



O Autômato Adaptativo como Modelo de Computação e sua Aplicação em Reconhecimento de Padrões*

I WORPEC

**Workshop de Pesquisa em
Engenharia e Computação**

Amaury Antônio de Castro Junior

amaury@ec.ucdb.br

Orientador: Prof. Dr. João José Neto

joao.jose@poli.usp.br

* Durante o desenvolvimento deste trabalho, o autor vem recebendo apoio financeiro da UCDB.

Roteiro

- Visão geral sobre tecnologias adaptativas.
 - Conceitos fundamentais:
 - Autômato finito.
 - Autômato de pilha estruturado.
 - Autômato Adaptativo (com exemplo).
 - Dispositivos adaptativos.
 - Autômato adaptativo como modelo de computação.
 - Aplicação: *Syntactic Pattern Recognition*
-

Tecnologias Adaptativas

- Componente teórica: conceitos e formalismos.
 - Componente tecnológica: aspectos práticos e de aplicação.
 - Componente metodológica: aspectos de modelagem, projeto e implementação (paradigma).
 - Componente pedagógica: expressividade e poder computacional.
-

Conceitos Fundamentais

- Autômato finito:
 - Estados e transições.
 - Função de transição simples.
 - Reconhecimento de linguagens regulares.
 - Autômato com pilha:
 - Estados, transições e pilha.
 - Função de transição “estendida”.
 - Reconhecimento de linguagens livre de contexto.
-

Conceitos Fundamentais

- Máquina de Turing:
 - Estados, transições e “fita”.
 - Função de transição “complexa”.
 - Reconhecimento de linguagens recursivamente enumeráveis.
-

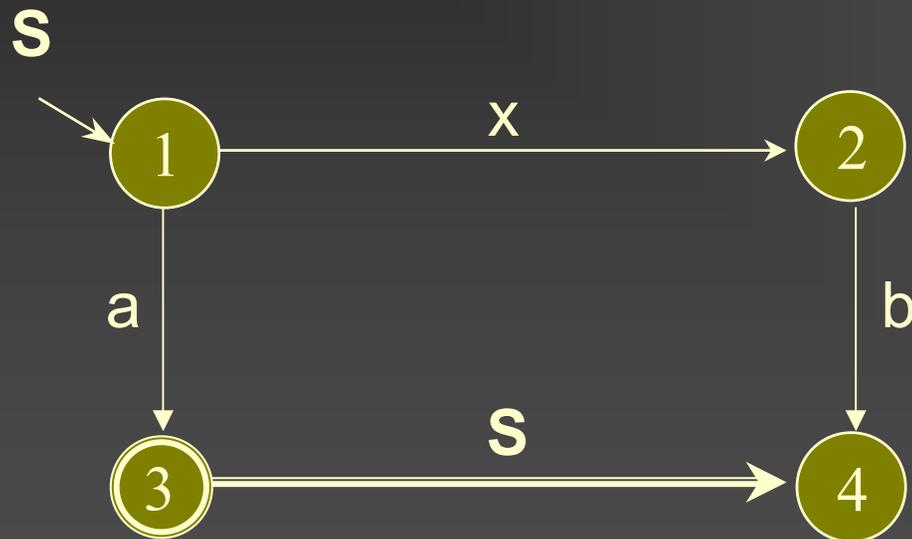
Conceitos Fundamentais

- Autômato de Pilha Estruturado:
 - Estados, transições, pilha e sub-máquinas.
 - “Novidades” em relação ao autômato de pilha:
 - Transições de “chamada”/”retorno” de sub-máquina.
 - Pilha => uso “restrito”.
 - Reconhecimento de linguagens livres de contexto.
-

Conceitos Fundamentais

- Exemplo: APE

- $L = \{ a^n x b^n \mid n \geq 0 \}$



Conceitos Fundamentais

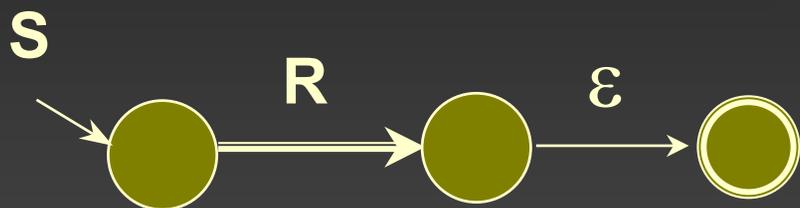
- Autômato Adaptativo:
 - Estados, transições, pilha, sub-máquinas, funções adaptativas e ações adaptativas.
 - “Novidades” em relação ao APE:
 - Chamadas de ações adaptativas.
 - Ações adaptativas elementares: consulta, remoção e inserção.
 - Poder de Máquina de Turing.
-

Conceitos Fundamentais

■ Exemplo 1: AA

$$L = \{ a^n b^n c^n \mid n \geq 0 \}$$

Sub-máquina principal



Ação Adaptativa *A*

Inserir uma transição consumindo o símbolo "c" na sub-máquina S para cada símbolo "b" consumido em R.

