Gramáticas Sensíveis ao Contexto (GSC)

Linguagens Sensíveis ao Contexto (LSC)

Autômatos Linearmente Limitados

(ALL)

Gramática Sensível ao Contexto

- **Definição**: Uma gramática G é **sensível ao contexto** se cada produção ou é da forma
- $yAz \rightarrow ywz$, para $A \in Vn$, $y; z \in (Vn \cup Vt)^*$, $w \in (Vn \cup Vt)^+$; ou
- $S \rightarrow \lambda$, dado que S não aparece no lado direito de nenhuma produção.
- Uma linguagem é sensível ao contexto se pode ser gerada por uma gramática sensível ao contexto.

Exemplo de Gramática Sensível ao Contexto:

G = (Vn,Vt,S,P); Vt ={a; b}, Vn= {S,B,C} e

P ={

1. S → aSBC
2. S → aBC
3. CB → BC
4. aB → ab
5. bB → bb
6. bC → bc
7. cC → cc

}

A linguagem L(G) contém a palavra $a^nb^nc^n$ para cada $n \geq 1$, pois:

- Pode-se aplicar a produção (1) n-1 vezes para chegar a $S \rightarrow^* a^{n-1}S(BC)^{n-1}$.
- Depois usa-se a (2) para $S \rightarrow^* a^n(BC)^n$.
- A produção (3) permite arranjar os Bs e Cs tal que todo B preceda todos os Cs. Por exemplo, se n = 3:

aaaBCBCBC → aaaBBCCBC → aaaBBCBCC → aaaBBBCCC

Portanto, $S \rightarrow^* a^nB^nC^n$.

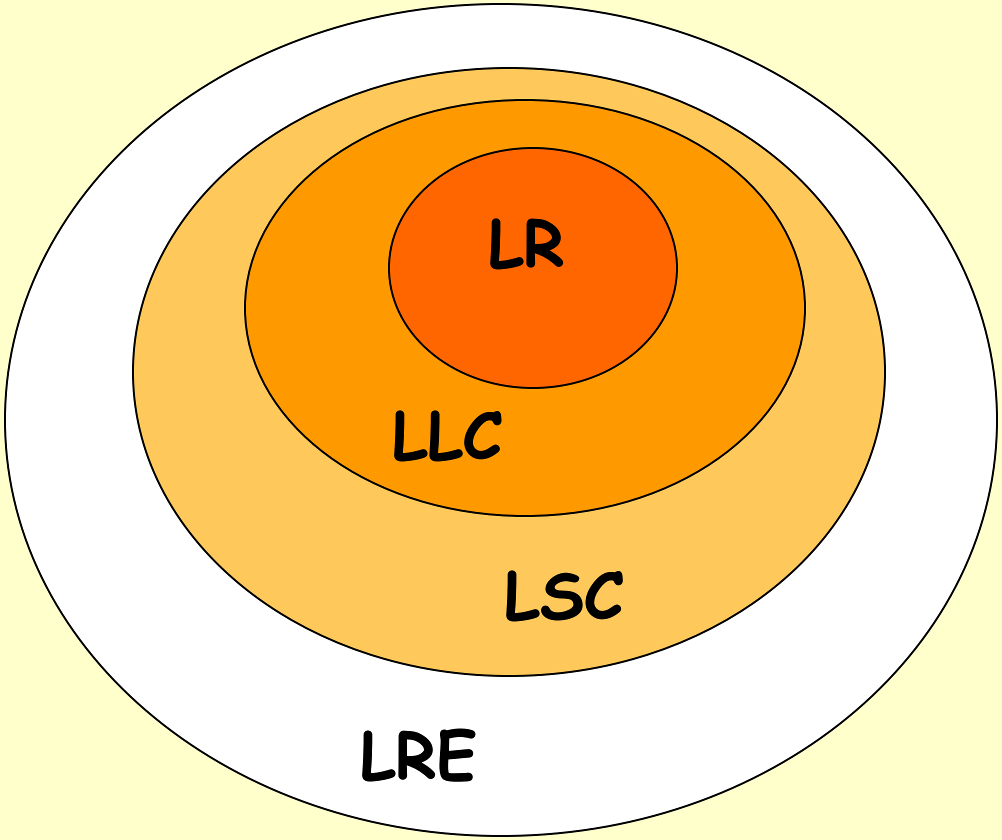
- Depois usa-se (4) uma vez para chegar a $S \rightarrow^* a^nbB^{n-1}C^n$.
- Daí usa-se a produção (5) n- 1 vezes: $S \rightarrow^* a^nb^nC^n$.
- Finalmente, usa-se a produção (6) uma vez e a produção (7) n - 1 vezes para chegar a $S \rightarrow^* a^nb^nc^n$.
- Seria necessário mostrar também que as cadeias $a^nb^nc^n$, $n \geq 1$, são as únicas cadeias terminais de L(G).

