécnica da USP  
Laboratório de Linguagens e Técnicas Adaptativas

# Uma Breve Introdução à Adaptatividade e à Tecnologia Adaptativa



LABORATÓRIO DE LINGUAGENS E TÉCNICAS ADAPTATIVAS

# Lab. de Ling. e Técnicas Adaptativas



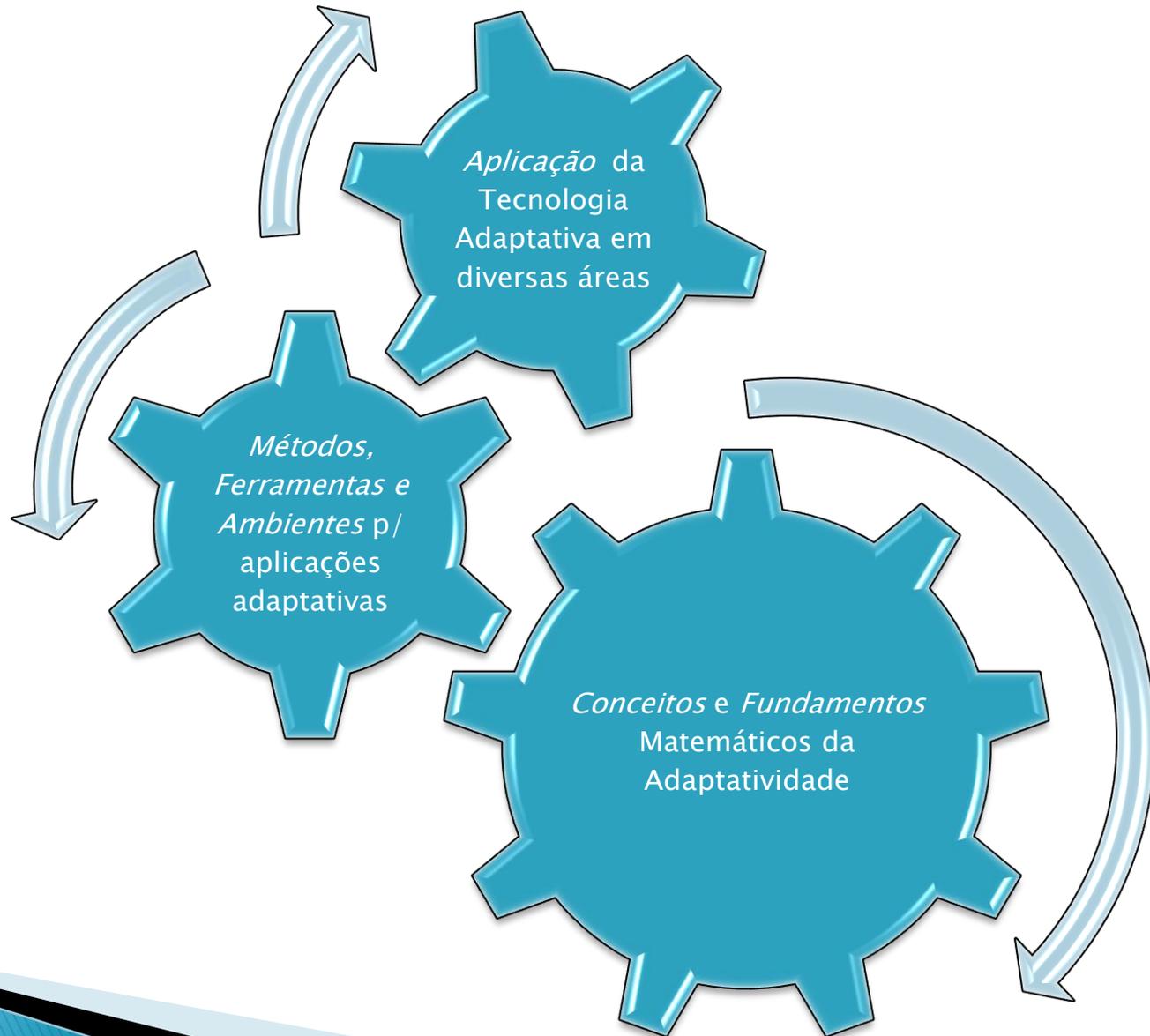
O LTA é um  
laboratório filiado  
ao PCS-EPUSP



Focos:

- Linguagens
- Adaptatividade
- Disseminação da  
Tecnologia Adaptativa

# Vertentes principais de atuação:



# Conceitos e Fundamentos

Dispositivos guiados por regras

Conceito de Adaptatividade

Arquitetura de máquinas aderentes à adaptatividade

Adaptatividade estratificada

Dispositivos adaptativos

# Dois exemplos

- modelo de computação
- equivalência com Máquina de Turing

Autômato adaptativo

- prática para desenvolvimento de software
- simulação de autômato adaptativo

Tabela de Decisão Adaptativa

# Conceito informal de Adaptatividade

Dispositivo  
definido com  
**regras estáticas:**

- Comportamento específico, invariante

Dispositivo  
definido com  
**regras adaptáveis:**

- Comportamentos diversificados, selecionáveis

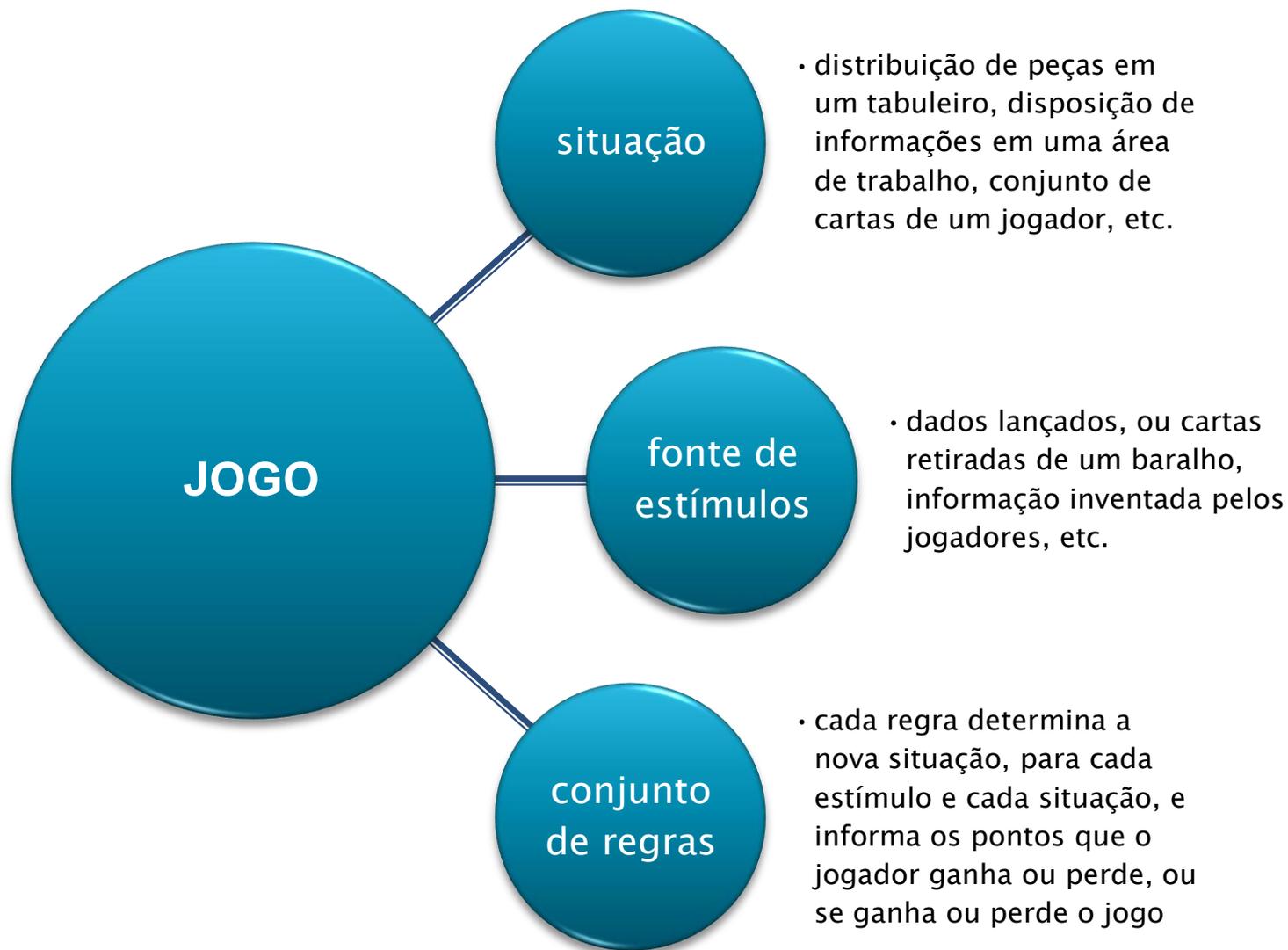
Dispositivo  
definido com  
**regras parametrizadas:**

- Comportamento geral, personalizado

Dispositivo  
definido com  
**regras adaptativas:**

- Comportamento dinâmico, auto-modificável

# Intuição: Exemplo prático – um jogo



# Funcionamento típico de um jogo



# O que seria um jogo adaptativo?

Jogo tradicional:

- conjunto de regras estabelecido de antemão

Jogo adaptativo:

- em operação, permite modificar suas próprias regras

A modificação do conjunto de regras

- resulta da aplicação de regras do próprio jogo



## Regras adaptativas

- São as que, executando Ações Adaptativas, permitem modificar o conjunto de regras



## Modifica-se o conjunto de regras

- removendo regras existentes
- incluindo novas regras



## Jogos adaptativos

- São jogos que contêm regras adaptativas
- Exigem atenção constante do jogador

# Conversão de jogo tradicional em adaptativo

Escolhe-se o jogo tradicional a ser convertido em adaptativo

Define-se a operação do jogo como um conjunto de regras

Escolhem-se regras às quais associar ações adaptativas

- Essa associação é feita conforme a intenção do projetista

Definem-se convenientemente as Ações Adaptativas

- Objetivo: manter interessante o jogo para seu usuário

Associam-se as Ações Adaptativas às regras escolhidas

- Com o cuidado de manter a coerência do conjunto de regras

Jogos adaptativos exigem constante atenção às regras dinâmicas

# Introdução à Adaptatividade e à TA

## A Adaptatividade

- Aplica-se a dispositivos definidos por conjuntos de regras
- Permite obter soluções com o auxílio de qualquer técnica clássica
- Altera dinamicamente o conjunto de regras que define o dispositivo



Não acrescenta à computação realizada nenhum poder adicional



Melhora o poder de expressão de atividades computacionais intrincadas



Permite criar soluções atraentes para diversas situações complexas.

Isso permite seu uso isolado ou combinado a outras técnicas



Uso natural em aplicações inteligentes, dotadas de aprendizagem.



Desenvolvimentos recentes permitem que procedimentos adaptativos alterem sua própria forma de aprender

# Adaptatividade

Encontram-se na literatura diversos sentidos para a palavra “*Adaptatividade*” e outras similares.

Neste contexto, Adaptatividade é a capacidade que um dispositivo tem de *modificar de forma espontânea seu comportamento*, sem auxílio externo

Confere-se adaptatividade, a um dispositivo definido por regras, incorporando-se-lhe um mecanismo que lhe permita *modificar*, em operação, *o próprio conjunto de regras* que define seu comportamento

# Principais Aplicações

Uso de dispositivo adaptativo como *modelo de computação*

*Jogos eletrônicos, Arte por computador, Ensino por computador*

*Formalização puramente sintática de linguagens de programação*

Aceitação e Análise de linguagens *dependentes de contexto*

Engenharia de Software, Programação evolutiva

*Inteligência Artificial, Robótica, Automação*

Representação e Manipulação de *Conhecimento*

*Modelagem, Simulação, Otimização, Controle*

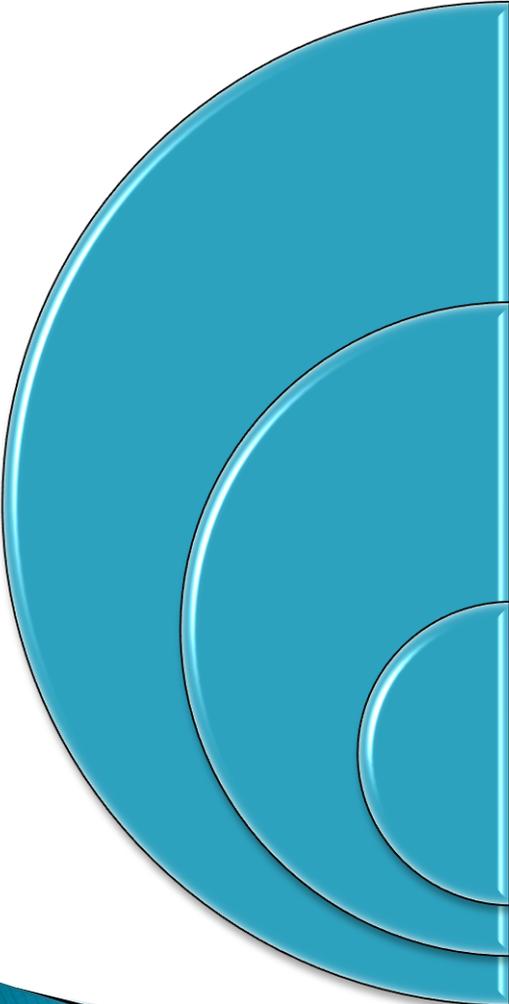
Formulação, Análise e Processamento de *Linguagem Natural*

Processamento de Sinais, *Reconhecimento de Padrões*

*Inferência, Tomada de Decisão, Classificação, Diagnóstico*

Paradigma e *Linguagens de Programação Adaptativas*

# Tecnologia Adaptativa



A adaptatividade exige, para o desenvolvimento dos seus conceitos, boa fundamentação em *matemática discreta*

*Tecnologia Adaptativa* refere-se àquela que aplica o conceito de adaptatividade para a resolução eficiente de certos problemas da prática

O domínio dessa tecnologia requer envolvimento nas três vertentes pesquisadas no LTA:

- *Teoria* – para fornecer os fundamentos matemáticos
- *Ferramentas* – para facilitar o desenvolvimento de aplicações
- *Aplicações* – para resolver problemas de diversas áreas

# Marcos principais da pesquisa

Formalização sintática de linguagens de programação (1984-7)

Autômatos de Pilha Estruturados (1987)

Construção Automática de reconhecedores/analísadores (1988)

Autômatos (de Pilha Estruturados) Adaptativos (1994)

Outros modelos adaptativos (statechart, rede de Markov) (1998-9)

Autômato Adaptativo como Modelo de Computação (2000)

Dispositivos Adaptativos gerais, guiados por Regras (2001)

Autômatos Finitos Adaptativos (2003)

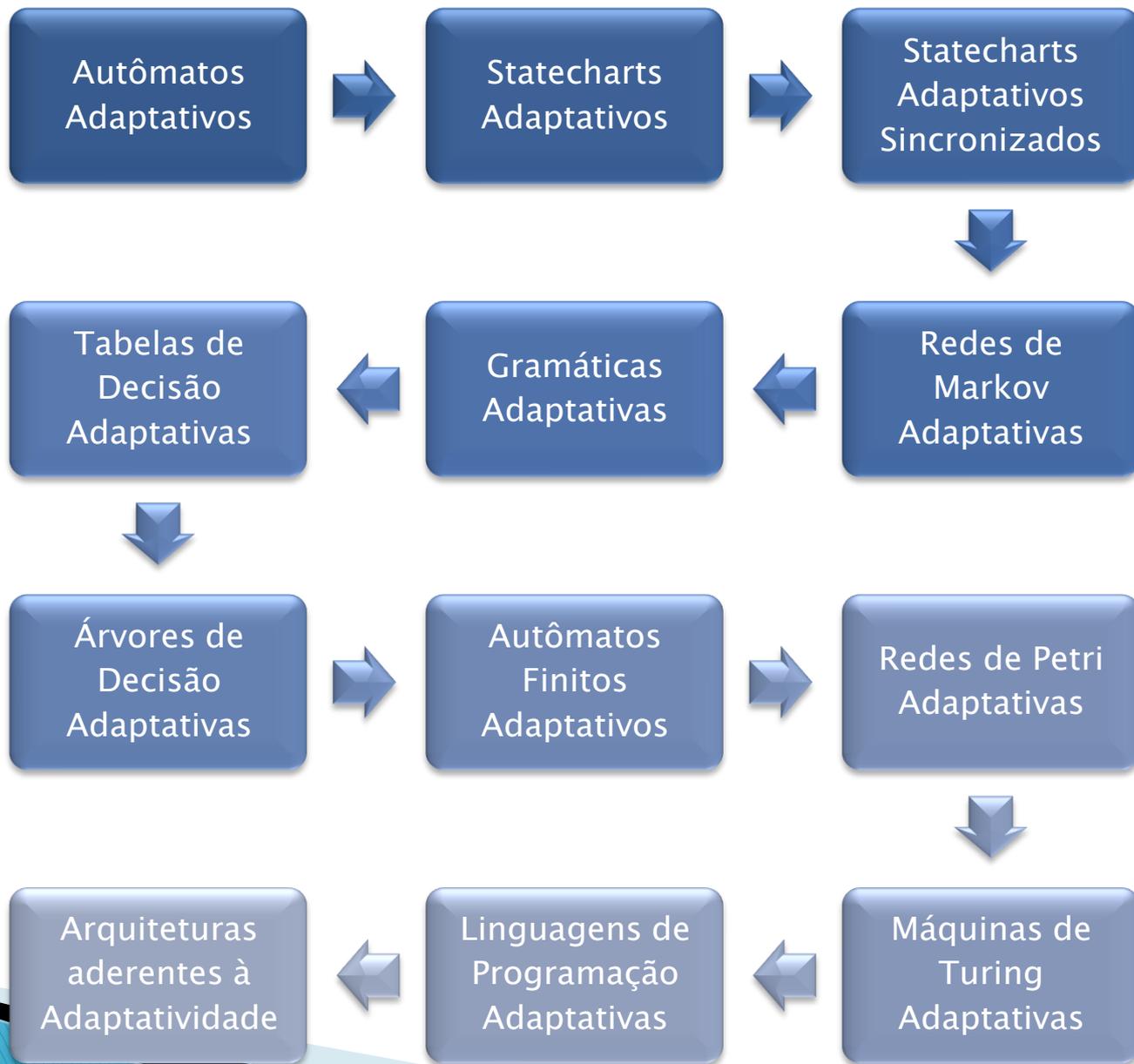
Tabelas e Árvores de Decisão Adaptativas (2001-3)