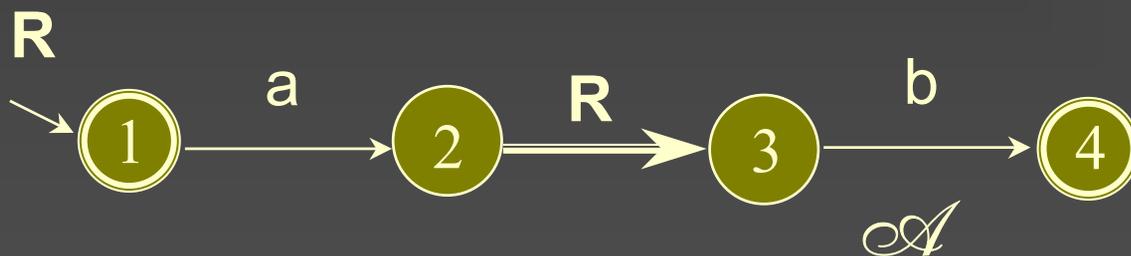
Conceitos Fundamentais

Sentença: aabbcc

- Símbolo corrente: a

Sub-máquina principal

Pilha

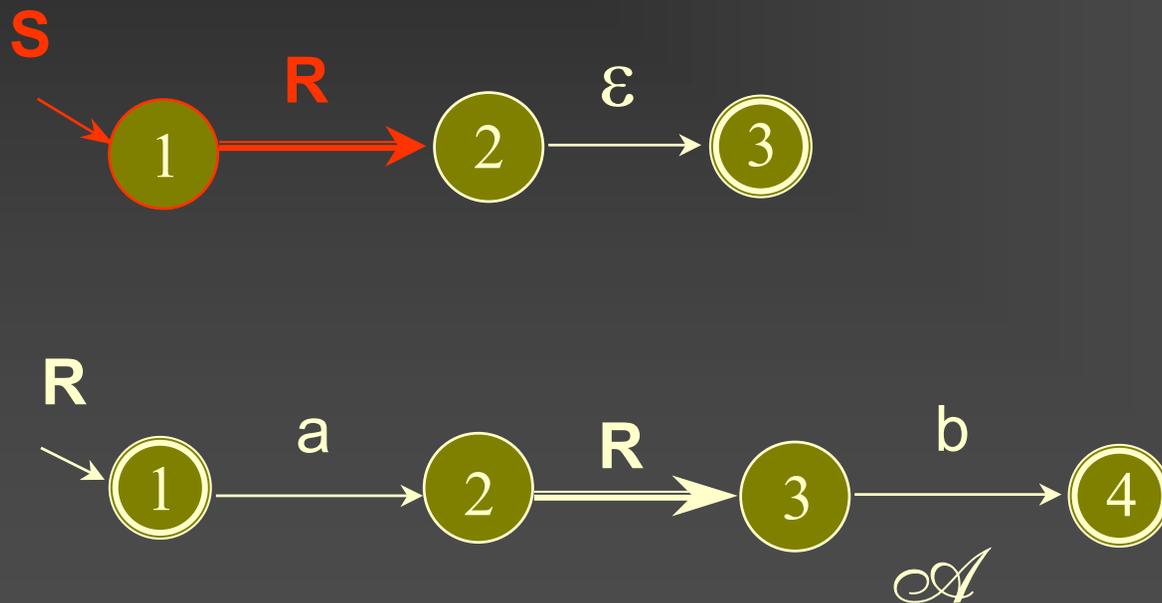


Conceitos Fundamentais

Sentença: aabbcc

- Começa na sub-máquina principal S.
- Chama sub-máquina R e empilha o estado de retorno (2) da máquina S.

Pilha

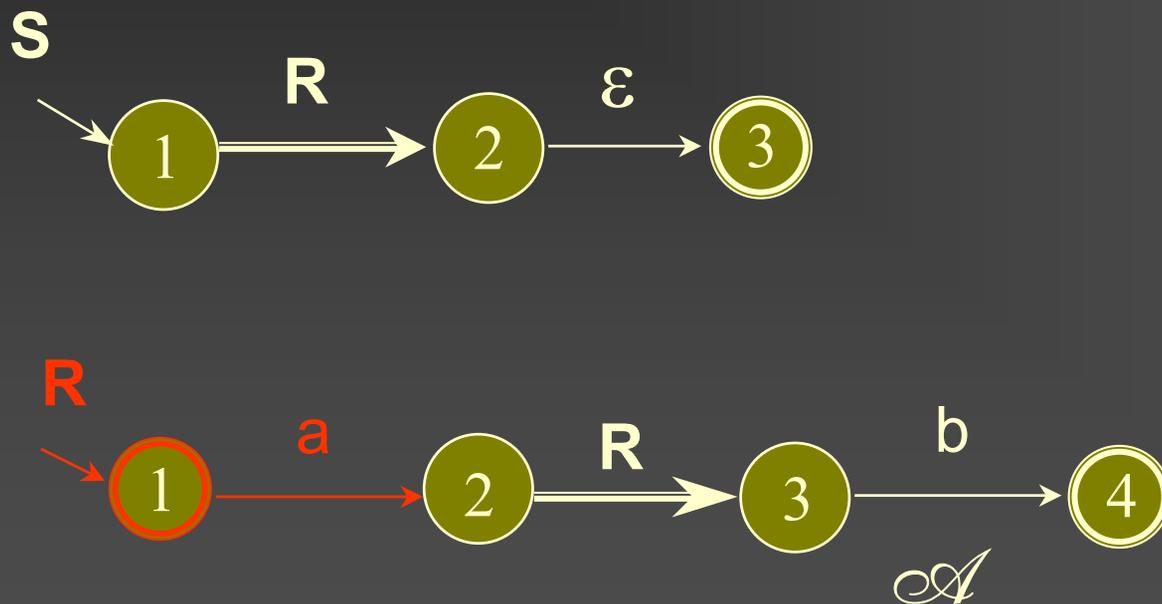


Conceitos Fundamentais

Sentença: abbcc

- Consome símbolo "a".

