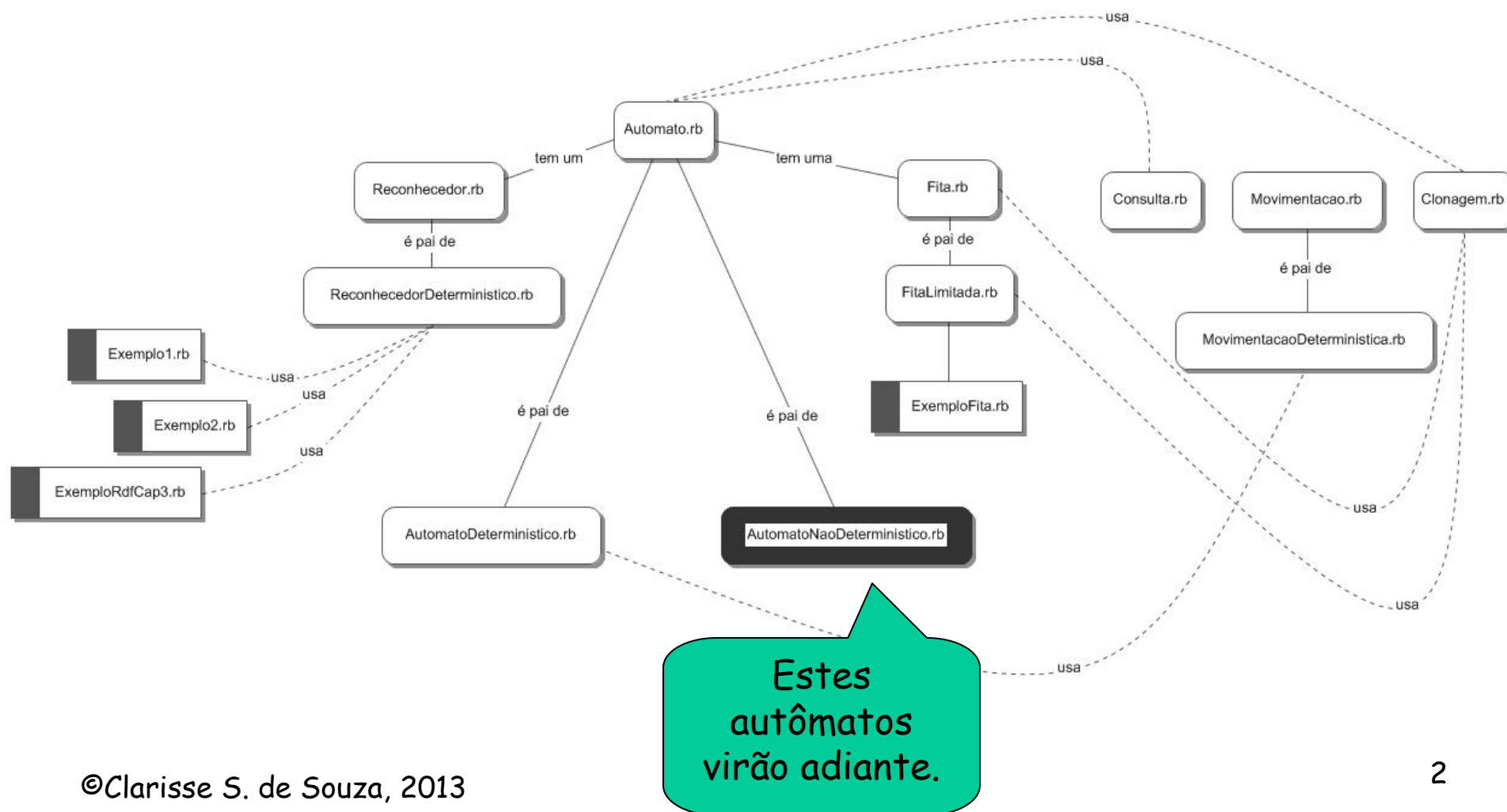


Linguagens Formais e Autômatos (LFA)

Aula de 28/08/2013

Reconhecedores x Gramáticas
Aceitação (decisão) x Derivação (geração)

Programas em Ruby da Aula Passada



Reconhecedores

- **Recebem** cadeias de entrada
- **Decidem** se elas pertencem ou não a uma linguagem
- **Como?**
 - Transicionando entre um conjunto finito de estados, de acordo com regras de transição especificadas.
 - Condição de aceitação: Fita totalmente lida (esgotada) e reconhecedor em um dos estados finais especificados.

Gramáticas

- **Geram** cadeias terminais de símbolos.
- **Como?**
 - Substituindo (ou reescrevendo) cadeias de símbolos, de acordo com regras de derivação (ou reescrita) especificadas. O processo de substituições consecutivas, guiadas pelas regras de uma gramática, se chama **derivação** (ou geração).
 - Condição de "gramaticalidade" de uma cadeia: haver pelo menos um caminho de derivação que leve do símbolo raiz da gramática ("S") até tal cadeia.