Note que nem toda gramática livre de contexto é sensível ao contexto, porque as produções livres de contexto podem ser da forma $A \rightarrow \lambda$. Entretanto, toda linguagem livre de contexto é uma linguagem sensível ao contexto.

Linguagens Sensíveis ao Contexto - LSC

Teorema: Uma linguagem L é sensível ao contexto se e somente se existe alguma gramática G tal que L = L(G) onde toda produção de G da forma $u \rightarrow v$ tem a propriedade de que $0 < |u| \leq |v|$ com uma exceção: se $\lambda \in L(G)$, então a regra $S \rightarrow \lambda$ está também presente e neste caso S não pode aparecer no lado direito de nenhuma produção.

Hierarquia de Chomsky



LR = Linguagens Regulares
LLC = Linguagens Livres de Contexto
LSL =Linguagens Sensíveis ao Contexto
LEF = Linguagens Recursivamente Enumeráveis

Gramáticas e reconhecedores

Linguagens/Gramáticas	**Reconhecedores**
Rec. Enumerável	Máquina de Turing
Sensível ao contexto	Máquina de Turing com memória limitada ou Autômato Linearmente Limitado
Livre de contexto	Autômato a pilha
Regular	Autômato finito

Linguagens e Reconhecedores

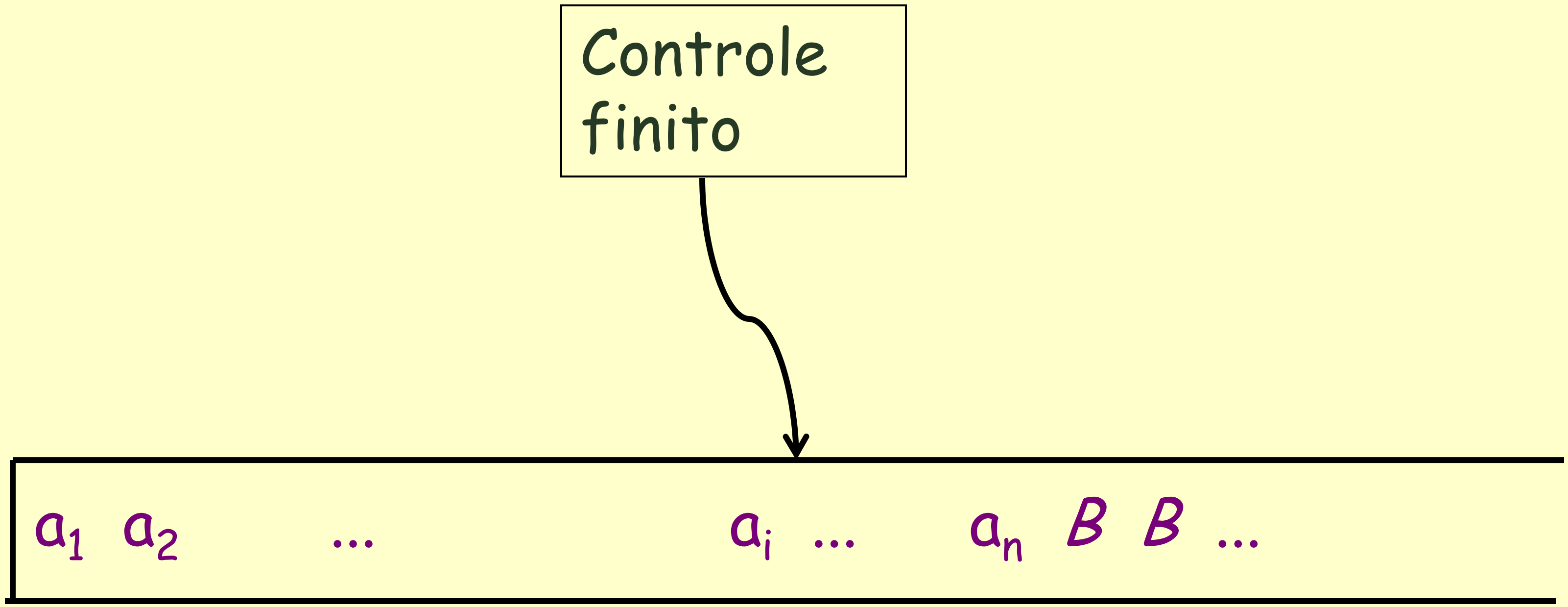
Linguagem	Gramática	Reconhecedor	Tempo para reconhecer w; |w|=n
Tipo 0: Linguagens Computáveis ou Recursivamente Enumeráveis	Gramáticas com Estrutura de Frase	Máquinas de Turing	NP-completo
Tipo 1: Sensíveis ao Contexto	Gramáticas Sensíveis ao Contexto	Máquinas de Turing com memória limitada	Exponencial: $O(2^n)$
Tipo 2: Livres de Contexto	Gramáticas Livres de Contexto	Autômatos à Pilha	Polinomial Espaço: $O(n)$; Tempo: Geral: $O(n^3)$; Não-ambíguas: $O(n^2)$; Se P= A->aB ou A->Ba ou A->a: $O(n)$
Tipo 3: Conjuntos Regulares	Gramáticas Regulares	Autômatos Finitos	Linear: $O(n)$ e $O(\|E\|)$ (no tamanho do AF)