Pilha

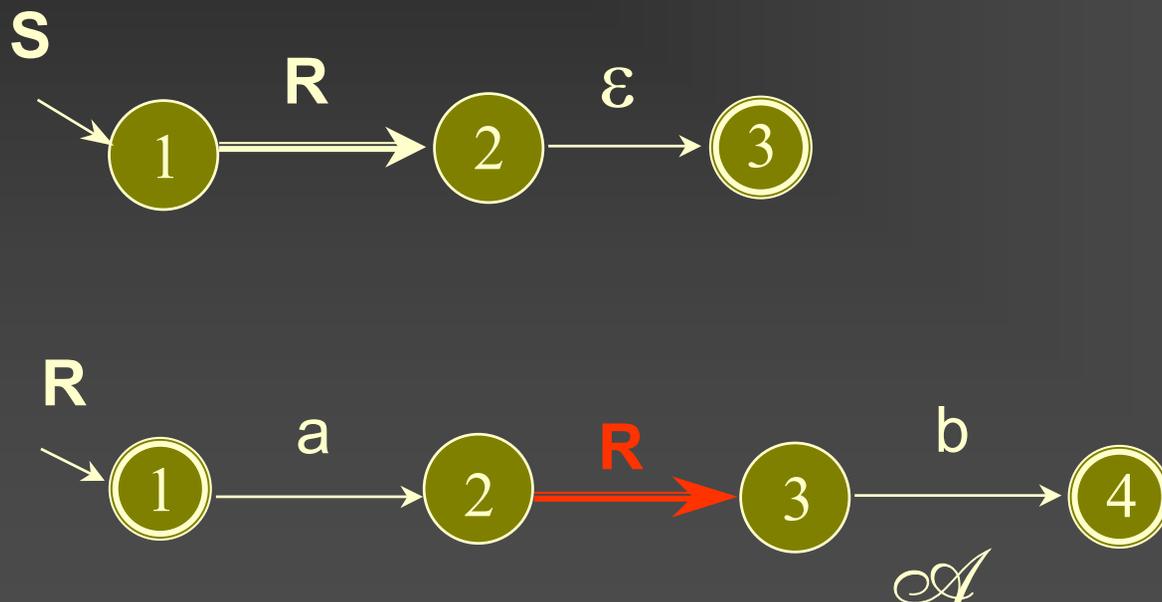


Conceitos Fundamentais

Sentença: abbcc

- Chama R (recursivamente) com o símbolo "a" a ser consumido, empilhando o estado de retorno (3) da máquina R.

Pilha

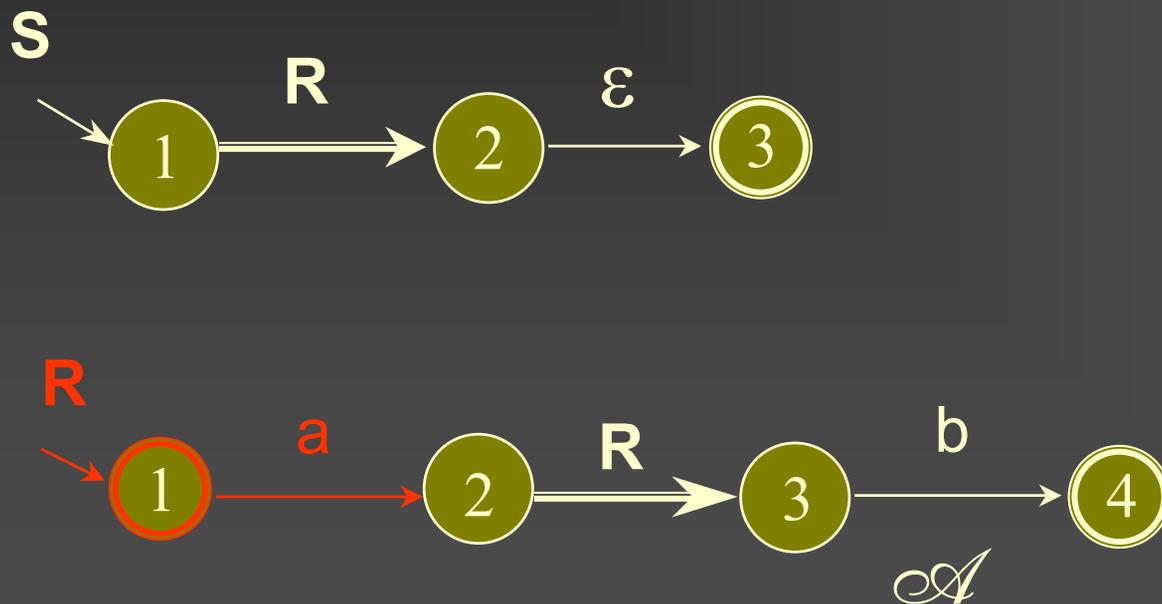


Conceitos Fundamentais

Sentença: bbcc

- Consume símbolo "a".

Pilha

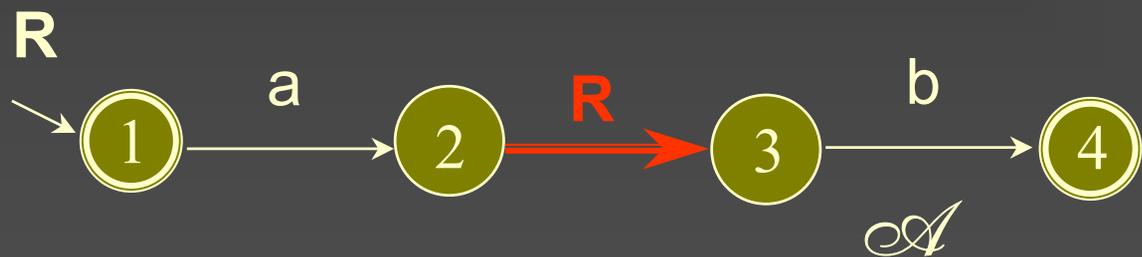


Conceitos Fundamentais

Sentença: bbcc

- Chama R (recursivamente) com o símbolo "b" a ser consumido, empilhando o estado de retorno (3) da máquina R.

