

Linguagens Formais e Autômatos (LFA)

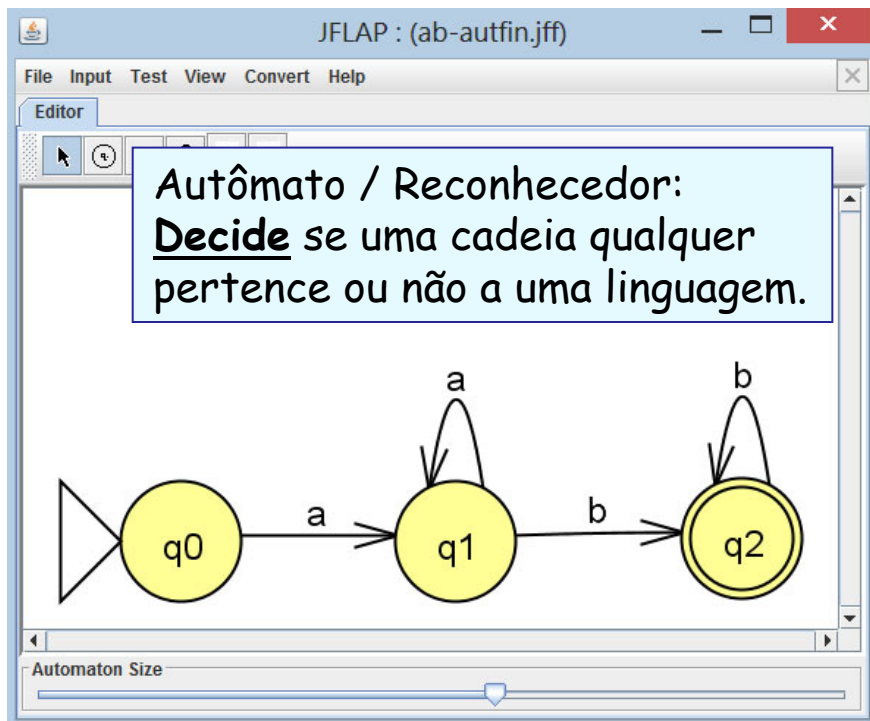
Aula de 21/08/2013

**Gramáticas, Seus Tipos, Algumas
Propriedades e a Hierarquia de Chomsky**

Representações Finitas de Conjuntos

Autômatos e Reconhecedores (aulas 1 e 2)

Gramáticas (aulas 3 e 4)



JFLAP : <untitled2>

File Input Test Convert Help

Editor

Table Text Size

LHS		RHS
S	→	aA
B	→	λ
A	→	aA
A	→	bB
B	→	bB

Gramática:
Gera todas e somente as cadeias
pertencentes a uma linguagem.

Representações Finitas de Conjuntos (de alta cardinalidade ou infinitos)

AUTÔMATOS

- Definem-se através de:
 - Um **conjunto finito de estados** (um único estado inicial, $0 \leq n$ estados intermediários, e $m \geq 1$ estados finais)
 - Um **conjunto finito de regras de transição entre estados**, que, cada uma:
 - especifica **NECESSARIAMENTE** o símbolo “lido” em uma fita de entrada com a cadeia que se quer reconhecer;
 - especifica **NECESSARIAMENTE**:
 - o estado em que a máquina está ao ler o símbolo especificado;
 - o estado para o qual o máquina transiciona ao ler o símbolo especificado
 - especifica **OPCIONALMENTE** o movimento (para esquerda ou para a direita) do **cabeçote de leitura** – o ‘default’ é o movimento para o próximo símbolo não lido (normalmente à direita do símbolo corrente);
 - especifica **OPCIONALMENTE** ações de leitura ou gravação de símbolos na fita de entrada e/ou em dispositivos de memória auxiliar (com cabeçotes próprios).

Exemplo de um autômato A1

Conjunto finito de estados: $Q = (q_0, q_1, q_2)$

Estado inicial: q_0 ; Estado final: q_2 (estado intermediário q_1)

Regras de transição:

ESTADO DE ACEITAÇÃO:
Todos os símbolos da entrada lidos (corrente é #) e estado da máquina é final.

símbolo corrente lido na entrada	estado corrente	próximo estado	movimento do cabeçote	ação do cabeçote
final de fita (#)	<pouco importa>	indefinido	nenhum	nenhuma
a	q_0	q_1	move-se para a próxima posição da fita de entrada direita	lê o símbolo da (nova) posição corrente
a	q_1	q_1	move-se para a próxima posição da fita de entrada direita	lê o símbolo da (nova) posição corrente
b	q_1	q_2	move-se para a próxima posição da fita de entrada direita	lê o símbolo da (nova) posição corrente
b	q_2	q_2	move-se para a próxima posição da fita de entrada direita	lê o símbolo da (nova) posição corrente