Geradores de Meta-ambientes (Adaptativos) (2007)

Linguagens de programação adaptativas (2008)

Dispositivos Adaptativos de ordens superiores (2008)

# Dispositivos Adaptativos mais Explorados



# Generalizações do Conceito

## 2001 – conceito de dispositivo adaptativo geral

- Dispositivo adaptativo = dispositivo subjacente + camada adaptativa
- Resultado: generalidade e independência entre as partes do dispositivo
- Propriedades: aprender; modelar, armazenar, manipular conhecimento

## 2008 – conceito de dispositivo com adaptatividade multinível

- O dispositivo subjacente pode também ser adaptativo
- Permite modificar as ações adaptativas (o modo de modificar regras)
- Incorpora a capacidade de aprender a aprender
- Dá margem à modelagem de fenômenos “criativos”

# Extensões dos Dispositivos Adaptativos

Inclusão do conceito de tempo (relógio)

Inclusão de elementos estocásticos (probabilidades)

Inclusão de elementos para a representação de concorrência

Inclusão de elementos de armazenamento (registradores)

Inclusão de elementos aritméticos (contadores, acumuladores)

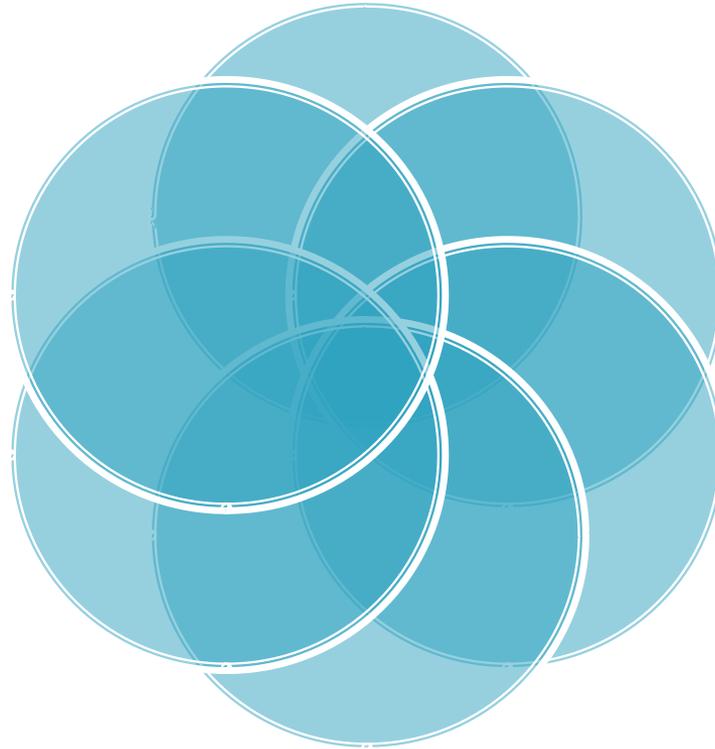
Inclusão de elementos lógicos (comparadores)

Extensões híbridas, incluindo procedimentos computacionais e elementos não discretos



# Algumas Ferramentas

**STAD e STAD/S:**  
ambientes de  
descrição/simulação de  
statecharts adaptativos



**TDA:** Ambiente de  
visualização e de  
simulação de tabelas de  
decisão adaptativas

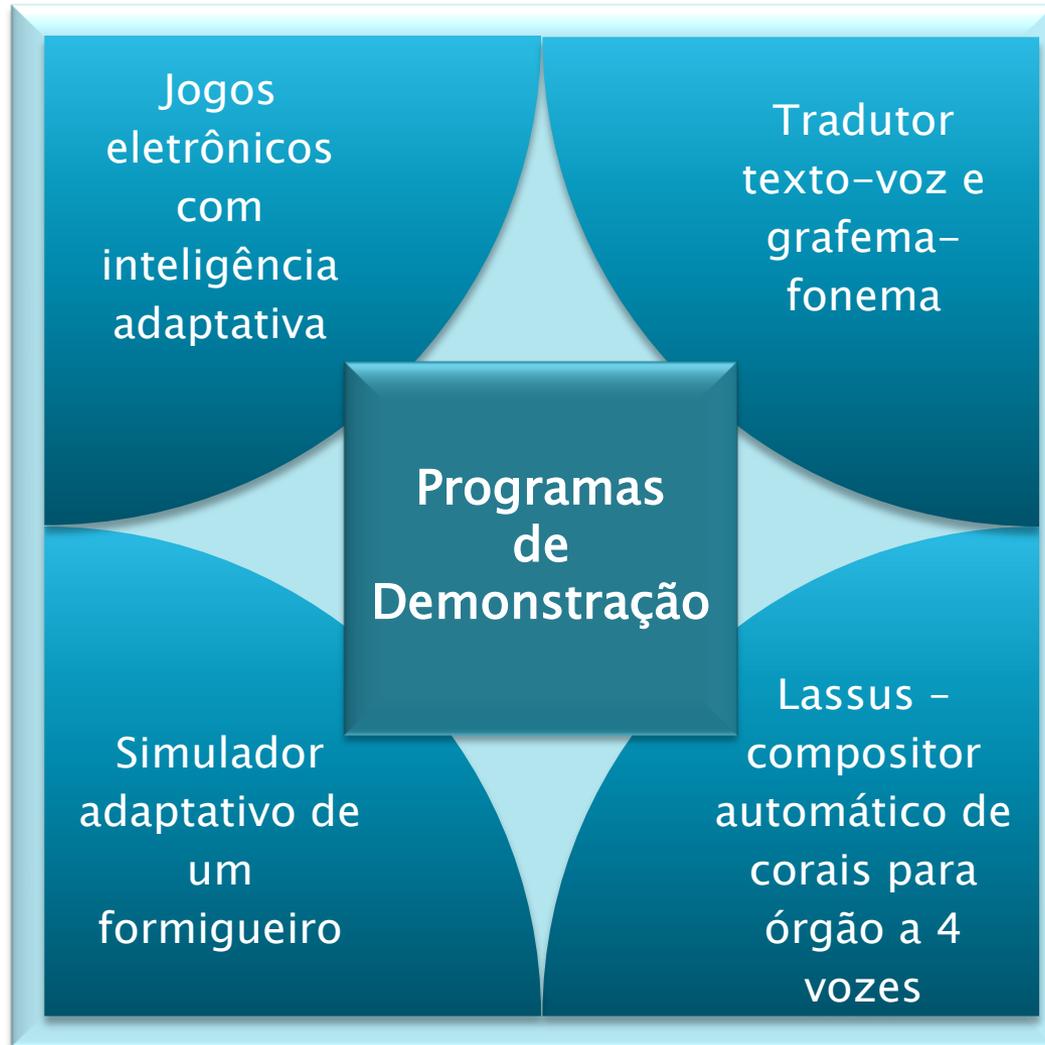
**Metamb:** Gerador de  
meta-ambientes, que  
permite descrever o  
formalismo a simular, e  
gera automaticamente  
um ambiente de  
desenvolvimento para o  
formalismo em questão

**RSW:** descrição e  
simulação de autômatos  
adaptativos, e  
construção automática  
de reconhecedores  
sintáticos

**AdapTools:** descrição e  
simulação de autômatos  
adaptativos, com  
visualização gráfica  
animada

**AdapTree:** descrição e  
simulação de árvores de  
decisão adaptativas

# Alguns programas de demonstração



# Aplicações

## Teoria

- modelo de computação baseado em autômatos adaptativos

## Linguagens

- em elaboração, as bases para um paradigma de programação baseado na adaptatividade

## Inteligência Artificial

- Representação, aquisição e manipulação do conhecimento; problemas complexos

## Tomada de decisão

- representação em árvores/tabelas automodificáveis, de mecanismos de tomada de decisão

## Processamento de Linguagem Natural

- representação, análise, tradução automática, conversão texto-voz, busca semântica etc.

## Segurança

- criptografia, compressão de dados, inferência, classificação e reconhecimento de padrões etc.

## Jogos/Simuladores

- técnicas inteligentes em jogos e simuladores usando formalismos adaptativos

# Formação e Pesquisa em Tecnologia Adaptativa

LTA-EPUSP

- <http://www.pcs.usp.br/~lta>
- Downloads

Publicações  
diversas

- Artigos
- Teses
- Dissertações
- Livros

Pós-graduação  
no LTA

- Disciplina de Pós-Graduação
- Mestrado
- Doutorado
- Pós-Doutorado

# Grupos de pesquisa

UCDB  
(Campo Grande - MS)

UFMS (Coxim - MS)

PUCP - (Lima, PE)

UCSP - (Arequipa, PE)

PUC SP (S. Paulo - SP)

UPM Mackenzie (S. Paulo - SP)

FEI (S. Bernardo do Campo - SP)

IMES (S. Caetano do Sul - SP)

FSA (Sto. André - SP)

UFSCar (S. Carlos - SP)

Unilins (Lins - SP)

FEMA (Assis - SP)

UNIVASF (Petrolina - PE)



## PRÉ-HISTÓRIA

- 1963 - Conway - diagramas separáveis de transições
- 1973 - Lomet - formalização dos diagramas separáveis de transições
- 1981 - Neto e Magalhães - autômatos de pilha estruturados
- 1987 - Neto - livro: compiladores com autômatos de pilha estruturados

## DISPOSITIVOS ADAPTATIVOS PARTICULARES

- 1988 - Neto - Concepção inicial dos Autômatos Adaptativos
- 1993 - Shutt - Gramáticas Automodificáveis
- 1993 - Neto - Autômatos Adaptativos
- 1995 - Almeida - Statecharts Adaptativos
- 1997 - Santos - Statecharts Adaptativos Sincronizados
- 1999 - Pereira - RSW ambiente p/ gramáticas dependentes de contexto
- 1999 - Neto, Pariente, Leonardi - Wirth → Aut. de Pilha Estruturados
- 2000 - Basseto - Redes de Markov Adaptativas para música
- 2000 - Iwai - Gramáticas Adaptativas
- 2000 - Jackson - Gramáticas §

## DISPOSITIVOS ADAPTATIVOS GERAIS

- 2001 - Neto - Dispositivos Adaptativos gerais
- 2003 - Pistori - Árvores de Decisão Adaptativas
- 2005 - Ricchetti - ger. de parsers baseado em aut. de pilha estruturados
- 2006 - Matsuno - Inferência de linguagens regulares e livres de contexto
- 2007 - Camolesi - gerador de ambientes p/ formalismos adaptativos
- 2008 - Freitas - Linguagem de Programação Adaptativa

## ADAPTATIVIDADE MULTINÍVEL

- 2008 - Neto - Dispositivos Adaptativos Multinível

# Evolução Nacional da Tecnologia Adaptativa (~2003)

( Extraída da tese de Hemerson Pistori, 2003 )

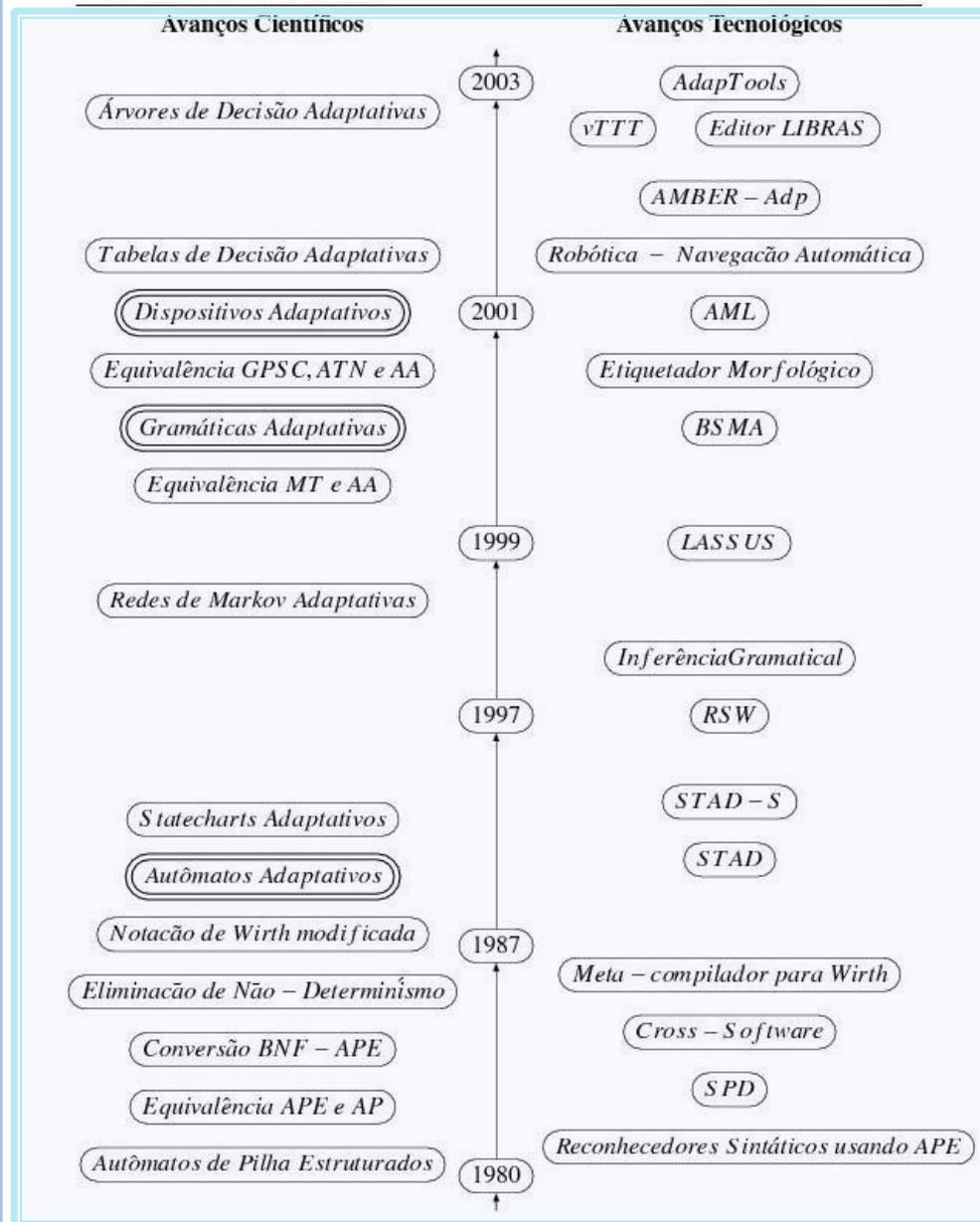


Figura 2.1: Principais acontecimentos nacionais na história dos dispositivos adaptativos

# Exemplo simples Dominó Adaptativo



# Dominó adaptativo – Configuração inicial

## Conjunto (inicial) de regras do jogo (adaptativo):

$J = \{(0,0,-), (1,1,-), (2,2,-), (3,3,-), (4,4,-), (5,5,-), (6,6,*)\}$

## Conjuntos de peças dos 4 jogadores (sorteados):

$A = \{(0,1), (0,5), (2,2), (2,6), (3,3), (4,5), (4,6)\}$

$B = \{(0,6), (1,5), (1,6), (2,3), (2,4), (2,5), (6,6)\}$

$C = \{(0,0), (0,4), (1,1), (1,2), (3,4), (3,5), (5,6)\}$

$D = \{(0,2), (0,3), (1,3), (1,4), (3,6), (4,4), (5,5)\}$

## Para jogar:

- ▶ Usar as regras usuais do dominó, a menos do mapeamento das extremidades da mesa segundo o conjunto J de regras adaptativas.
- ▶ Se usar a regra marcada com \*, o jogador poderá:
  - permutar o segundo número entre duas regras de J
  - mudar de posição o asterisco, movendo-o para outra regra de J
  - ou ainda, manter inalterado o conjunto de regras J

# Dominó adaptativo: Notação

Conjunto de regras:

$J = \{(0,0,-), (1,1,-), (2,3,-), (3,6,-), (4,2,-), (5,5,-), (6,4,*)\}$

Uma regra

Primeiro número: extremidade da mesa

Segundo: substitui o primeiro, nesta jogada

Traço: significa que a regra não é adaptativa

Asterisco: significa que a regra é adaptativa

# Lance 34 – Jogador B: (3,4) \*

Regras:  $\{(0,0,-), (1,1,-), (2,3,-), (3,6,-), (4,2,-), (5,5,-), (6,4,*)\}$

Mapeamento de L

$A = \{ (0,3), (1,4), (2,5), (3,6) \}$

Mapeamento de R

$B = \{ (0,6), (1,5), (1,6), (2,3), (2,4), (2,5) \}$

Indica que há uma regra adaptativa associada a este número.