Pilha

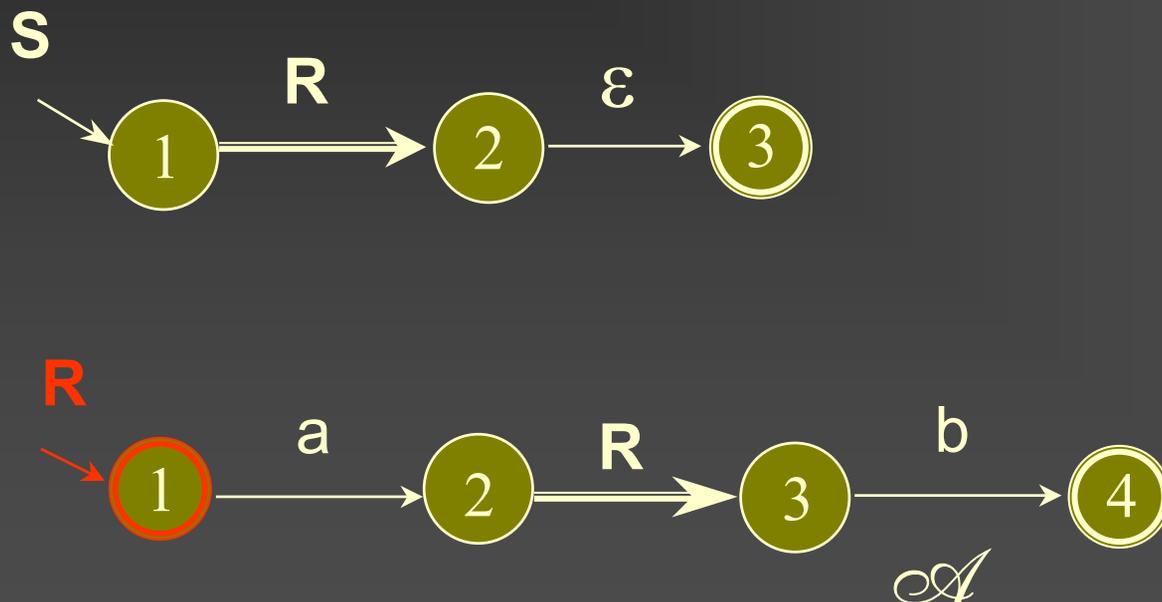


Conceitos Fundamentais

Sentença: bbcc

- Símbolo “b” não pode ser consumido.
- Estado final (1): desempilha estado de “retorno” (no topo da pilha).

Pilha

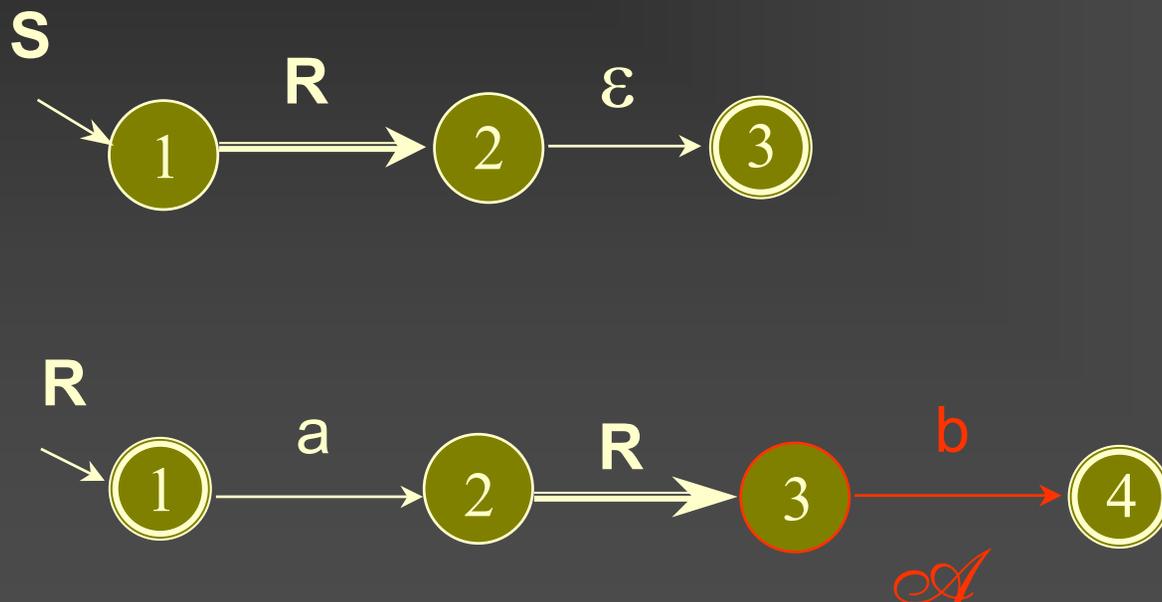


Conceitos Fundamentais

Sentença: bcc

- Consome o símbolo "b".
- Executa ação adaptativa \mathcal{A} .