Exemplo de um autômato A2

Conjunto finito de estados: $Q = (q_0, q_1, q_2)$, uma pilha de controle

Estado inicial: q_0 ; Estado final: q_2 (estado intermediário q_1)

Regras de transição:

ESTADO DE ACEITAÇÃO:

Todos os símbolos da entrada lidos (corrente é #), pilha vazia e estado da máquina é final.

símbolo corrente lido na entrada	estado corrente	próximo estado	movimento do cabeçote de leitura da fita (F)	ação do cabeçote de leitura da fita (F) e leitura/gravação da pilha (P)
final de fita (#)	<pouco importa>	indefinido	nenhum	nenhuma
ϵ	q_0	q_1	F fica onde está	P lê Z (símbolo de fundo da pilha), consome (pop) Z e grava (push) SZ
ϵ	q_1	q_1	F fica onde está	P lê S no topo da pilha, consome (pop) S e grava (push) ab
ϵ	q_1	q_1	F fica onde está	P lê S no topo da pilha, consome (pop) S e grava (push) aSb
a	q_1	q_1	F move-se para a próxima posição da fita de entrada direita	P lê a no topo da pilha, consome (pop) a e grava (push) ϵ
b	q_1	q_1	F move-se para a próxima posição da fita de entrada direita	P lê b no topo da pilha, consome (pop) b e grava (push) ϵ
ϵ	q_1	q_2	F fica onde está	P lê Z no topo da pilha, consome (pop) Z e grava (push) ϵ

A1 e A2 são autômatos bem diferentes

Quais as diferenças mais óbvias?

INF1626 Linguagens Formais e Autômatos (2013-2)

Exemplo de um autômato A1

Conjunto finito de estados: $Q = (q_0, q_1, q_2)$

Estado inicial: q_0 ; Estado final: q_2 (estado intermediário q_1)

Regras de transição:

símbolo corrente lido na entrada	estado corrente	próximo estado	movimento do cabeçote de leitura da fita (F) e leitura/gravação da pilha (P)
final de fita (#)	<pouco importa>	indefinido	nenhuma
a	q_0	q_1	mov. pos. entr.
a	q_1	q_1	mov. pos. entr.
b	q_1	q_2	mov. pos. entr.
b	q_2	q_2	mov. pos. entrada direita

INF1626 Linguagens Formais e Autômatos (2013-2)

Exemplo de um autômato A2

Conjunto finito de estados: $Q = (q_0, q_1, q_2)$, uma pilha de controle

Estado inicial: q_0 ; Estado final: q_2 (estado intermediário q_1)

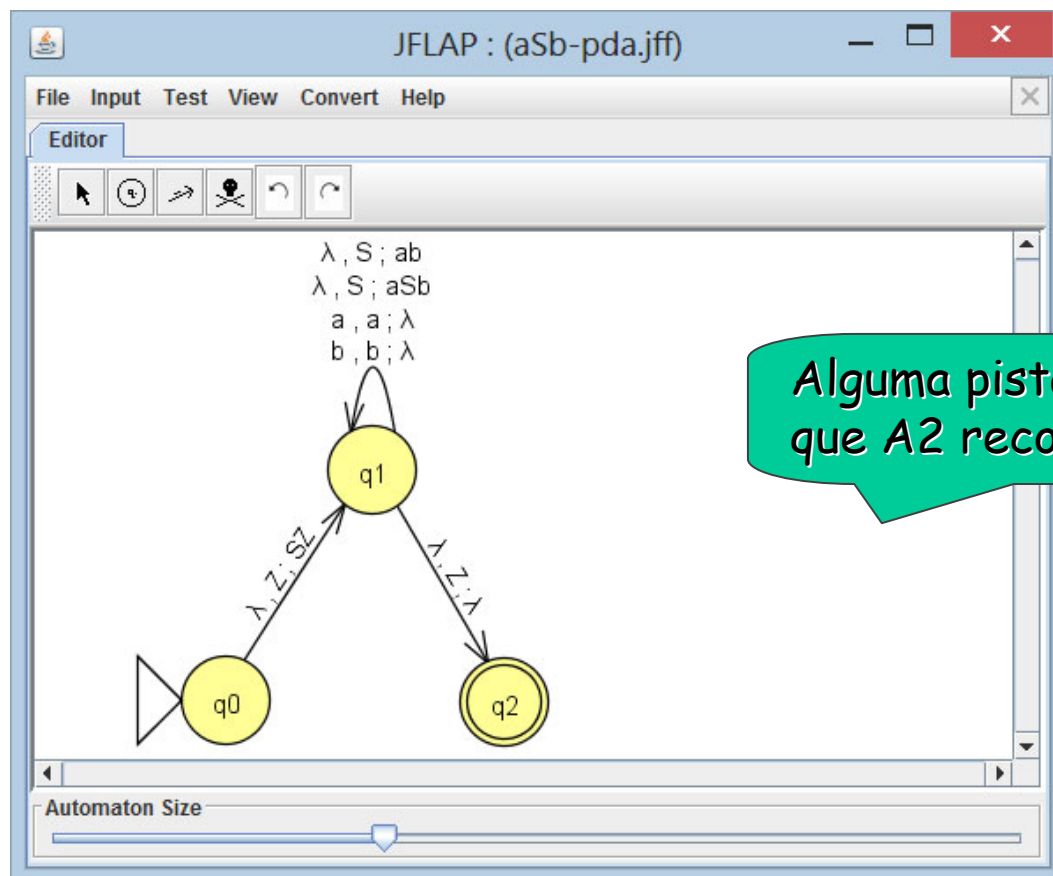
Regras de transição:

símbolo corrente lido na entrada	estado corrente	próximo estado	movimento do cabeçote de leitura da fita (F)	ação do cabeçote de leitura da fita (F) e leitura/gravação da pilha (P)
final de fita (#)	<pouco importa>	indefinido	nenhum	nenhuma
ϵ	q_0	q_1	F fica onde está	P lê Z (símbolo de fundo da pilha), consome (pop) Z e grava (push) SZ
ϵ	q_1	q_1	F fica onde está	P lê S no topo da pilha, consome (pop) S e grava (push) ab
ϵ	q_1	q_1	F fica onde está	P lê S no topo da pilha, consome (pop) S e grava (push) aSb
a	q_1	q_1	F move-se para a próxima posição da fita de entrada direita	P lê a no topo da pilha, consome (pop) a e grava (push) ϵ
b	q_1	q_1	F move-se para a próxima posição da fita de entrada direita	P lê b no topo da pilha, consome (pop) b e grava (push) ϵ
ϵ	q_1	q_2	F fica onde está	P lê Z no topo da pilha, consome (pop) Z e grava (push) ϵ

ESTADO DE ACEITAÇÃO:

Todos os símbolos da entrada lidos (corrente é #), pilha vazia e estado da máquina é final.

Representação do autômato A2



Alguma pista da linguagem que A2 reconhece? :-)

Gramáticas

Especificam linguagens, potencialmente "infinitas" (i.e. com um conjunto infinito de sentenças válidas), por meio de regras de reescrita.

Nas regras de reescrita, cuja forma geral é $\alpha \rightarrow \beta$, os símbolos à esquerda de \rightarrow são substituídos pelos símbolos à direita de \rightarrow . (Ver exercício da aula 3)

Equivalência entre gramáticas e reconhecedores

Uma gramática é equivalente a um reconhecedor se as seguintes duas condições forem satisfeitas:

1. Toda cadeia gerada pela gramática é aceita pelo reconhecedor.
2. Toda cadeia aceita pelo reconhecedor é gerada pela gramática.

Gramáticas tratam da SINTAXE da linguagem

Sintaxe = Características ESTRUTURAIS das cadeias válidas.

As gramáticas vistas em LFA não tratam da **semântica** das linguagens (i.e. das características de SIGNIFICADO das cadeias válidas).

Em linguagens computacionais, a estrutura deve estar fortemente ligada ao significado (ou então o *design* da linguagem é ineficiente, pois exige mais camadas intermediárias de manipulação simbólica para determinar o significado de uma cadeia).

Especificação FORMAL de gramáticas

Gramáticas são QUÁDRUPLAS: $G = \{V, \Sigma, P, S\}$ onde

- V é um vocabulário finito e não vazio com “todos” os símbolos que aparecem à esquerda ou direita de \rightarrow nas regras de reescrita (terminais e não terminais).
- Σ é o alfabeto, finito e não vazio, de símbolos que podem aparecer em cadeias bem-formadas da linguagem (terminais).
- P é o conjunto finito e não vazio de regras de reescrita (de substituição ou produções).
- S é a raiz ou símbolo inicial da gramática (e $S \in V$).

Características das regras de produção de uma Gramática

Recordando a forma geral: $\alpha \rightarrow \beta$

Símbolos não-terminais: pertencem ao vocabulário da gramática, mas não ao seu alfabeto.