$C = \{ (0,0), (0,4), (1,1), (1,2), (3,5), (5,6) \}$

Próximo

$D = \{ (0,3), (1,3), (1,4), (3,6) \}$

Indica o próximo a jogar

Sublinhado indica o próximo lance

Mesa:

Última peça utilizada

L

R

Peça do próximo lance

Notação Utilizada

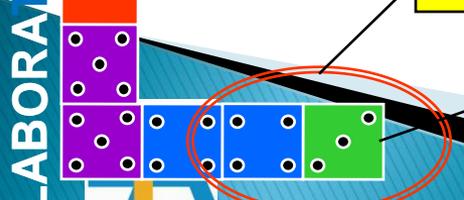
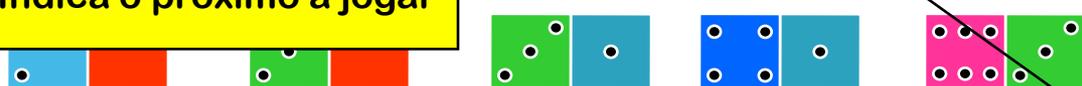
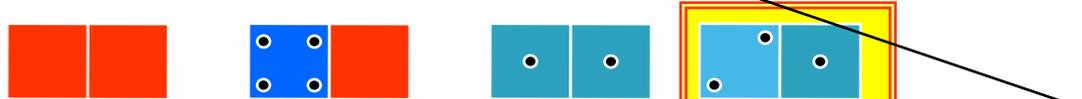
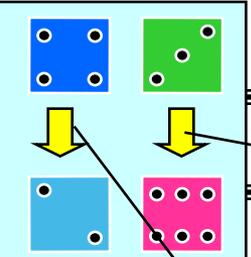
Indica uso de regra adaptativa

Anuncia a alteração das 2 regras após o próximo lance

NS E TÉCNICAS

LABORATÓRIO DE LINGUA

LABORATÓRIO DE LINGUA



# Exemplo de um lance: Jogador D: (5,5)



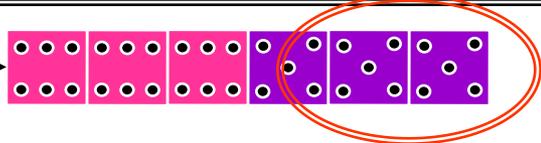
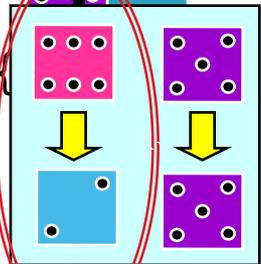
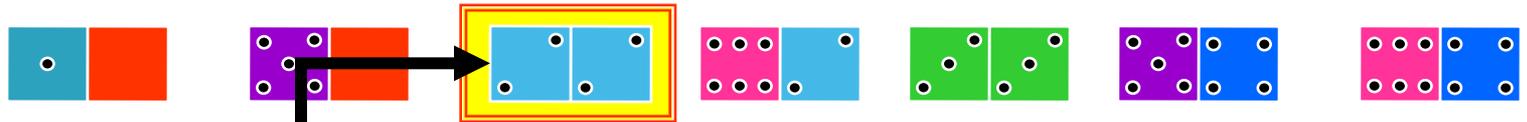
Regras:  $\{(0,0,-), (1,1,-), (2,6,-), (3,3,-), (4,4,-), (5,5,*), (6,2,-)\}$

$A = \{ (0,1), (0,5), (2,2), (2,6), (3,3), (4,5), (4,6) \}$

$B = \{ (0,6), (1,5), (1,6), (2,3), (2,4), (2,5) \}$

$C = \{ (1,1), (1,2), (3,4), (3,5) \}$

$D = \{ (0,2), (0,3), (1,3), (1,4), (3,6), (4,4) \}$



Mesa:

O jogador D vai alterar a posição do asterisco da regra  $(5,5,*)$  para a regra  $(3,3,-)$ , vigorando então  $(5,5,-)$  e  $(3,3,*)$  a partir do próximo lance.



# Aplicações

Dependência de contexto e linguagens de programação

Processamento de cadeias

Processamento de linguagem natural

- Análise de texto e identificação de conteúdo
- Análise e síntese de fala
- Busca semântica
- Tradução automática

Inferência. Resolução de problemas. Aplicação: ensino por computador

Controle, Planejamento e Decisão. Aplicação em administração e comércio

Reconhecimento de padrões. Aplicação: visão computacional, classificação, diagnóstico

Robótica: mapeamento, identificação de formas, roteamento, navegação autônoma

Simulação e Jogos. Gestão de recursos em redes e em sistemas operacionais

Ferramentas: AdapTools, MetAmb, Tabelas de decisão

Inteligência artificial: Representação, aquisição e manipulação de conhecimento



# Mecanização da Aprendizagem

Um autômato adaptativo pode representar o processo de aprendizagem incremental através da incorporação passo a passo de informação extraída dos dados de entrada.

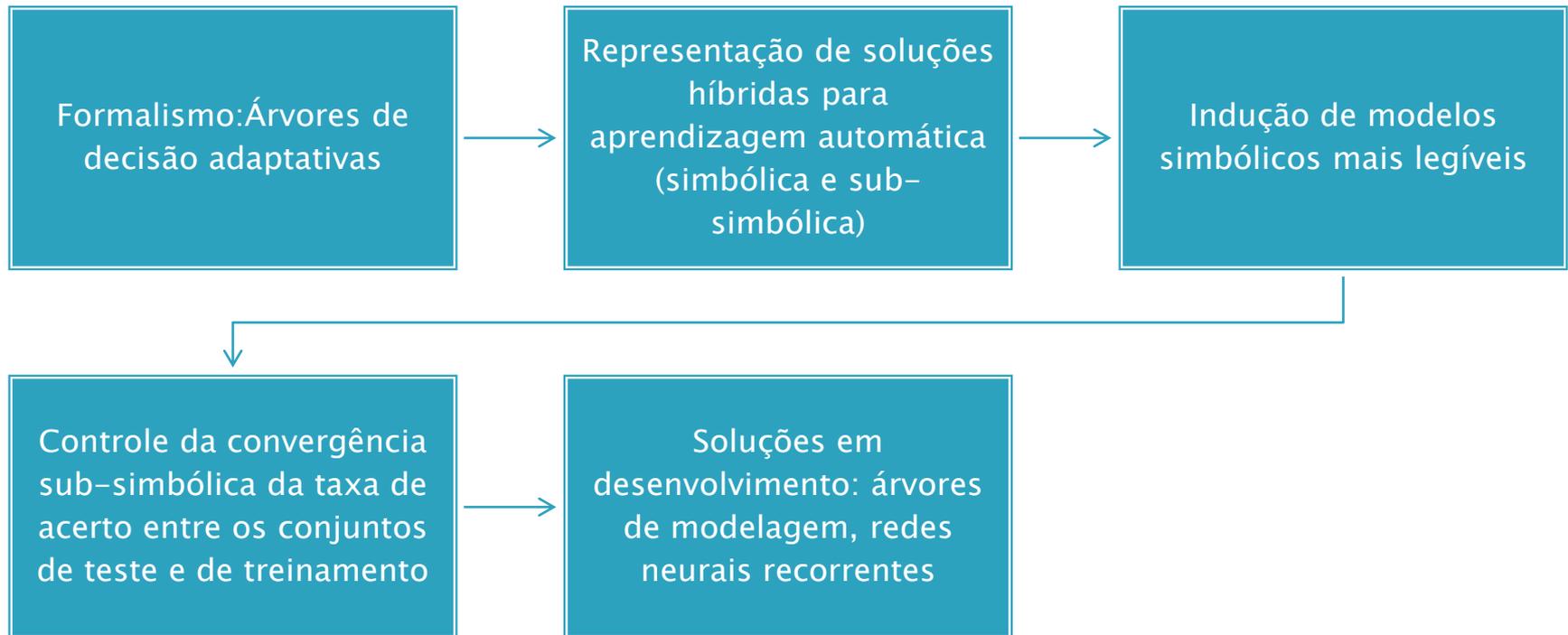


Um caso representativo já estudado é o coletor de nomes, no qual o conhecimento é o conjunto de nomes encontrado



O mesmo princípio pode ser aplicado em outras situações, como ensino por computador, controladores inteligentes, etc.

# Tomada de decisão – AdapTree



# Autômato Adaptativo

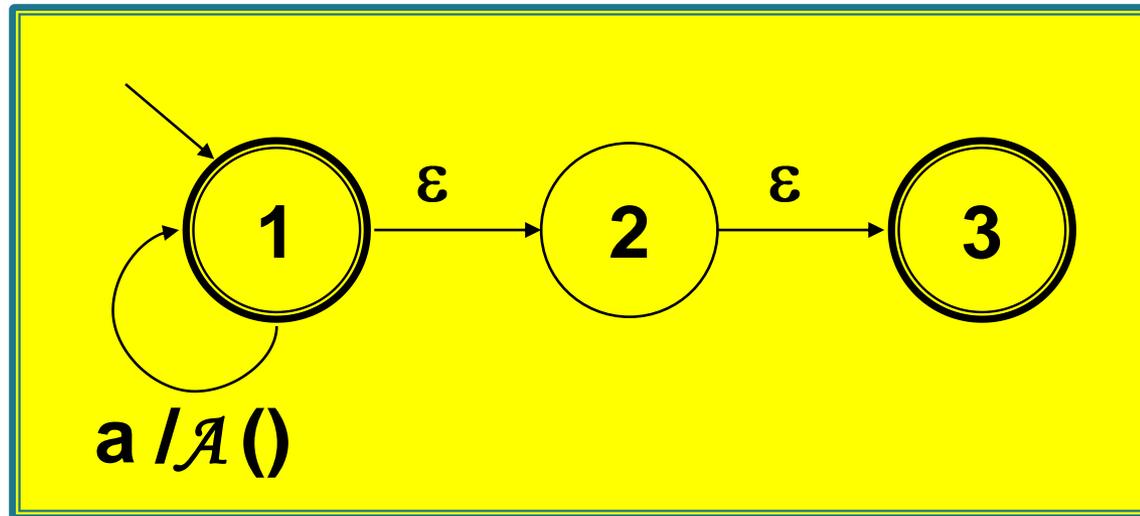


# Exemplo:

Autômato Adaptativo que aceita  
(linguagem dependente de contexto)

$a^n b^{2n} c^{3n}$

Topologia inicial:



Cadeia de entrada a ser reconhecida:

$a^2 b^4 c^6$

# Definição da ação adaptativa $\mathcal{A}()$ :

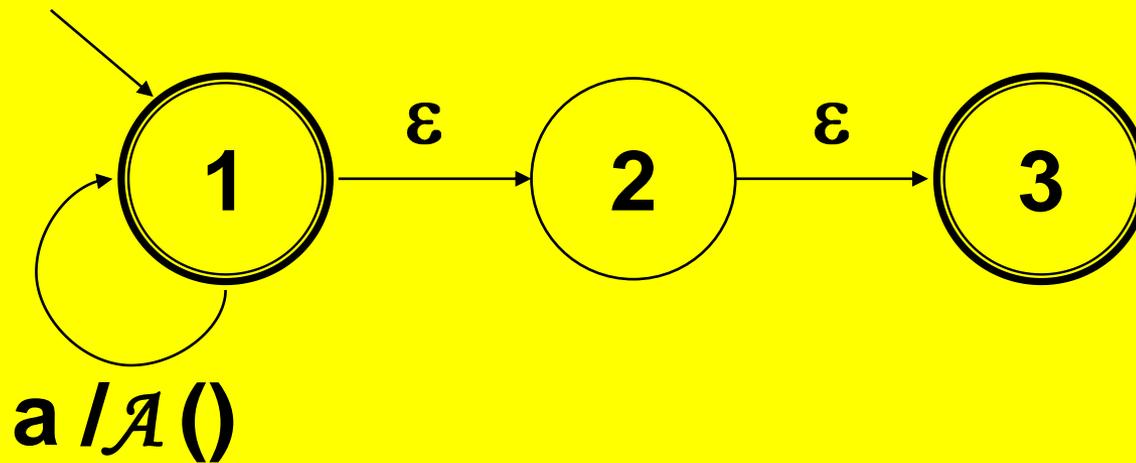
$$\mathcal{A}() = \{ \quad ? [ \textcircled{x} \xrightarrow{\varepsilon} \textcircled{2} ]$$

$$\quad ? [ \textcircled{2} \xrightarrow{\varepsilon} \textcircled{y} ]$$

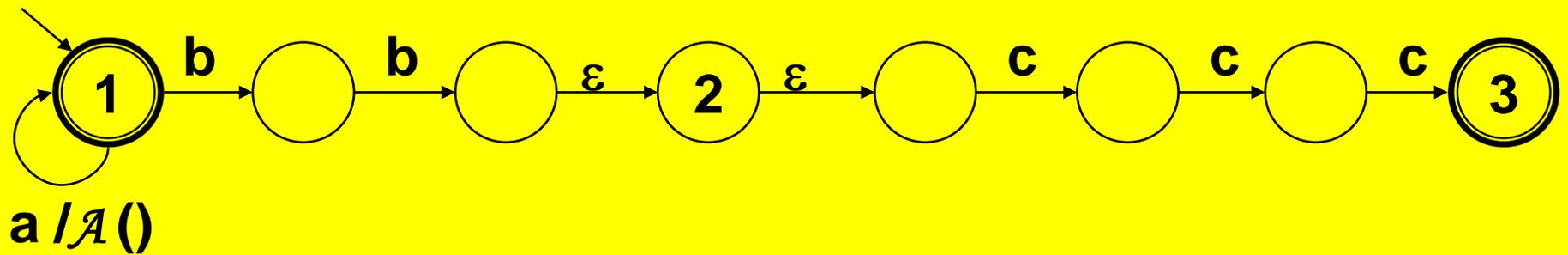
$$\quad - [ \textcircled{x} \xrightarrow{\varepsilon} \textcircled{2} \xrightarrow{\varepsilon} \textcircled{y} ]$$

$$+ [ \textcircled{x} \xrightarrow{b} \textcircled{\quad} \xrightarrow{b} \textcircled{\quad} \xrightarrow{\varepsilon} \textcircled{2} \xrightarrow{\varepsilon} \textcircled{\quad} \xrightarrow{c} \textcircled{\quad} \xrightarrow{c} \textcircled{\quad} \xrightarrow{c} \textcircled{y} ] \quad \}$$

# Topologia inicial do autômato:

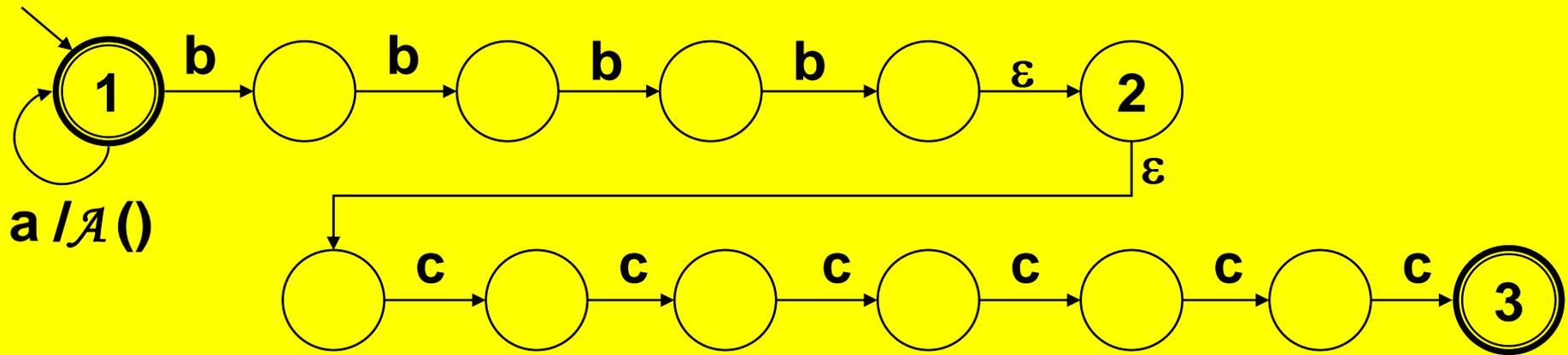


# Aparência do autômato após aceitar o primeiro “a” da cadeia $a^2b^4c^6$ :



# Aparência do autômato após aceitar o segundo "a" da cadeia $a^2b^4c^6$ :

CAS A



(Esta topologia permanecerá estável até o final da aceitação da cadeia de entrada)