Pilha

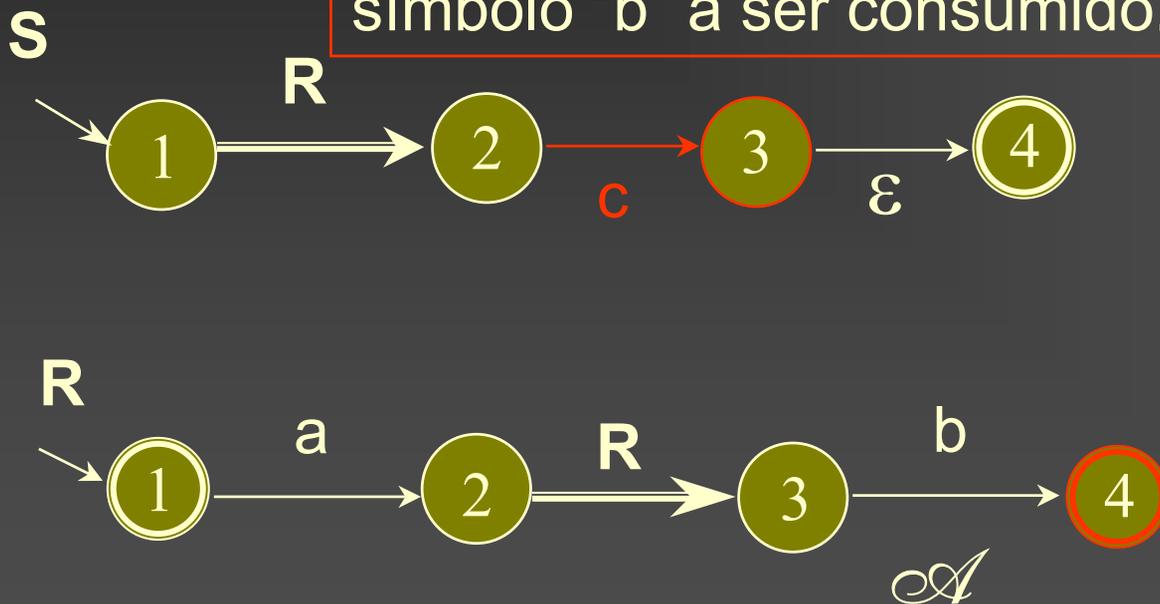


Conceitos Fundamentais

Sentença: bcc

- Acrescenta uma transição de consumo do símbolo "c", na máquina S.
- Atinge o estado 4 (final) em R com o símbolo "b" a ser consumido.

Pilha

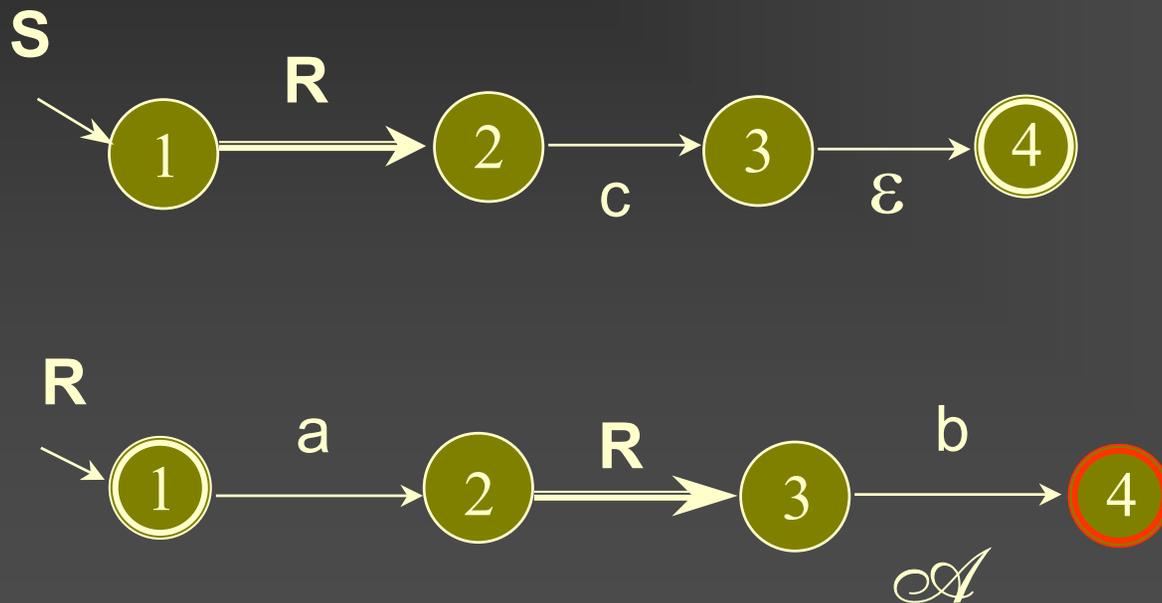


Conceitos Fundamentais

Sentença: bcc

- Símbolo “b” não pode ser consumido.
- Estado final (4): desempilha estado de “retorno” (no topo da pilha).

Pilha

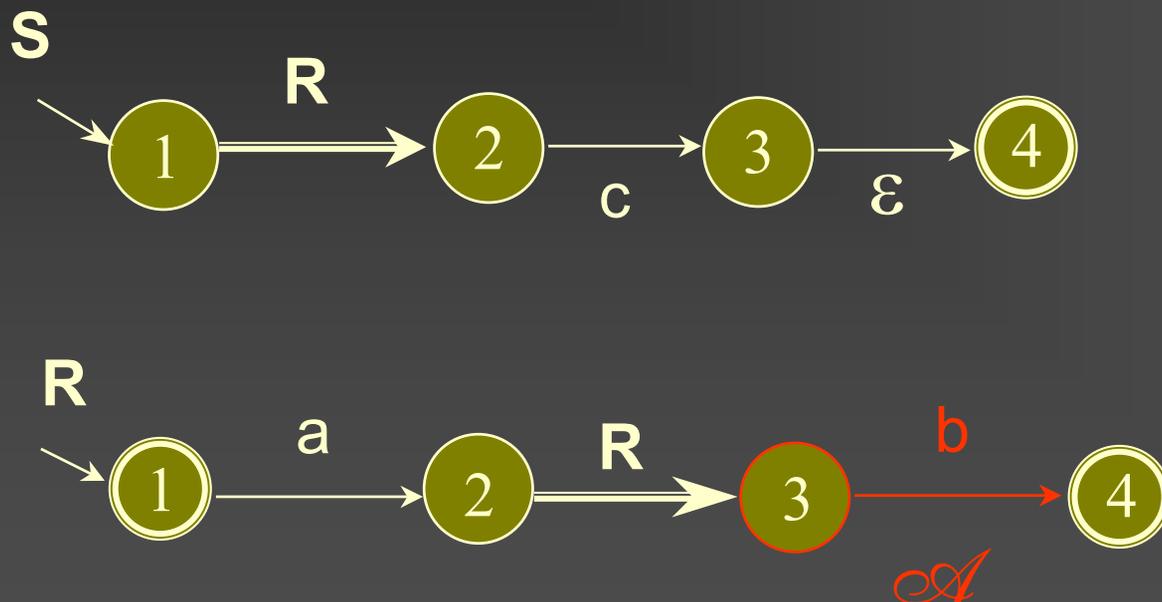


Conceitos Fundamentais

Sentença: cc

- Consome o símbolo "b".
- Executa ação adaptativa \mathcal{A} .