Símbolos terminais = alfabeto da gramática (e da linguagem por ela definida).

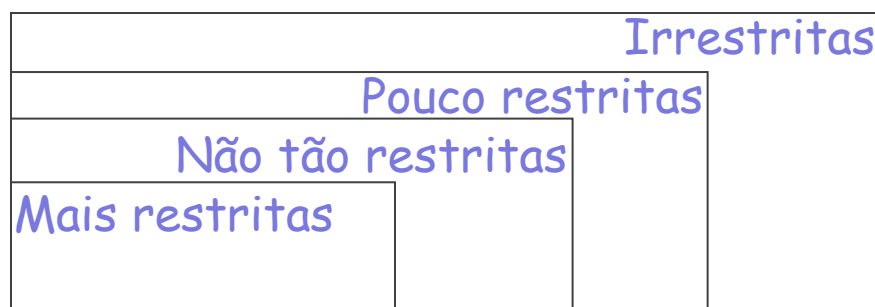
As gramáticas variam de acordo com **restrições** que se imponham a α e β .

- Que **tipos** de símbolos podem aparecer em α e β ?
- **Quantos** símbolos podem aparecer em α e β ?
- Em que **ordem** os símbolos podem aparecer?

A Hierarquia de Chomsky

Princípio geral: O relaxamento sucessivo de restrições sobre α e β defini "tipos" de linguagens, contidas umas nas outras.

São 4 os tipos **interessantes** de linguagens formais:



As linguagens mais restritas, tipo 3, REGULARES

Restrições sobre $\alpha \rightarrow \beta$

$|\alpha| = 1$ e α é um não terminal

β é um terminal, um não-terminal, um terminal seguido de um não terminal (ou vice versa) ou ainda uma cadeia vazia.

$$\alpha \in N \text{ e } |\alpha| = 1$$

$$\beta \in \Sigma, \beta \in N, \beta \in \Sigma N, \beta = \varepsilon \quad (\text{LL})$$

$$\beta \in \Sigma, \beta \in N, \beta \in N\Sigma, \beta = \varepsilon \quad (\text{LR})$$

Linguagens lineares à direita são aquelas onde, nas produções em que β tem um terminal um não-terminal, o **terminal vem à direita (i.e. sucede o não terminal)**.

Linguagens lineares à esquerda são aquelas onde, nas produções em que β tem um terminal um não-terminal, o **terminal vem à esquerda (i.e. precede o não terminal)**.

As linguagens não tão restritas, tipo 2, LIVRES DE CONTEXTO

Restrições sobre $\alpha \rightarrow \beta$

$|\alpha| = 1$ e α é um não terminal

$\beta \in V^*$ é uma cadeia qualquer formada por terminais e não terminais, ou uma cadeia vazia.

$$\begin{aligned}\alpha &\in N \text{ e } |\alpha| = 1 \\ V &= N \cup \Sigma \\ \beta &\in V^*\end{aligned}$$

As linguagens pouco restritas, tipo 1, **SENSÍVEIS A CONTEXTO**

Restrições sobre $\alpha \rightarrow \beta$

$|\alpha| \leq |\beta|$ e α contém ao menos um não-terminal.

$\beta \in$ é uma cadeia qualquer formada por terminais e não terminais.

Nota: como α tem de conter ao menos 1 não terminal, seu tamanho mínimo é 1. E como $|\alpha| \leq |\beta|$ as linguagens sensíveis a contexto não podem conter a cadeia vazia à direita de suas regras de produção.

(Linguagens que aceitam a produção vazia são a rigor linguagens de outro tipo.)

$$\begin{aligned} V &= N \cup \Sigma \\ \alpha &\in V^* N V^* \text{ e } |\alpha| \leq |\beta| \\ \beta &\in V^+ \end{aligned}$$

As linguagens **IRRESTRITAS**, tipo 0

Restrições sobre $\alpha \rightarrow \beta$

α contém ao menos um não-terminal.

$\beta \in$ é uma cadeia qualquer formada por terminais e não terminais.

$$\begin{aligned} V &= N \cup \Sigma \\ \alpha &\in V^* N V^* \\ \beta &\in V^* \end{aligned}$$

Perguntas

1. Que tipos de estruturas cada gramática gera?
2. Que tipos de símbolos podem ser substituídos? Por quais?

Para motivar as respostas:

Que tipo de linguagem é esta?

$V \rightarrow \{a, b, c, S, B, C\}$

$\Sigma \rightarrow \{a, b, c\}$

$P \rightarrow \{S \rightarrow aSBC, S \rightarrow abC, CB \rightarrow BC, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

Que linguagem está definida?