Exemplos de Derivação

Regras de Produção de uma Gramática já vista

G_{SC}

$S \rightarrow aSBC$

$S \rightarrow abC$

$CB \rightarrow BC$

$bB \rightarrow bb$

$bC \rightarrow bc$

$cC \rightarrow cc$

$S \rightarrow aSBC \rightarrow aabCBC \rightarrow aabBCC$
 $\rightarrow aabbCC \rightarrow aabbccC \rightarrow aabbcc$

$S \rightarrow abC \rightarrow abc$

Podemos designar o processo inteiro de derivação através de uma forma abreviada:

$\alpha \xRightarrow{G} w$ para

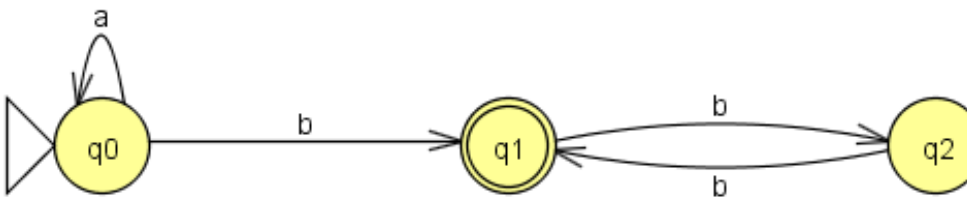
α = símbolo inicial de G

w = cadeia de terminais
resultante

de uma derivação válida

($w \in \Sigma^*$ ou então $w \in \Sigma^+$)

Exercício

Seja o autômato $A =$ 

1. Utilizando o seguinte formalismo simplificado:

$A : Q = \langle \text{conjunto completo de estados} \rangle$

$I = \langle \text{conjunto unitário de estados iniciais} \rangle$

$F = \langle \text{conjunto completo de estados finais} \rangle$

$\Sigma = \langle \text{alfabeto reconhecido} \rangle$

$\delta =$ tuplas de transição (q_i, α, q_j) onde
 $q_i =$ estado corrente

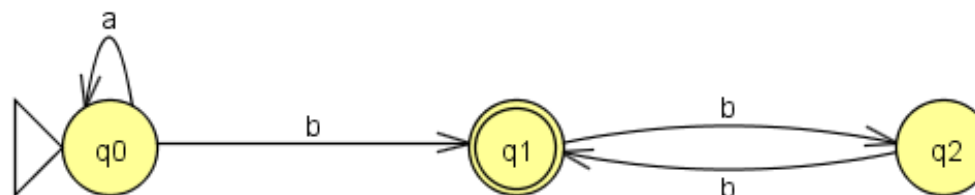
$\alpha =$ símbolo lido pelo cabeçote

$q_j =$ estado-alvo da transição

defina formalmente o autômato A .

Exercício

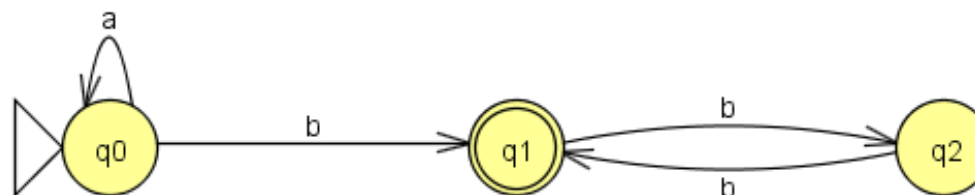
Seja o autômato $A =$



2. Que tipos de cadeias este autômato aceita?
3. Utilizando tuplas (q_i, α, q_j) para representar (estado corrente, símbolo lido, próximo estado), apresente a sequência completa de reconhecimento para as seguintes cadeias:
 - ab
 - aaaaab
 - abbbbb
 - b
 - a
 - bb

Exercício

Seja o autômato $A =$



- Utilizando os programas em Ruby apresentados na aula passada, “implemente” o reconhecedor associado a A .
- Escreva uma **gramática regular** que gere exatamente as mesmas cadeias aceitas pelo reconhecedor que você implementou.

Lembrete - Uma gramática regular é definida por uma tupla $\{V, \Sigma, P, S\}$ onde: V =vocabulário finito e não vazio com TODOS os símbolos que aparecem à esquerda ou direita de regras de reescrita; Σ é o alfabeto da linguagem (isto é, os símbolos terminais que podem aparecer em cadeias “gramaticais” da linguagem); P é o conjunto de regras de reescrita; e S é o símbolo raiz de todas as derivações.

Exercícios

Sejam as gramáticas $G1$, $G2$, $G3$, $G4$ e $G5$, cujas regras de reescrita são as seguintes:

$G1$

$S \rightarrow a$
 $S \rightarrow aS$

$G2$

$S \rightarrow AS$
 $bS \rightarrow Sb$
 $A \rightarrow a$
 $A \rightarrow b$
 $A \rightarrow aA$

$G3$

$S \rightarrow AS$
 $S \rightarrow b$
 $A \rightarrow a$
 $A \rightarrow aA$

$G4$

$S \rightarrow ASB$
 $S \rightarrow c$
 $A \rightarrow a$
 $A \rightarrow aA$
 $B \rightarrow b$
 $B \rightarrow bB$

$G5$

$S \rightarrow XC$
 $X \rightarrow x$
 $X \rightarrow xX$
 $xxxX \rightarrow xxXx$
 $xxC \rightarrow \varepsilon$
 $xxC \rightarrow C$

1. Diga que tipo de gramática é cada uma delas, segundo a Hierarquia de Chomsky.
2. Mostre o caminho de derivação de pelos menos duas cadeias diferentes para cada uma delas, usando a notação do slide 5.