Pilha

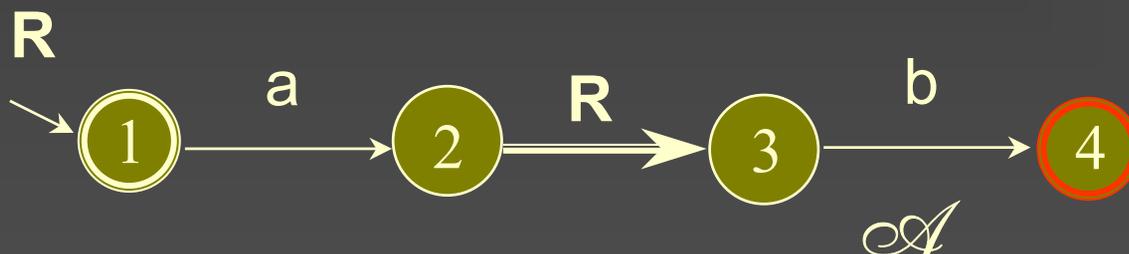
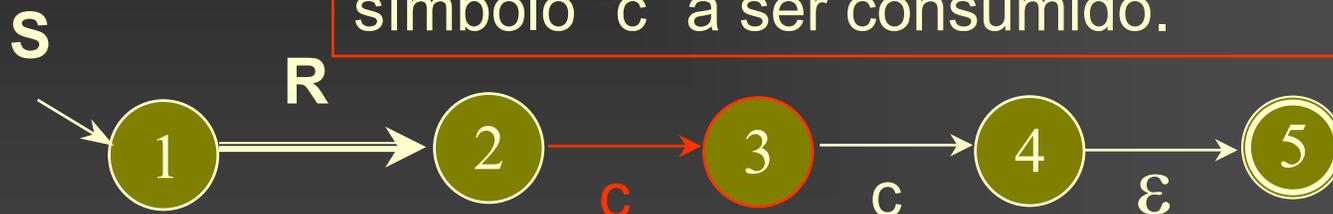


Conceitos Fundamentais

Sentença: cc

- Acrescenta uma transição de consumo do símbolo "c", na máquina S.
- Atinge o estado 4 (final) em R com o símbolo "c" a ser consumido.

Pilha



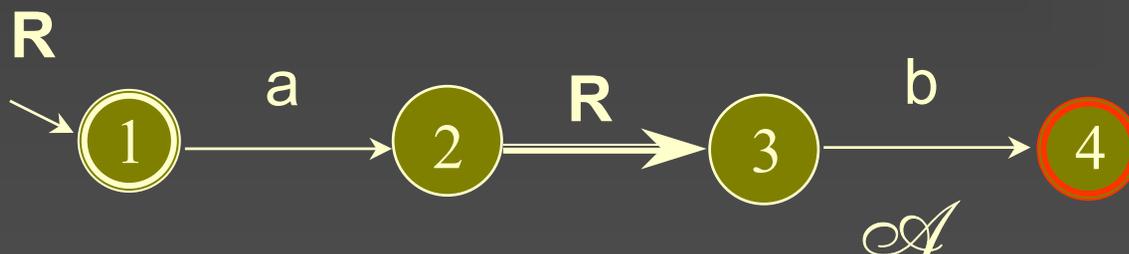
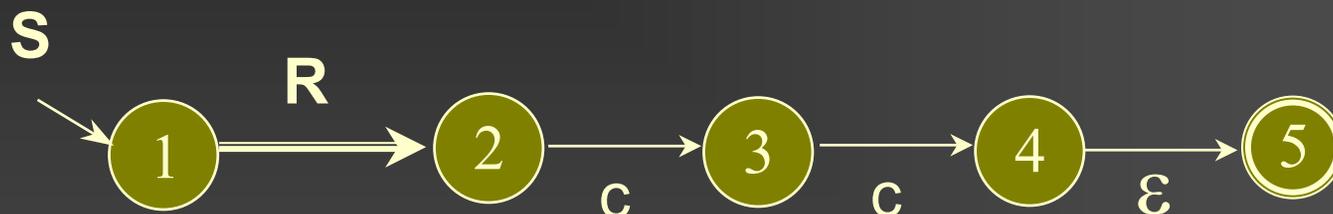
A

Conceitos Fundamentais

Sentença: cc

- Símbolo “c” não pode ser consumido.
- Estado final (4): desempilha estado de “retorno” (no topo da pilha).