LABORATÓRIO DE



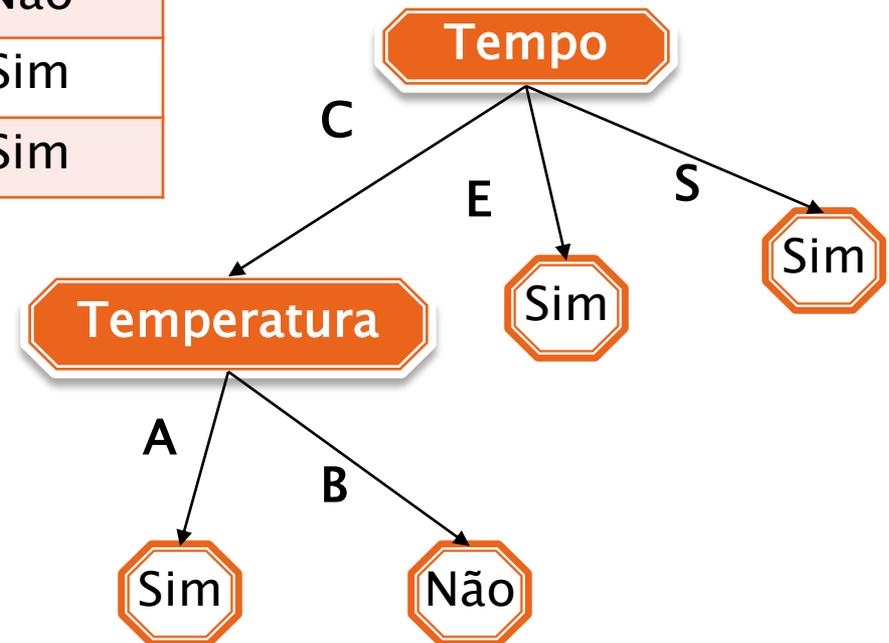
# Tabela / Árvore de Decisões Adaptativas



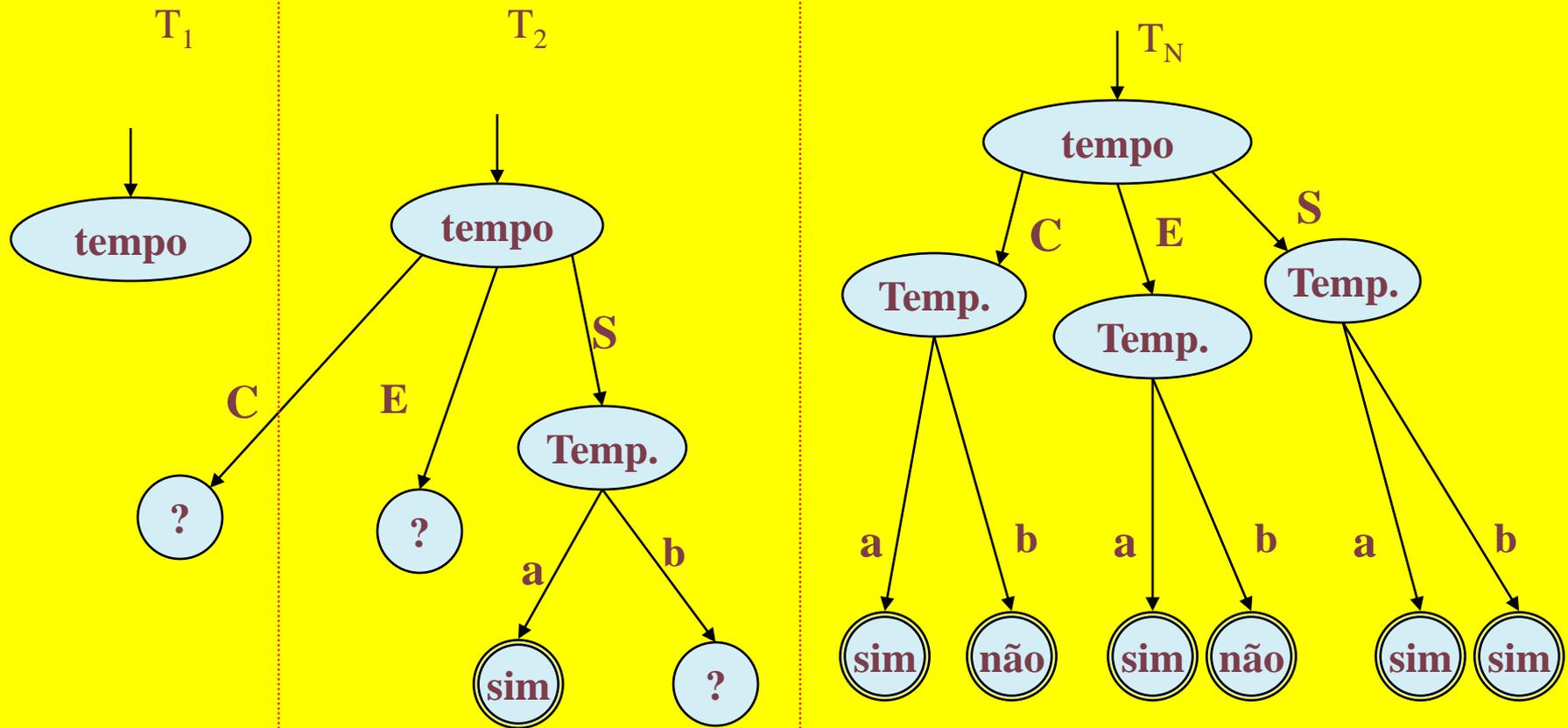
# Inferência de árvores de decisão

Alguns algoritmos disponíveis: ID3, C4.5, ID5, ITI

Tempo	Temperatura	Passear
Ensolarado	Alta	Sim
Ensolarado	Baixa	Sim
Chuvoso	Baixa	Não
Chuvoso	Alta	Sim
Encoberto	Alta	Sim



# AdapTree (Simbólico: Autômato finito adaptativo)



Entradas: Sa-sim, Sa, Sb-Sim, Cb, Cb-não, Cb, Ca-sim, Ea- sim.

Tempo: C=Chuvoso, E=Encoberto, S=Sol.

Temperatura: a=alta, b=baixa.

Decisão: sim: sair; não: não sair; ?: indeterminado.

# Ações Adaptativas Dinâmicas

É possível criar um formalismo em que seja permitido alterar e mesmo criar dinamicamente funções adaptativas novas

Assim, por decisão do autômato, novas funções podem ser introduzidas, em resposta a informações captadas pelo autômato

Mudar o comportamento de uma função adaptativa pode permitir o uso de dispositivos adaptativos na modelagem formal de processos criativos e de aprendizagem

Esta área é muito fértil para a elaboração de novas pesquisas e investigações na área de sistemas inteligentes adaptativos

Desafio: sugerir uma extensão de algum formalismo adaptativo para que passe a permitir a definição e utilização de funções e ações adaptativas dinâmicas.

# Representação de conhecimento

Dispositivos adaptativos, que possuem a capacidade de reter em suas regras informações sobre suas entradas, podem acumular tais informações e utilizá-las quando e como for necessário

Seguindo uma convenção adequada, passam a servir de substrato para a representação do conhecimento coletado

O uso de procedimentos de busca e de edição permitem consultar, avaliar e atualizar o conhecimento armazenado

A partir desse conhecimento contido no dispositivo, é possível inferir novas regras de interpretação

As novas regras podem ser materializadas na forma de novas ações adaptativas caso o dispositivo tenha a capacidade de modificar suas próprias ações adaptativas dinamicamente

# Inferência

ADAPTATIVAS

JAGENS

LABORATÓ

Autômatos adaptativos podem ser utilizados para o aprendizado de linguagens a partir de um conjunto de amostras da mesma



Amostras positivas permitem criar um autômato aceitador de uma linguagem que contenha as sentenças da amostra utilizada



Amostras negativas permitem acrescentar a tal autômato restrições que rejeitem as sentenças contidas nesta amostra



Algoritmos adicionais podem fundir as duas árvores para obter o autômato inferido



Tais alterações podem para isso construir árvores de prefixos e de sufixos da linguagem finita fornecida pelas amostras



Regras adicionais de generalização ou de particularização permitem aproximar melhor o autômato construído do desejado



# Impactos na Ciência da Computação

Formalismo universal, não convencional (potência máxima)

Modelo de computação

Fundamento para novo paradigma e para a elaboração de ferramentas

Viabiliza a geração automática de dispositivos adaptativos

Reduz a fenômenos sintáticos todos os aspectos lingüísticos usuais

Nova metodologia de especificação e implementação de linguagens

Uniformiza a representação de todas as classes de linguagens

Unifica a descrição de todos os aspectos das linguagens

Reduz a descrição de linguagens, mesmo extensíveis, a um só nível

Não introduz formulações novas, mas respeita e estende as usuais

Excelente desempenho potencial:  $O(n)$  nos casos de interesse

Isola o mecanismo de adaptação do formalismo clássico subjacente

Apresenta, em consequência, um elevado potencial prático e didático

# Impactos na Inteligência Artificial

Mecanismos adaptativos mecanizam o processo de aprendizagem

Têm o potencial de variar dinamicamente as ações adaptativas

Permite inferir, analisar, representar e armazenar conhecimento

Permite automatizar processos dinâmicos de tomada de decisões

São adequados para o processamento de linguagens naturais

- montagem de dicionários      à decomposição de palavras
- classificação de palavras      à flexão de palavras
- análise sintática      à concordância e regência

Como transdutores, permitem executar muitas tarefas úteis:

- codificação e decodificação      à criptografia
- compactação de dados      à tradução e transcodificação
- busca semântica      à protocolos e segurança de acesso
- interface homem-máquina      à string-matching e seqüenciação

# Impactos na Eng. de Computação

Modelos adaptativos são dinâmicos, em contraste com os tradicionais

Suas características proporcionam alto desempenho a baixo custo

Estendem dispositivos familiares, reduzindo o treinamento necessário

O mecanismo adaptativo é universal: independe do modelo subjacente

Implantá-lo consiste em acoplá-lo ao dispositivo subjacente disponível

Statecharts adaptativos (sincronizados): para sistemas reativos, sistemas embutidos, sistemas de tempo real, controle de processos

Em versões estatísticas: arte, entretenimento e ensino por computador

Autômatos adaptativos: sistemas seqüenciais, linguagens, protocolos, software básico, interfaces homem-máquina, representação de conhecimento

Gramáticas adaptativas: compiladores, protocolos, linguagens naturais

Tabelas de decisão adaptativas: poderosa ferramenta para Engenharia de Software, sistemas de apoio à tomada de decisão

# Impactos em Compiladores e Linguagens

Desmistificação da área: generalização e unificação das notações

Automatização da obtenção de reconhecedores

Automatização da obtenção de analisadores léxicos e sintáticos

Uniformização do tratamento léxico, sintático e semântico

Redução a um único nível de descrição formal

Uso de uma única infra-estrutura para linguagens de todos os tipos

Ambiente de execução único suporta linguagens extensíveis

Unifica tratamento de linguagens por interpretação ou compilação

Dá suporte a linguagens de paradigma adaptativo

Suporta meta-programação

Utilizável em ambientes de desenvolvimento de aplicações inteligentes

# Perspectivas em Pesquisa

Busca de notações e formulações mais simples, práticas e legíveis

Linguagens de programação aderentes às técnicas adaptativas

Experimentação com formalismos adaptativos ainda não ensaiados

Investigação analítica da complexidade dos formalismos adaptativos

Estabelecimento de formulações minimais equivalentes

Busca de métodos automáticos de produção de dispositivos adaptativos

Comparação formal com modelos não-adaptativos consagrados

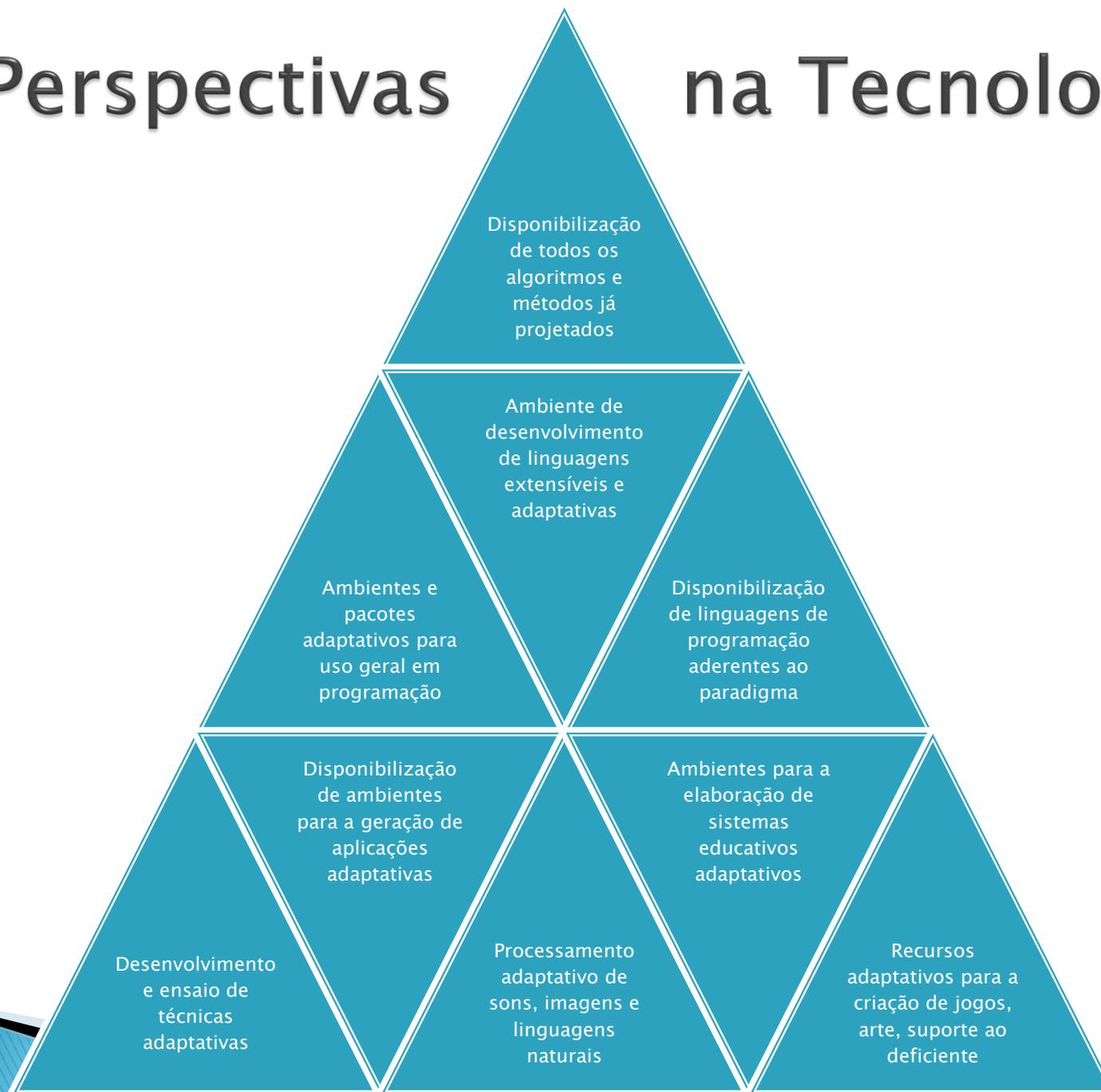
Métricas para a avaliação experimental de dispositivos adaptativos

Busca de métodos de criação automática dos mecanismos adaptativos

Estabelecimento e caracterização formal do novo paradigma

# Perspectivas

# na Tecnologia



# Perspectivas nas Aplicações

Busca de soluções adaptativas eficientes para problemas complexos

Uso da nova tecnologia em nichos de aplicação ainda não explorados

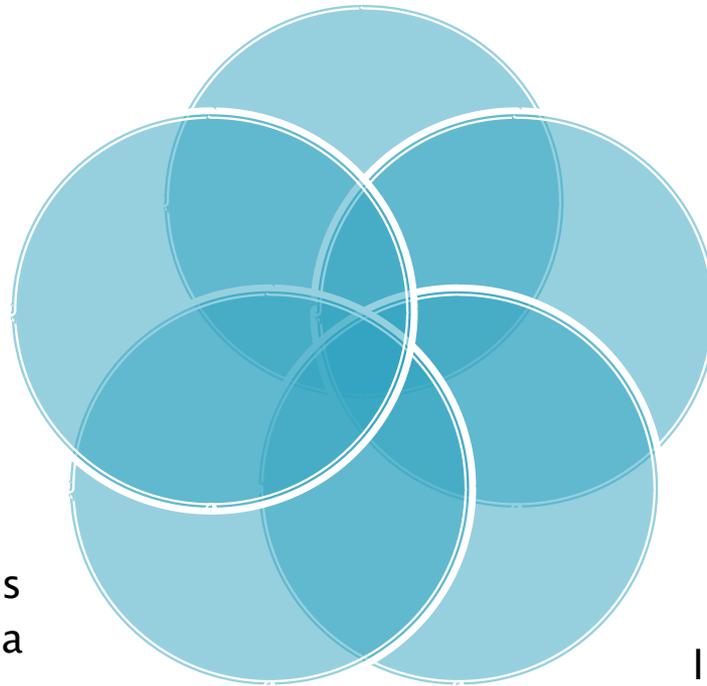
- Aplicações adaptativas a problemas de engenharia em geral
- Aplicações adaptativas a sistemas dinamicamente reconfiguráveis
- Aplicações adaptativas a sistemas fortemente interativos e de realidade virtual
- Aplicações adaptativas a sistemas de ensino e treinamento por computador
- Aplicações adaptativas a sistemas inteligentes autônomos
- Aplicações adaptativas a problemas de processamento de seqüências
- Aplicações adaptativas a sistemas de processamento de voz, imagens e padrões
- Aplicações adaptativas a sistemas inteligentes robóticos, reativos, de tempo real
- Aplicação da tecnologia adaptativa em programas de apoio à decisão
- Aplicações adaptativas em utilitários inteligentes para WEB
- (Busca semântica, roteamento automático inteligente, tradução automática multilíngüe, ...)