Pilha

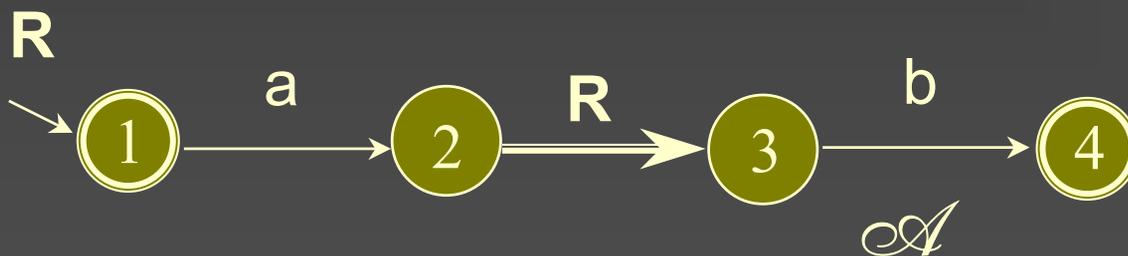
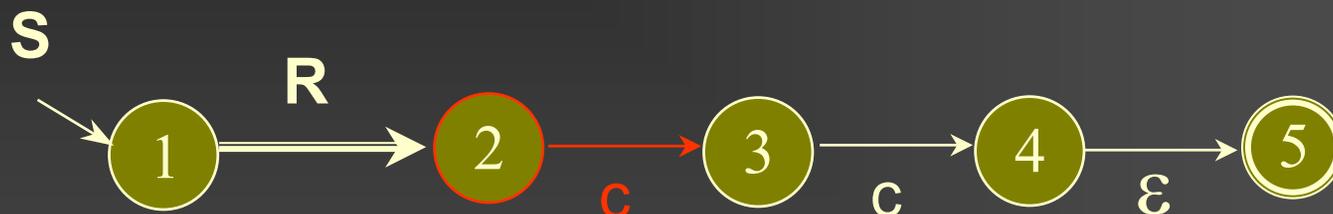


Conceitos Fundamentais

Sentença: c

- Consome o símbolo "c".

Pilha

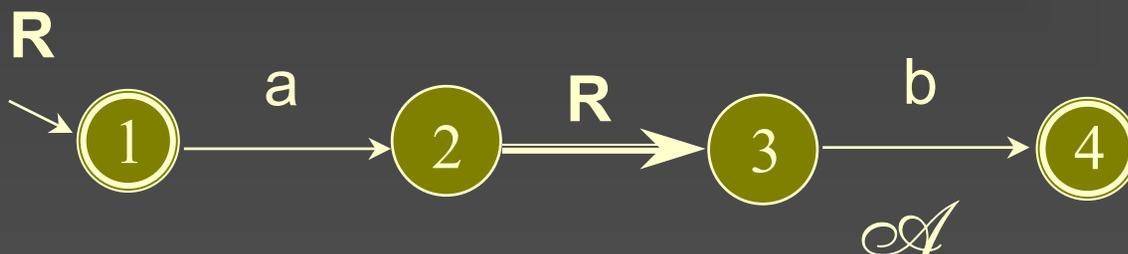
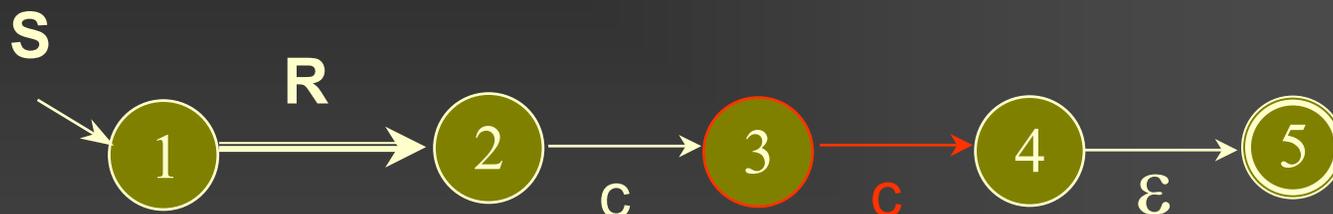


Conceitos Fundamentais

Sentença: ϵ

- Consome o último símbolo "c".

Pilha

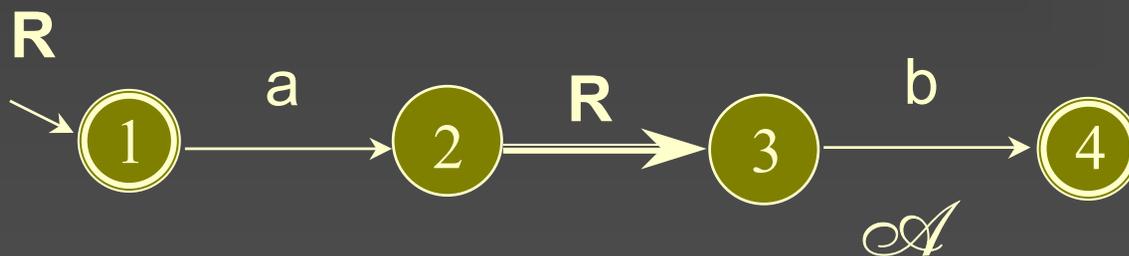
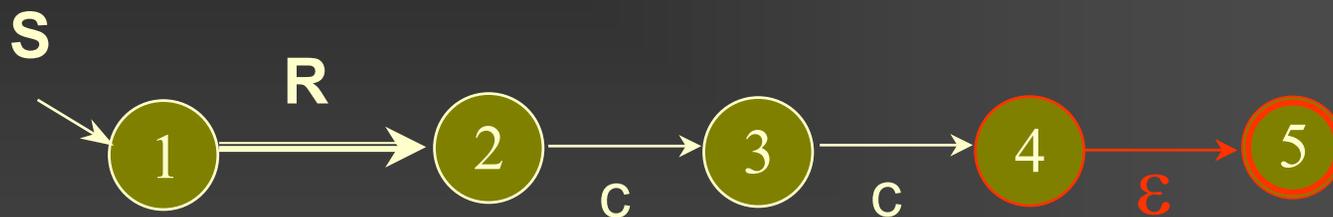


Conceitos Fundamentais

Sentença: ϵ

- FIM.

Pilha



Dispositivos Adaptativos

- Dispositivo subjacente:
 - Descrição formal de um dispositivo não-adaptativo.
 - Mecanismo adaptativo:
 - Funções adaptativas, ações adaptativas e ações adaptativas elementares.
 - Pesquisas:
 - Formalização de “novos” dispositivos adaptativos.
 - “Generalização” do modelo formal.
-

Autômato Adaptativo como Modelo de Computação

- Representação de linguagens de programação imperativas usando AA:
 - Constantes;
 - Variáveis;
 - Operações aritméticas;
 - Estruturas seqüenciais;
 - Estruturas condicionais;
 - Estruturas de repetição;
 - Funções, etc...
-

Autômato Adaptativo como Modelo de Computação

- Cronograma de desenvolvimento:
 - Definição de uma linguagem-base (automatização do processo de obtenção do compilador);
 - “Mapeamento” das operações da linguagem-base em autômatos adaptativos (AdapTools);
 - Definição de um “padrão” de extensão;
 - Implementação de um mecanismo de extensão com base no “padrão” definido;
 - “Síntese” do ambiente completo de programação.
-

Reconhecimento Sintático de Padrões (*Syntactic Pattern Recognition*)

- Padrão (complexo):
 - Sub-padrões.
 - Primitivas.
 - Estrutura de padrões \Leftrightarrow Teoria das linguagens formais.
 - Padrões: sentenças das linguagens.
 - Primitivas: alfabeto da linguagem.
 - Gramática: gera e identifica sentenças (padrões).
-

Reconhecimento Sintático de Padrões (*Syntactic Pattern Recognition*)

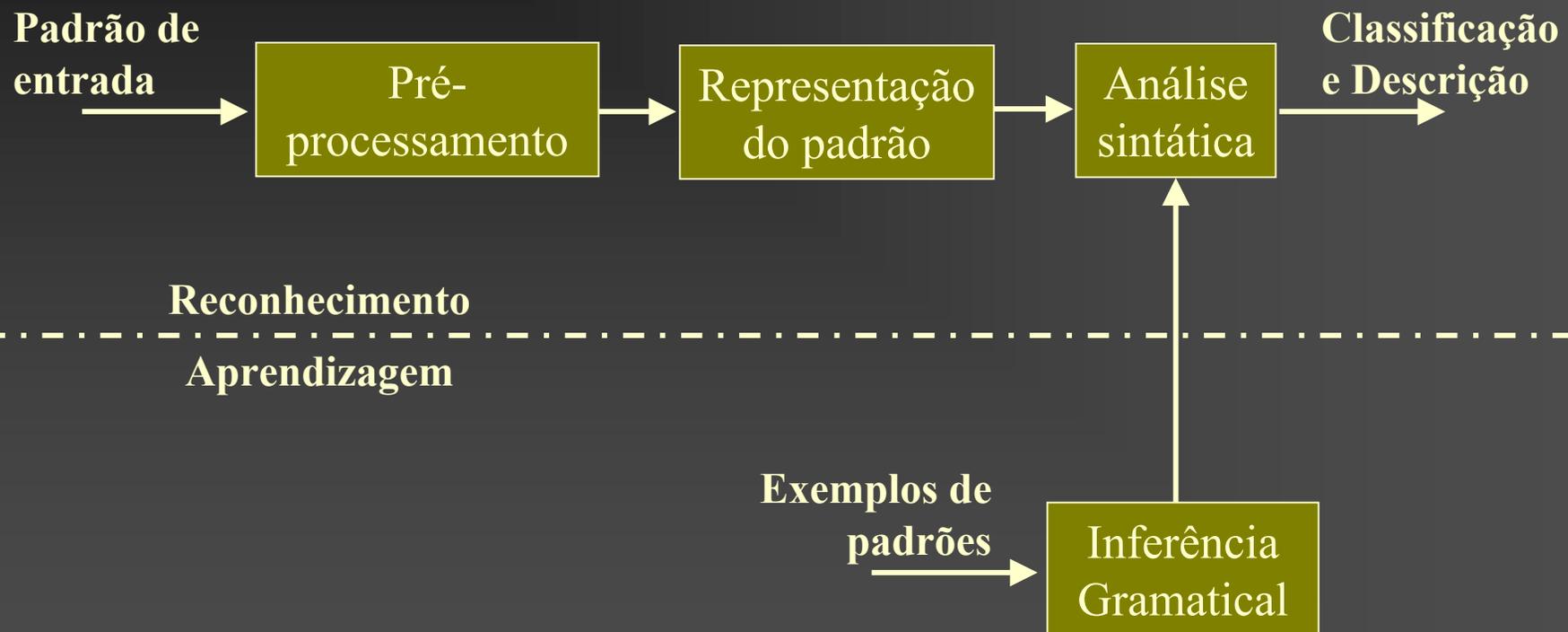
■ Pesquisas:

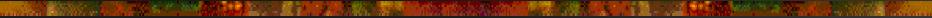
- Escolha do tipo de gramática e suas características;
 - Definição do nível de complexidade descritiva dos padrões;
 - Definição de um conjunto de primitivas ótimas;
 - Inferência gramatical (“treinamento”/aprendizagem);
 - *Parsing* (reconhecimento).
-

Reconhecimento Sintático de Padrões (*Syntactic Pattern Recognition*)

- Aplicações:
 - Processamento de imagens;
 - Visão computacional;
 - Classificação/análise de sinais;
 - Reconhecimento de fisionomia, voz e caracteres;
 - Biologia computacional;
 - Análise de manuscritos;
 - Diagnósticos médicos;
 - Robótica, etc...
-

Reconhecimento Sintático de Padrões (*Syntactic Pattern Recognition*)





LTA – Laboratório de Linguagens e Técnicas Adaptativas
<http://www.pcs.usp.br/~lta>

GPEC – Grupo de Pesquisa em Engenharia e Computação
<http://www.gpec.ucdb.br>