# RESULTADOS HISTÓRICOS

desmistificação de  
compiladores,  
autômatos e  
linguagens formais

apesar disso, não  
impõem qualquer  
restrição à solução  
de problemas gerais



uniformização da  
linguagem usada no  
ensino de  
compiladores e  
linguagens

técnicas e métodos  
são voltados para a  
resolução dos  
problemas da prática

métodos e  
ferramentas que  
liberam o usuário de  
aprofundamentos  
teóricos

# RESULTADOS RECENTES



firmou-se um lastro conceitual muito sólido para tratar problemas complexos



dominou-se a geração automática de programas a partir dos formalismos



disponibilizaram-se ferramentas para o ensaio de dispositivos adaptativos



as notações e os níveis de descrição de linguagens foram unificadas



estabeleceram-se os formalismos adaptativos como modelo de computação



está sendo promovido o estabelecimento do novo paradigma, adaptativo

# Obrigado

João José Neto  
Escola Politécnica da USP  
[joao.jose@poli.usp.br](mailto:joao.jose@poli.usp.br)













# Modelos Baseados em Mecanismos Adaptativos e suas Aplicações

Formalismo Adaptativo Geral

João José Neto e  
Ricardo Luis de Azevedo da Rocha



LABORATÓRIO DE LINGUAGENS E TÉCNICAS ADAPTATIVAS

# Agenda

1. Dispositivos não-adaptativos

2. Dispositivos adaptativos

- conceitos e propriedades
- contribuições científicas
- contribuições tecnológicas
- aplicações na Eng.de Computação

3. Exemplo ilustrativo

4. Formalismos Adaptativos

5. Impactos

- na Ciência da Computação
- na Inteligência Artificial
- na Engenharia de Computação
- em Linguagens e Compiladores

6. Perspectivas em pesquisa

7. Resultados

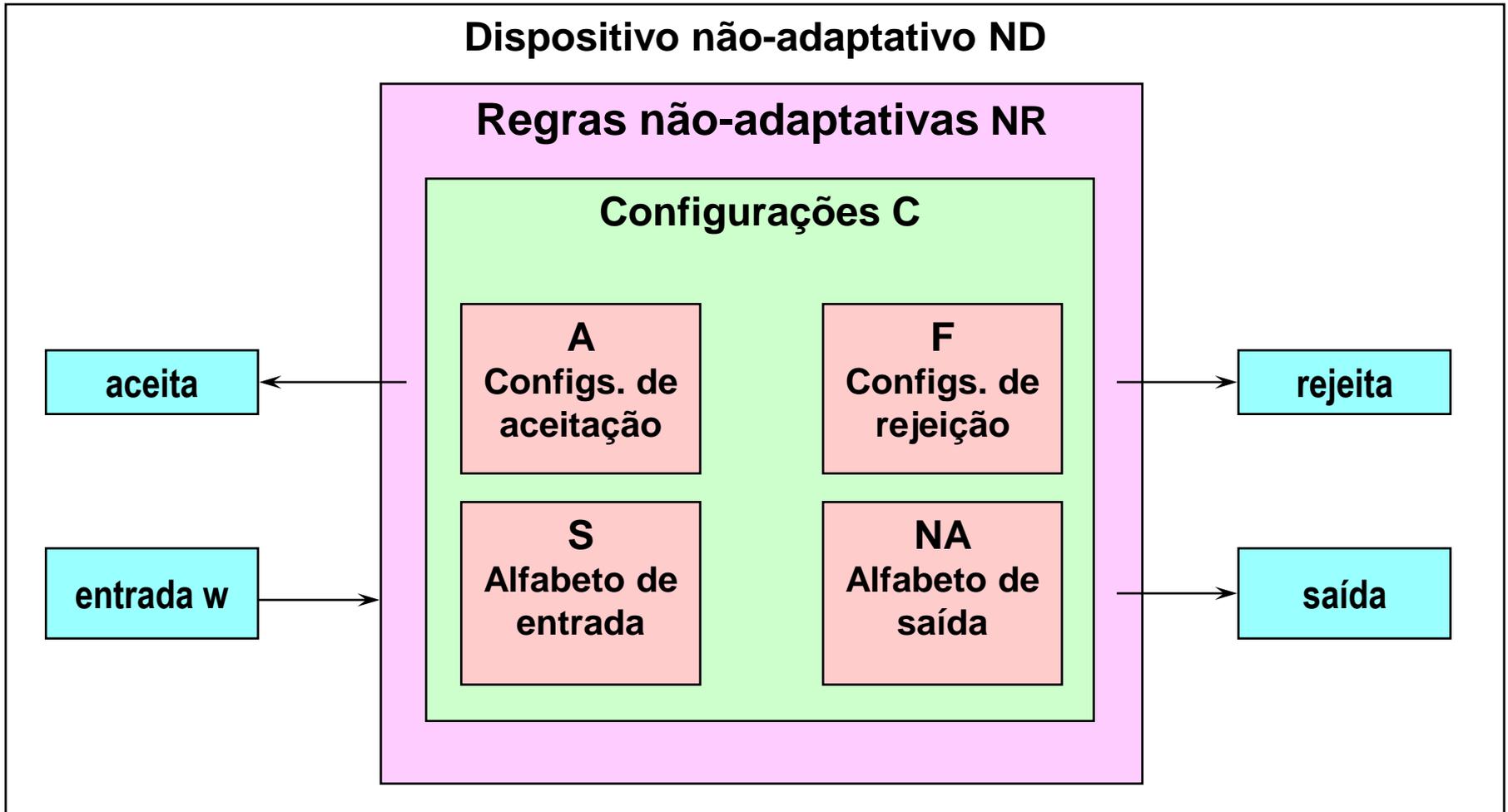


# 1. Dispositivos não-adaptativos (1)

**ND = (C, NR, S, c<sub>0</sub>, A, NA)**      dispositivo não adaptativo

- **C**                      conjunto de configurações possíveis de **ND**
- **c<sub>0</sub> ∈ C**              configuração inicial de **ND**
- **S**                      conjunto de eventos (estímulos de entrada) **ND**,  $\varepsilon \in S$
- **A ⊆ C**                subconjunto de configurações de aceitação
- **F ⊆ C - A**            subconjunto de configurações de rejeição
- **W = w<sub>1</sub>...w<sub>n</sub>**      é uma seqüência de estímulos de entrada,  
 $w_k \in S - \{\varepsilon\}$ ,  $k=1, \dots, n$ , **com  $n \geq 0$**
- **NA**                    conjunto de símbolos de saída,  $\varepsilon \in NA$
- **NR**                    conjunto das regras que definem **ND**

# 1. Dispositivos não-adaptativos (2)



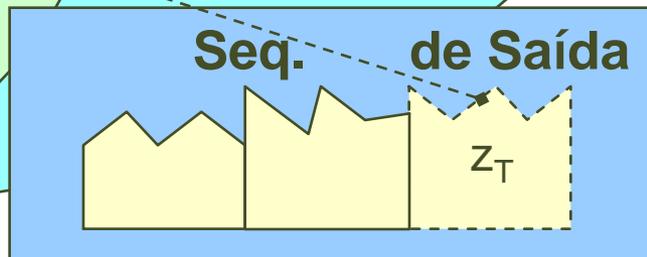
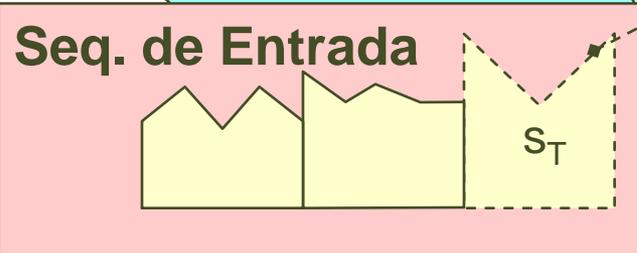
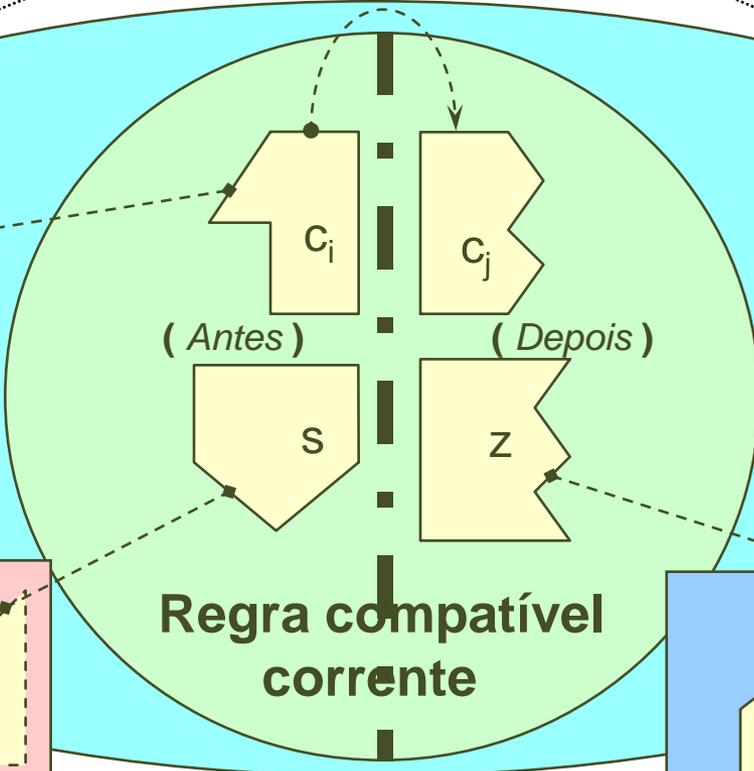
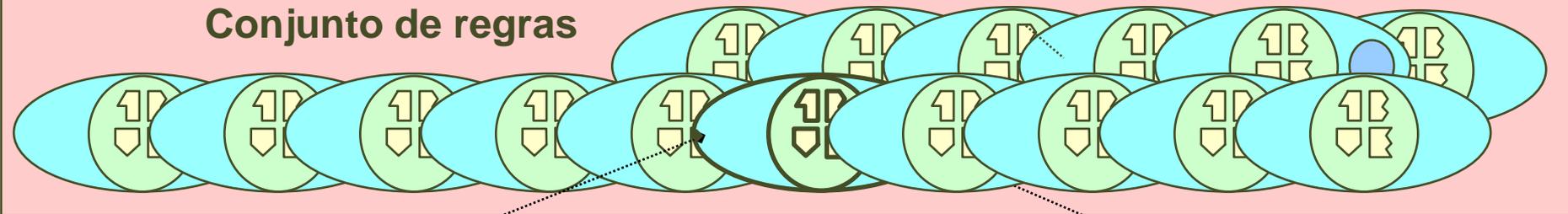
# 1. Dispositivos não-adaptativos (3)

- $\mathbf{NR} \subseteq \mathbf{C} \times \mathbf{S} \times \mathbf{C} \times \mathbf{NA}$

as regras em **NR** convertem a configuração corrente e o símbolo de entrada corrente em uma nova configuração e um símbolo de saída

- Regras  $\mathbf{r} \in \mathbf{NR}$  têm a forma  $\mathbf{r} = (\mathbf{c}_i, \mathbf{s}, \mathbf{c}_j, \mathbf{z})$ 
  - $\mathbf{c}_i$  configuração corrente
  - $\mathbf{s}$  símbolo de entrada corrente
  - $\mathbf{c}_j$  próxima configuração
  - $\mathbf{z}$  símbolo de saída
- Em resposta a qualquer símbolo de entrada  $\sigma \in \mathbf{S}$ , a regra  $\mathbf{r}$  altera a configuração corrente  $\mathbf{c}_i$  para  $\mathbf{c}_j$ , consome  $\mathbf{s}$  e gera como efeito colateral o símbolo de saída  $\mathbf{z} \in \mathbf{NA}$ .

# Conjunto de regras



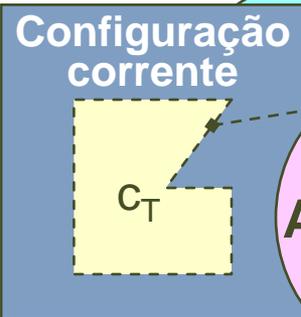
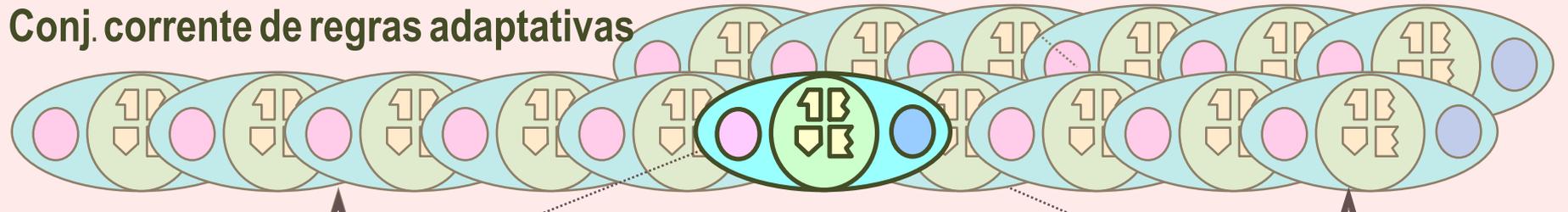
# 1.1 Operação não-adaptativa (1)

- Regra  $r=(c_i, s, c_j, z)$ , com  $r \in NR$ ;  $c_i, c_j \in C$ ;  $s \in S$ ;  $z \in NA$
- Uma regra é compatível com uma configuração  $c$  se e só se  $c_i=c$  e  $s$  for vazia ou então igual ao estímulo corrente de entrada do dispositivo.
- A aplicação de uma única regra compatível leva o dispositivo à configuração  $c_j$  (denotada por  $c_i \Rightarrow^s c_j$ ) e anexa  $z$  ao final da sua seqüência de saída.
- Note-se que  $s$  ou  $z$  ou ambos podem ser vazios.
- $c_i \Rightarrow^{\sim} c_m$ ,  $m \geq 0$  denota  $c_i \Rightarrow c_1 \Rightarrow c_2 \Rightarrow \dots \Rightarrow c_m$ , uma seqüência opcional de movimentos em vazio.
- $c_i \Rightarrow^{\sim w_k} c_j$ , denota  $c_i \Rightarrow^{\sim} c_m \Rightarrow^{w_k} c_j$ , uma seqüência opcional de movimentos em vazio seguida por um movimento que resulte no consumo de algum símbolo  $w_k$ .

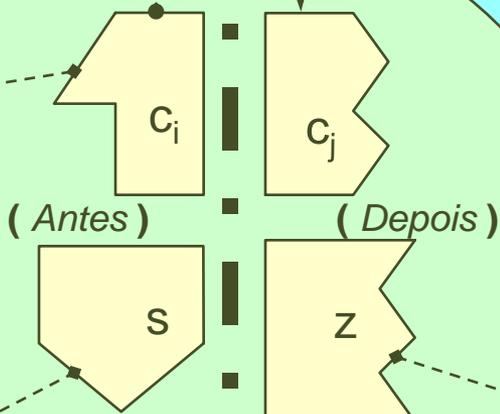
# 1.1 Operação não-adaptativa (2)

- Uma seqüência de entrada  $\mathbf{w} = w_1 w_2 \dots w_n$  é *aceita* por **ND** quando  $\mathbf{c}_0 \Rightarrow^{\sim \mathbf{w} \mathbf{k}} \mathbf{c}_1 \Rightarrow^{\sim \mathbf{w} \mathbf{k}} \mathbf{c}_2 \dots \Rightarrow^{\sim \mathbf{w} \mathbf{k}} \mathbf{c}_n \Rightarrow^{\sim} \mathbf{c}$  (abreviadamente,  $\mathbf{c}_0 \Rightarrow^{\mathbf{w}} \mathbf{c}$ , com  $\mathbf{c} \in \mathbf{A}$ ). Caso contrário, diz-se que **ND** *rejeita*  $\mathbf{w}$ .
- Uma seqüência de saída  $\mathbf{y} \in \mathbf{NA}^*$  é *gerada* por **ND** como resposta à aceitação de  $\mathbf{w}$  sempre que  $\mathbf{y}$  for a seqüência de símbolos de saída associados a cada um dos movimentos de **ND**.
- A *linguagem de entrada* descrita por **ND** é o conjunto  $\mathbf{L}_S(\mathbf{ND}) = \{\mathbf{w} \in \mathbf{S}^* \mid \mathbf{c}_0 \Rightarrow^{\mathbf{w}} \mathbf{c}, \mathbf{c} \in \mathbf{A}\}$  de todas as seqüências  $\mathbf{w} \in \mathbf{S}^*$  aceitas por **ND**.
- A *linguagem de saída* descrita por **ND** é o conjunto  $\mathbf{L}_{\mathbf{NA}}(\mathbf{ND}) = \{\mathbf{y} \in \mathbf{NA}^* \mid \mathbf{c}_0 \Rightarrow^{\mathbf{w}} \mathbf{c}, \mathbf{c} \in \mathbf{A}\}$  de todas as seqüências  $\mathbf{y} \in \mathbf{NA}^*$  que possam ser geradas por **ND** em resposta à aceitação de seqüências de entrada  $\mathbf{w}$ .

Conj. corrente de regras adaptativas



Ação Adaptativa Anterior



Regra compatível corrente

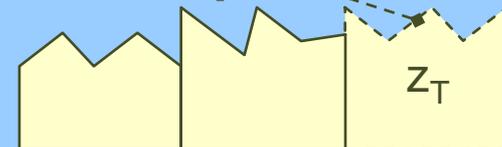
Ação Adaptativa Posterior

Seq. de Entrada



**ADAPTATIVO**

Seq. de Saída



## 2. Dispositivos adaptativos (1)

- **AD=(ND<sub>0</sub>,AM)** adaptativo: para todos os passos de operação **k**≥0, **AD** opera segundo **ND<sub>k</sub>** até que uma ação adaptativa altere o seu conjunto de regras, iniciando o passo **k+1**
- Em cada passo **k**≥0 da operação de **AD**, sendo **ND<sub>k</sub>** seu dispositivo subjacente definido por **NR<sub>k</sub>**, a execução de alguma ação adaptativa não-nula evolve **ND<sub>k</sub>** para **ND<sub>k+1</sub>**.
- Assim, **AD** inicia seu passo de operação **k+1** criando o conjunto **NR<sub>k+1</sub>** como uma versão evoluída de **NR<sub>k</sub>**.
- Subseqüentemente, **AD** seguirá o comportamento de **ND<sub>k+1</sub>** até que uma nova ação adaptativa não-nula inicie um novo passo de operação.
- Esse procedimento é repetido até que toda a seqüência de entrada tenha sido totalmente processada.

## 2. Dispositivos adaptativos (2)

- **AD** inicia a operação em  $\mathbf{c}_0$ , na forma seguinte:

$$\mathbf{AD}_0 = (\mathbf{C}_0, \mathbf{AR}_0, \mathbf{S}, \mathbf{c}_0, \mathbf{A}, \mathbf{NA}, \mathbf{BA}, \mathbf{AA}).$$

- No passo  $\mathbf{k} \geq 0$ , um estímulo de entrada (sempre) leva **AD** a alguma configuração, iniciando então um novo passo  $\mathbf{k}+1$  se e só se uma ação adaptativa não-nula tiver sido executada.

- Então, estando **AD** no passo  $\mathbf{k}$ , e na forma:

$$\mathbf{AD}_k = (\mathbf{C}_k, \mathbf{AR}_k, \mathbf{S}, \mathbf{c}_k, \mathbf{A}, \mathbf{NA}, \mathbf{BA}, \mathbf{AA}),$$

a execução de uma ação adaptativa não-nula o leva à nova forma:

$$\mathbf{AD}_{k+1} = (\mathbf{C}_{k+1}, \mathbf{AR}_{k+1}, \mathbf{S}, \mathbf{c}_{k+1}, \mathbf{A}, \mathbf{NA}, \mathbf{BA}, \mathbf{AA}).$$

## 2. Dispositivos adaptativos (3)

$\varepsilon$  (“vazio”) denota a ausência de qualquer outro elemento válido do conjunto correspondente.

- $\mathbf{w} = \mathbf{w}_1 \mathbf{w}_2 \dots \mathbf{w}_n$ ,  $\mathbf{k} = 1, \dots, \mathbf{n}$  é uma seqüência de estímulos de entrada, onde  $\mathbf{w}_k \in \mathbf{S} - \{\varepsilon\}$ .
- $\mathbf{AD} = (\mathbf{ND}_0, \mathbf{AM})$  é algum dispositivo adaptativo, dado por um dispositivo subjacente inicial  $\mathbf{ND}_0$  e um mecanismo adaptativo  $\mathbf{AM}$ .
- $\mathbf{ND}_k$  é o dispositivo não-adaptativo subjacente de  $\mathbf{AD}$  em algum passo de operação  $\mathbf{k}$ .  $\mathbf{ND}_0$  é o dispositivo subjacente inicial de  $\mathbf{AD}$ , definido por um conjunto  $\mathbf{NR}_0$  de regras não-adaptativas.
- Por definição, qualquer regra não-adaptativa, em qualquer dos  $\mathbf{NR}_k$  espelha a regra adaptativa correspondente em  $\mathbf{AR}_k$ .
- $\mathbf{C}_k$  é o conjunto de todas as configurações possíveis para  $\mathbf{ND}$  no passo  $\mathbf{k}$ , e  $\mathbf{c}_k \in \mathbf{C}_k$  é sua configuração inicial no passo  $\mathbf{k}$ .
- Para  $\mathbf{k} = 0$ , tem-se  $\mathbf{C}_0$ , o conjunto inicial de configurações válidas, e  $\mathbf{c}_0 \in \mathbf{C}_0$ , a configuração inicial comum para  $\mathbf{ND}_0$  e  $\mathbf{ND}$ .

## 2. Dispositivos adaptativos (4)

- **S** é o conjunto (fixo, finito) de todos os possíveis eventos que são estímulos de entrada válidos para **AD** ( $\varepsilon \in \mathbf{S}$ ).
- **A**  $\subseteq$  **C** (respectivamente **F** = **C** - **A**) é o seu subconjunto de configurações de aceitação (respectivamente rejeição).
- **BA** e **AA** são conjuntos de ações adaptativas, ambos contendo a ação adaptativa nula ( $\varepsilon \in \mathbf{BA} \cap \mathbf{AA}$ )
- **NA**, com  $\varepsilon \in \mathbf{NA}$ , é um conjunto (finito, fixo) de todos os símbolos de saída de **AD**, a serem gerados como efeitos colaterais da aplicação das ações adaptativas.
- Como no caso dos dispositivos não-adaptativos, a seqüência de saída pode ser interpretada como uma seqüência correspondente de chamadas de procedimentos.

## 2. Dispositivos adaptativos (5)

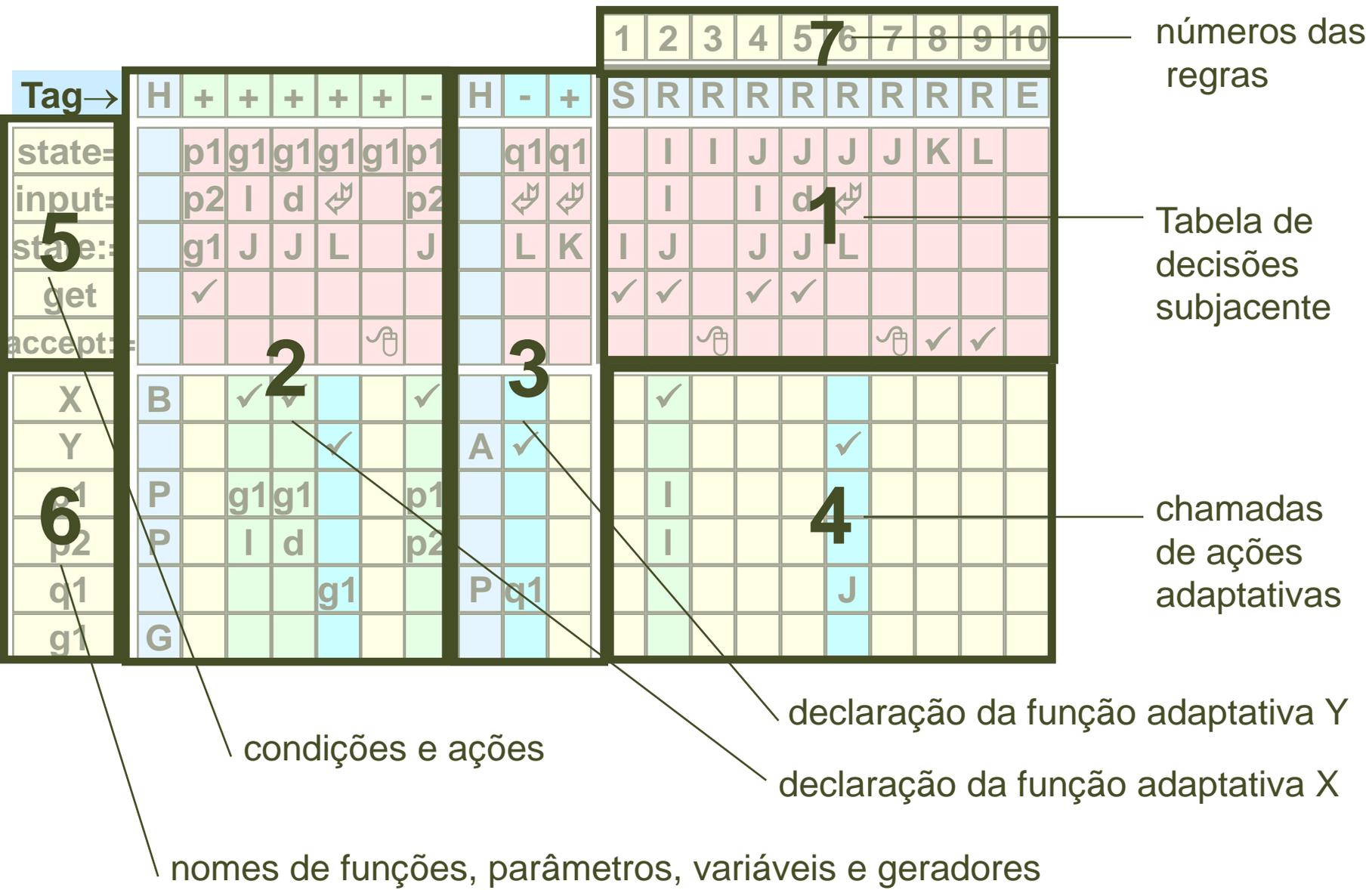
- $\mathbf{AR}_k$  é um conjunto de regras adaptativas, dadas pela relação  
$$\mathbf{AR}_k \subseteq \mathbf{BA} \times \mathbf{C} \times \mathbf{S} \times \mathbf{C} \times \mathbf{NA} \times \mathbf{AA}.$$
- $\mathbf{AR}_0$  define o comportamento inicial de  $\mathbf{AD}$ .
- Ações adaptativas mapeiam o conjunto corrente de regras adaptativas  $\mathbf{AR}_k$  de  $\mathbf{AD}$  em um novo conjunto  $\mathbf{AR}_{k+1}$  pela adição e/ou eliminação de regras a  $\mathbf{AR}_k$ .
- Regras  $\mathbf{ar} \in \mathbf{AR}_k$  têm a forma  $\mathbf{ar} = (\mathbf{ba}, \mathbf{c}_i, \mathbf{s}, \mathbf{c}_j, \mathbf{z}, \mathbf{aa})$ , significando que, em resposta a algum estímulo  $\mathbf{s} \in \mathbf{S}$  de entrada,  $\mathbf{ar}$  executa inicialmente a ação adaptativa  $\mathbf{ba} \in \mathbf{BA}$ ;
- Se a execução de  $\mathbf{ba}$  eliminar  $\mathbf{ar}$  de  $\mathbf{AR}_k$ , então a execução de  $\mathbf{ar}$  será abortada;
- caso contrário, é aplicada a regra adaptativa subjacente  $\mathbf{nr} = (\mathbf{c}_i, \mathbf{s}, \mathbf{c}_j, \mathbf{z}) \in \mathbf{NR}_k$ , conforme descrito anteriormente;
- finalmente, a ação adaptativa  $\mathbf{aa} \in \mathbf{AA}$  é executada.

## 2. Dispositivos adaptativos (6)

- Defina-se **AR** como o conjunto das possíveis regras adaptativas para **AD**.
- Defina-se **NR** como o conjunto de todas as possíveis regras não-adaptativas para **AD**.
- $\mathbf{AM} \subseteq \mathbf{BA} \times \mathbf{NR} \times \mathbf{AA}$ , definida para um particular dispositivo adaptativo **AD**, é um *mecanismo adaptativo* a ser aplicado em qualquer passo de operação **k** a cada regra em  $\mathbf{NR}_k \subseteq \mathbf{NR}$ .
- **AM** deve ser tal que opere como uma função quando aplicado a qualquer subdomínio  $\mathbf{NR}_k \subseteq \mathbf{NR}$ .
- Isto determina um único par de ações adaptativas a ser associado a cada regra não-adaptativa.
- O conjunto  $\mathbf{AR}_k \subseteq \mathbf{AR}$  pode ser construído coletando-se todas as regras adaptativas obtidas associando-se tais pares de ações adaptativas às regras não-adaptativas correspondentes em  $\mathbf{NR}_k$ .
- Equivalentemente, pode-se obter  $\mathbf{NR}_k$  eliminando-se todas as referências a ações adaptativas das regras em  $\mathbf{AR}_k$ .

## 3. Exemplo Ilustrativo

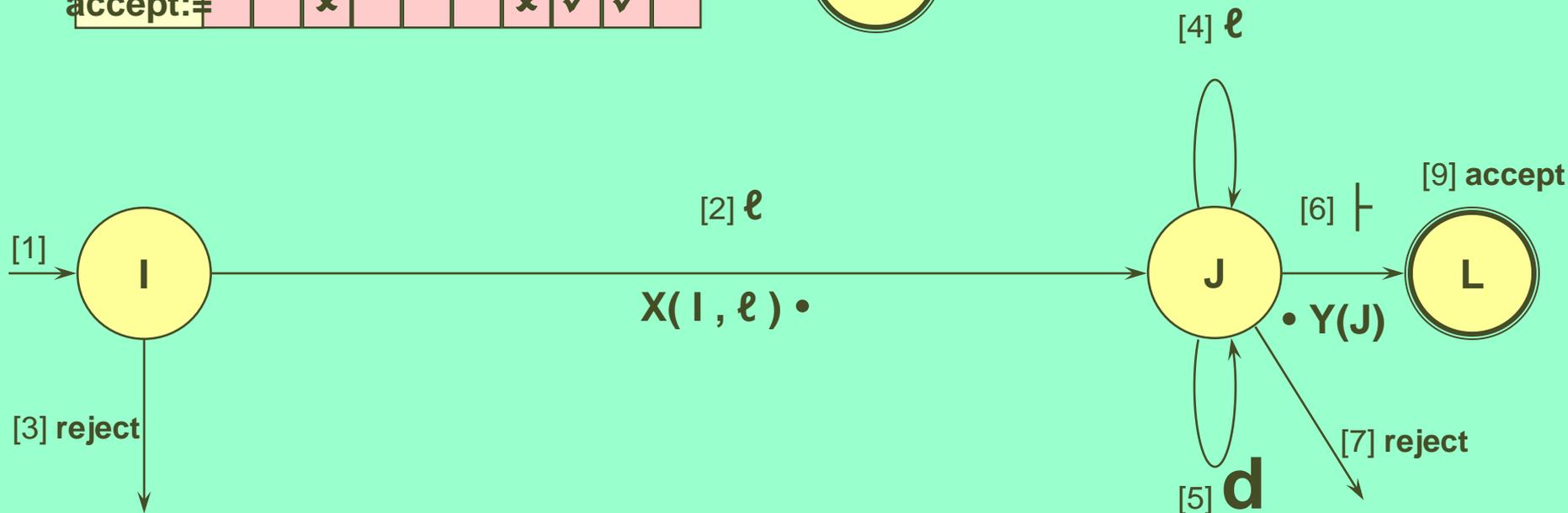
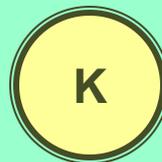
aplicação de tabela de decisão adaptativa para  
representar um autômato adaptativo usado em  
processamento de linguagem natural



# Tabela de decisões inicial

	1	2	3	4	5	6	7	8	9	10
state=		I	I	J	J	J	J	K	L	
input=		ℓ		ℓ	d	ℓ				
state:=	I	J		J	J	L				
get	✓	✓		✓	✓					
accept:=			x				x	✓	✓	

[8] accept



**Topologia inicial do autômato adaptativo e aspecto da sua representação na Tabela de decisões subjacente**

		1	2	3	4	5	6	7	8	9	10
Tag→	H + + + + -	H - +	S	R	R	R	R	R	R	R	E
state=	p1 g1 g1 g1 g1 p1	q1 q1	I	I	J	J	J	J	K	L	
input=	p2 ℓ d † p2	† †	ℓ	ℓ	d	†					
state:=	g1 J J L J	L K	I	J	J	J	L				
get	✓		✓	✓	✓	✓					
accept:=	x		x				x	✓	✓		
X	B ✓ ✓		✓								
Y		A ✓				✓					
p1	P g1 g1		I								
p2	P ℓ d		ℓ								
q1		P q1				J					
g1	G										

$X(i, l)$ :  
 acrescenta  
**11, 12, 13, 14, 25;**  
 remove regra 2

11	12	13	14	25
R	R	R	R	R
I	#1	#1	#1	#1
ℓ	ℓ	d	†	
#1	J	J	L	
✓				
				x
	✓	✓		
			✓	
	#1	#1		
	ℓ	d		
			#1	

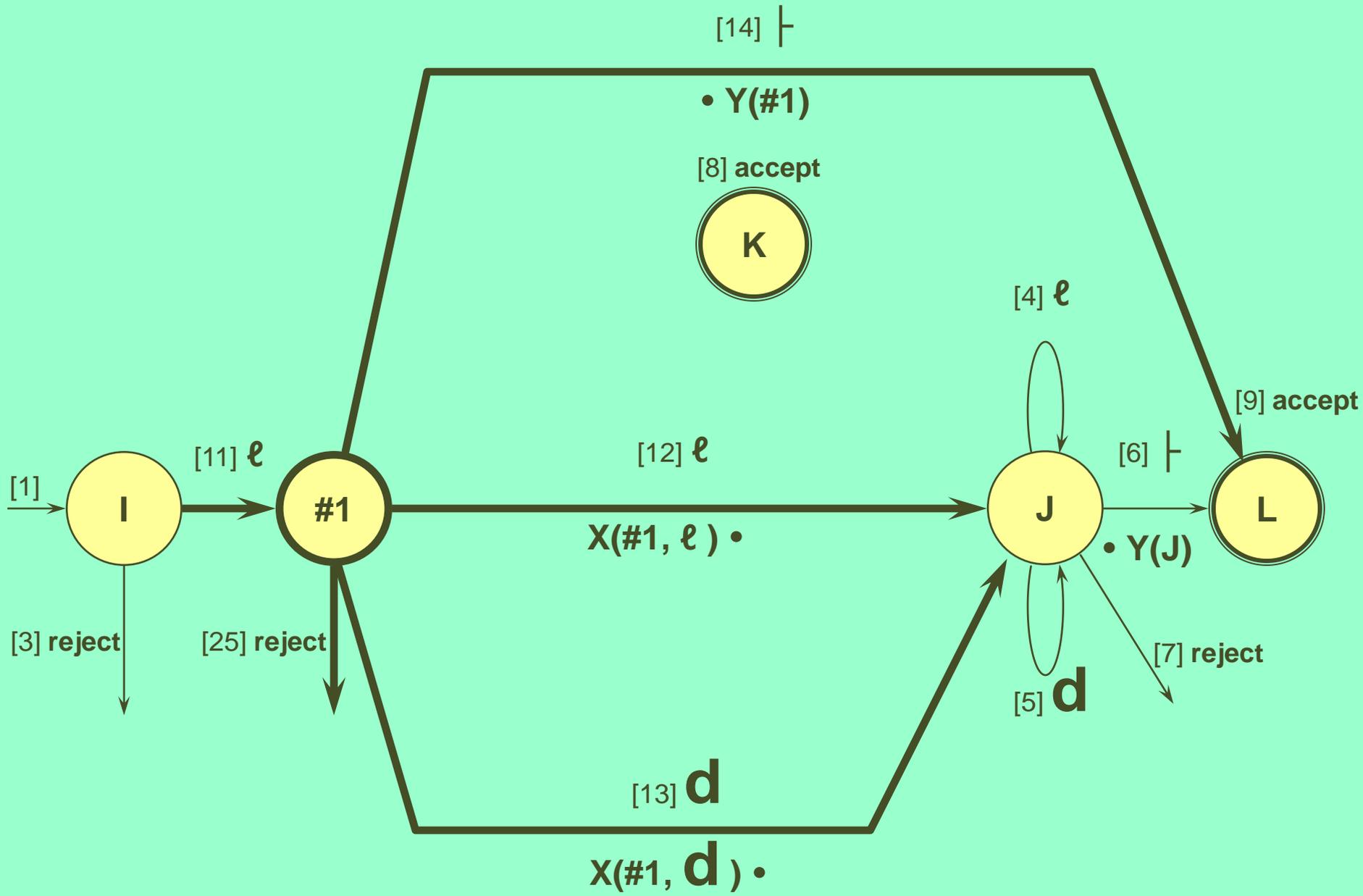
seq. de entrada: ℓ ℓ †

Regras aplicadas:  
regra 1: (inicializa):  
 faz state = I; lê "ℓ"

regra 2: processa "ℓ":

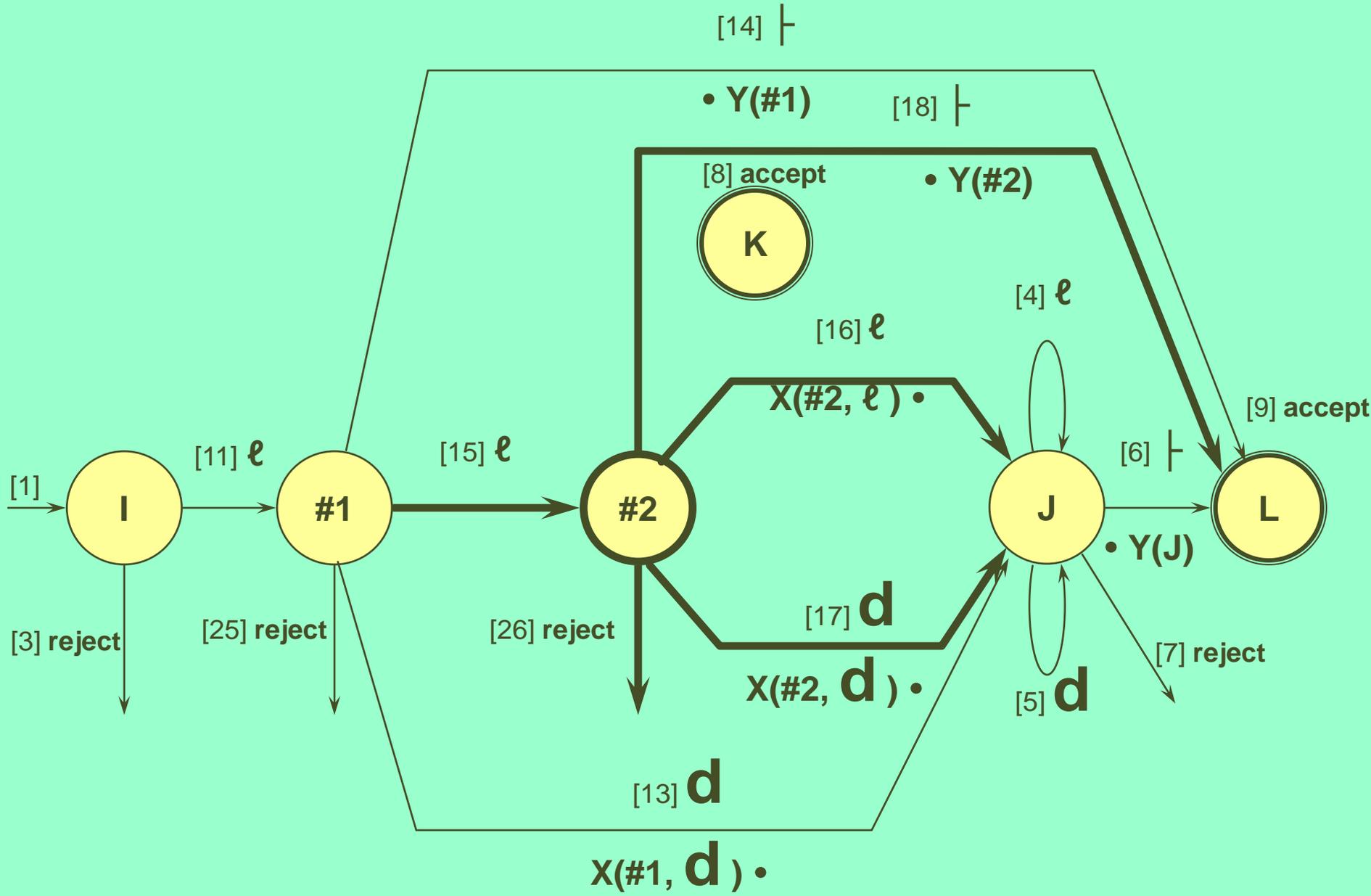
chama  $X(i, \ell)$   
 Note que a regra 2 é  
 uma regra que se  
 auto-elimina, e não  
 termina.

**Processa ℓ ℓ †** : (1) Inicialização; (2) Processa "ℓ"



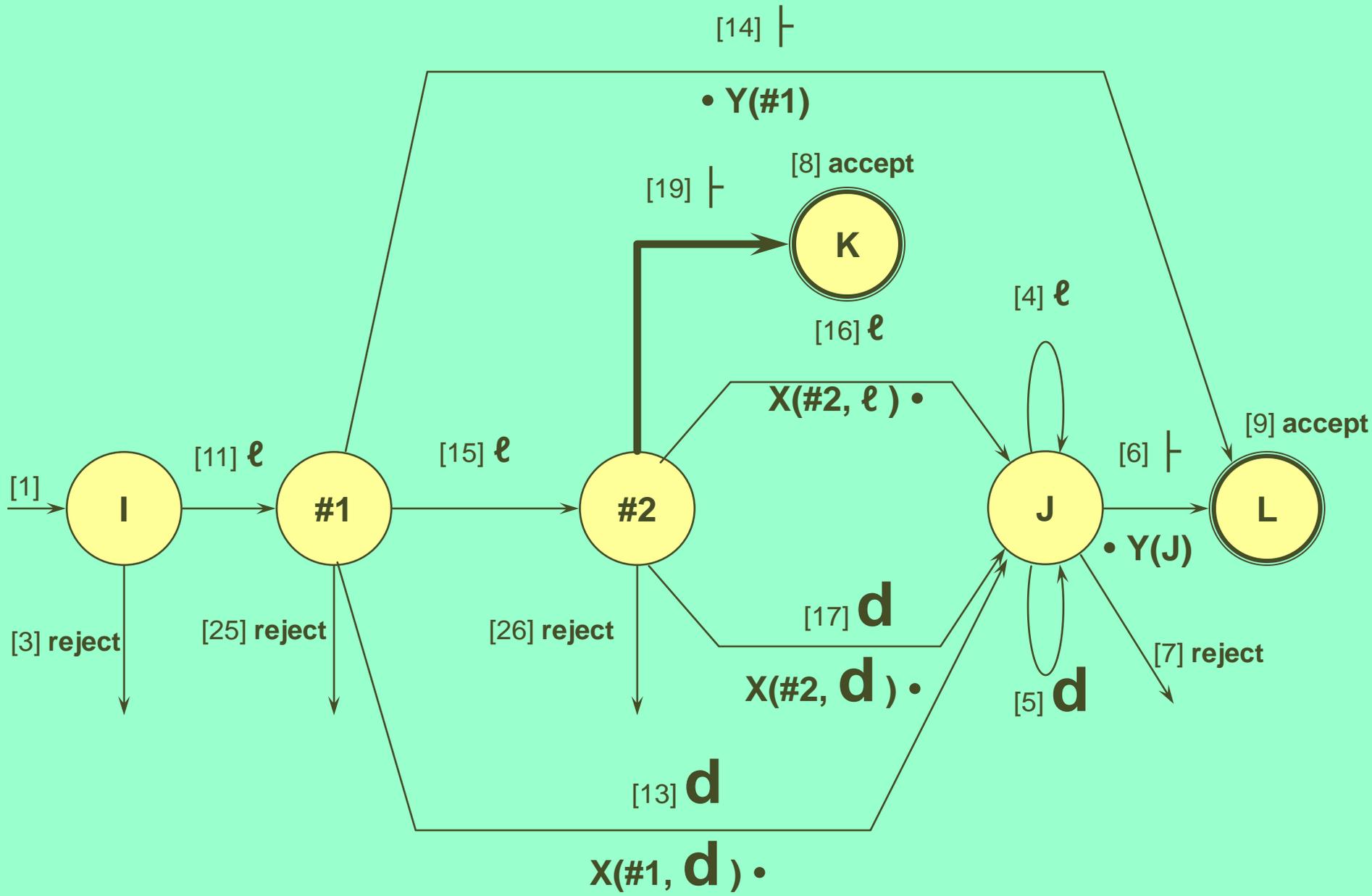
autômato adaptativo depois de executar  $X(I, \ell) \bullet$





autômato adaptativo após executar  $X(\#1, \ell) \cdot$





autômato adaptativo após executar  $\bullet Y(\#2)$

1 11 3 4 5 6 7 8 9 15 13 14 25 16 17 19 26 10

Tag→ H + + + + + - H - +

S R R R R R R R R R R R R R R R R R E

state= p1 g1 g1 g1 g1 p1 q1 q1

I I J J J J K L #1 #1 #1 #1 #2 #2 #2 #2

input= p2 ℓ d † p2 † †

ℓ ℓ d † ℓ d †

state:= g1 J J L J L K

I #1 J J L #2 J L J J K

get ✓

✓ ✓ ✓ ✓ ✓

accept:= x

x x ✓ ✓ x x x

X B ✓ ✓ ✓

✓ ✓ ✓

Y A ✓

✓ ✓

p1 P g1 g1 p1

#1 #2 #2

p2 P ℓ d p2

d ℓ d

q1 g1 P q1

J #1

g1 G

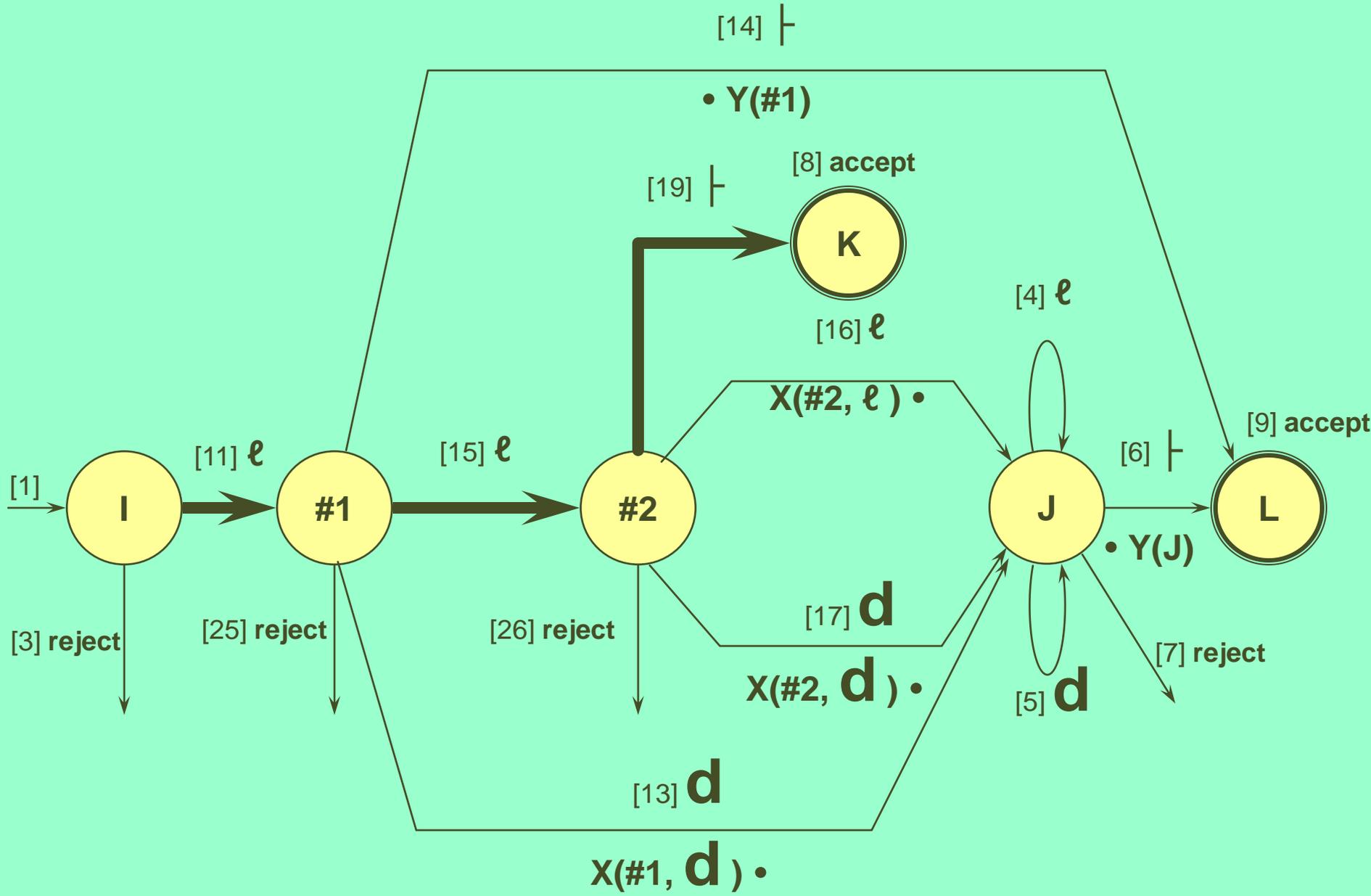
J

state = L; input = †

Regras aplicadas:

regra 9: aceita “ℓ ℓ †”

Processa ℓ ℓ †



autômato adaptativo após aceitar  $\ell \ell \vdash$

1	11	3	4	5	6	7	8	9	15	13	14	25	16	17	19	26	10
Tag→	S	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	E
state=	I	I	J	J	J	J	K	L	#1	#1	#1	#1	#2	#2	#2	#2	
input=	ℓ		ℓ	d	⊥				ℓ	d	⊥		ℓ	d	⊥		
state:=	I	#1	J	J	L				#2	J	L		J	J	K		
get	✓	✓		✓	✓				✓								
accept:=			x				x	✓	✓				x				x
X										✓			✓	✓			
Y					✓						✓						
p1									#1				#2	#2			
p2									d				ℓ	d			
q1					J						#1						
g1																	

X( #1, d ):  
 acrescenta  
 20, 21, 22, 23, 27;  
 remove 13

20	21	22	23	27
R	R	R	R	R
#1	#3	#3	#3	#3
d	ℓ	d	⊥	
#3	J	J	L	
✓				
				☹

	✓	✓		
			✓	
	#3	#3		
	ℓ	d		
			#3	

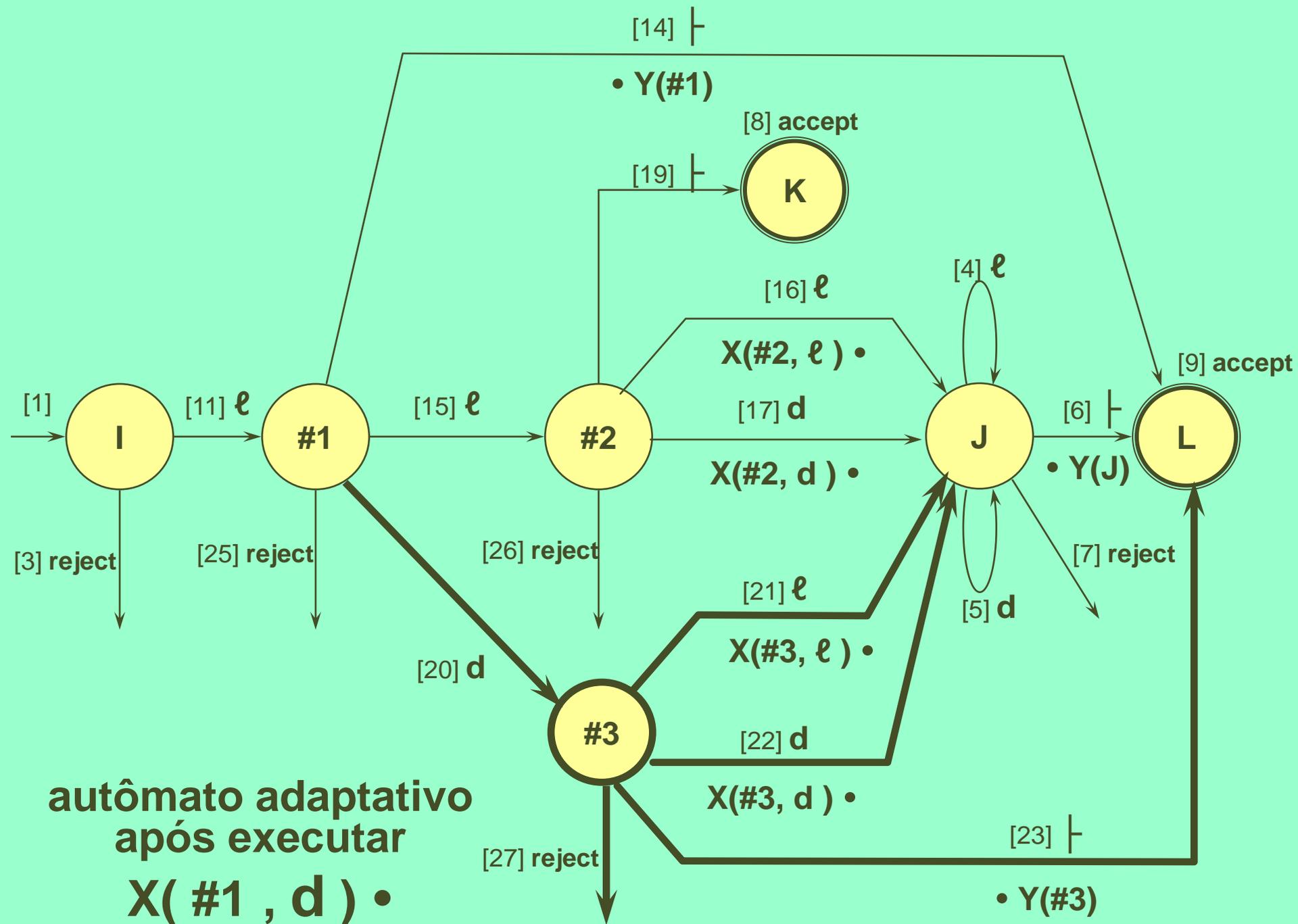
seq. de entrada: ℓ d ⊥

Regras aplicadas:  
regra 1: (inicializa):  
 faz state = I; lê “ℓ”

regra 11: processa “ℓ”:  
 faz state = #1 ; lê “d”

regra 13: processa “d”:  
 chama X( #1, d )

Processa ℓ d ⊥



	1	11	3	4	5	6	7	8	9	15	20	14	25	16	17	19	26	21	22	23	27	10
Tag→	S	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	E
state=		I	I	J	J	J	J	K	L	#1	#1	#1	#1	#2	#2	#2	#2	#3	#3	#3	#3	
input=		ℓ		ℓ	d	└				ℓ	d	└		ℓ	d	└		ℓ	d	└		
state:=	I	#1		J	J	L				#2	#3	L		J	J	K		J	J	L		
get	✓	✓		✓	✓					✓	✓											
accept:=			x				x	✓	✓				x				x					x
X														✓	✓			✓	✓			
Y					✓							✓								✓		
p1														#2	#2			#3	#3			
p2														ℓ	d			ℓ	d			
q1					J						#1									#3		
g1																						

Y( #3 ):  
acrescenta 24;  
remove 23

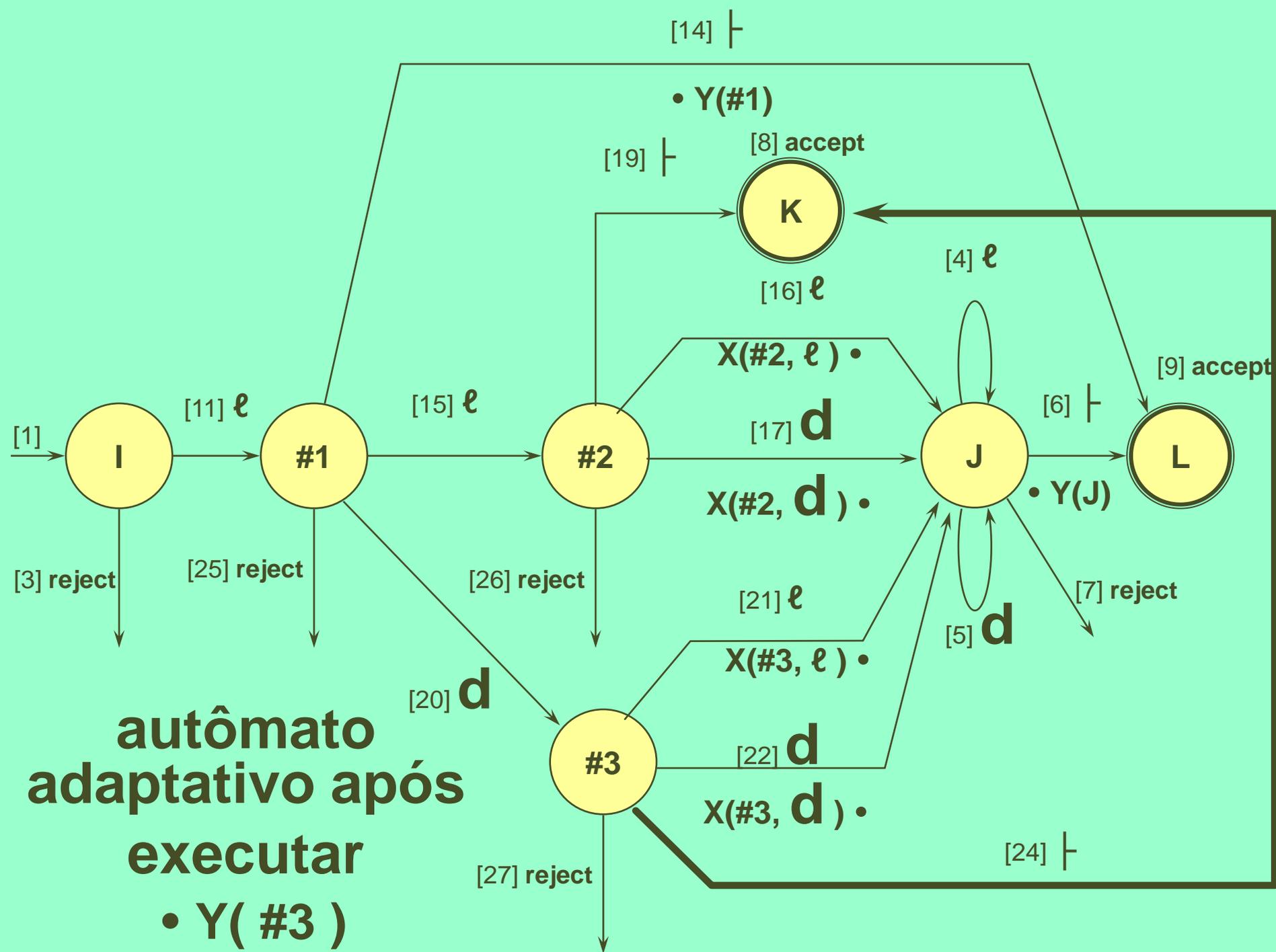
- 24
- R
- #3
- └
- K
- 
- 
- 
- 
- 

state = #1; input = d ; cadeia de entrada: ℓ d └

Regras aplicadas: regra 20: processa “d”: faz state = #3; lê “└”

regra 23: processa“└”: faz state = L; chama Y ( #3 )

Processa ℓ d └

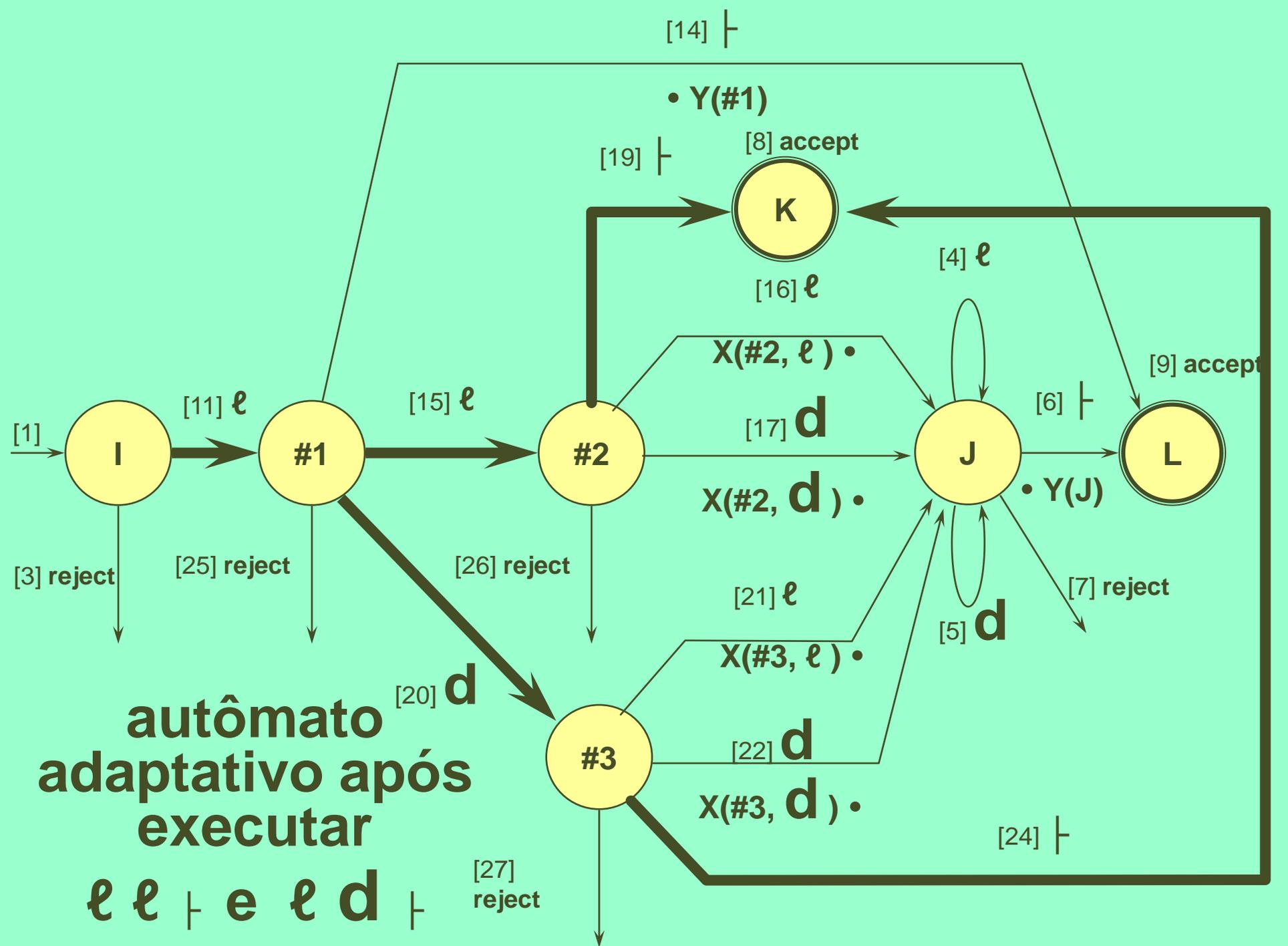


	1	11	3	4	5	6	7	8	9	15	20	14	25	16	17	19	26	21	22	24	27	10
Tag→	S	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	E
state=		I	I	J	J	J	J	K	L	#1	#1	#1	#1	#2	#2	#2	#2	#3	#3	#3	#3	
input=		ℓ		ℓ	d	⊥				ℓ	d	⊥		ℓ	d	⊥		ℓ	d	⊥		
state:=		I	#1		J	J	L				#2	#3	L		J	J	K		J	J	K	
get		✓	✓		✓	✓					✓	✓										
accept:=				x				x	✓	✓				x				x				x
X															✓	✓				✓	✓	
Y						✓						✓										
p1															#2	#2				#3	#3	
p2															ℓ	d				ℓ	d	
q1						J						#1										
g1																						

state = L; input = ⊥ ; cadeia de entrada: ℓ d ⊥

Regras aplicadas: regra 9: aceita “ℓ d ⊥ ”

Processa ℓ d ⊥





**F I M**