Autômatos Linearmente Limitados -ALL: Reconhecedores de LLCs

Para que o autômato que conhecemos até o momento (AP) seja capaz de reconhecer LSCs que não sejam LLCs, ele é estendido da forma:

- A cabeça da fita pode mover-se à direita ou à esquerda;
- É possível ler E escrever na fita.
- As 2 ações acima só acontecem no espaço da fita ocupado pela cadeia de entrada.

ALL



Inicialmente, a entrada é colocada na fita. Todas as outras células (infinitamente à esquerda e à direita) têm um símbolo especial da fita, *B* (*branco*).
A cabeça da fita fica posicionada em uma das células. No início, a cabeça está posicionada na célula mais à esquerda que contém a entrada.

Um *movimento* do ALL é uma função do estado do controle finito e do símbolo atual da fita. Em um movimento:

1. Mudará de estado (opcionalmente para o mesmo).
2. Gravará um símbolo de fita na célula atual, substituindo o existente (podendo ser o mesmo).
3. Movimentará (necessariamente) a cabeça da fita uma célula à esquerda ou à direita.

Notação formal

$M = (Q, \Sigma, \Gamma, \delta, qo, F)$

Controle finito

Q=conj. finito de estados;
F = conj. estados finais (de aceitação)

Γ = alfabeto finito da fita

cabeça da fita

a_1 a_2 ... a_i ... a_n B B ...

Σ = alfabeto finito de entrada

Função de transição δ (determinística):

δ: Q x $\Gamma \rightarrow$ Q x Γ x {L,R}

Ou seja, $\delta(q,a) = (p,b,D)$ onde:

- *p* é o próximo estado em Q;
- *b* é o símbolo que substituirá *a* na fita;
- *D* é uma direção (esquerda ou direita) em que a cabeça da fita irá se mover.

A função de transição δ pode ser não-determinística.

Ou seja, $\delta(q,a) = \{(p_1,b_1,D_1), \ldots(pi,bi,Di)\}$

As ALLs não determinísticas reconhecem todas as LSCs.

Ex. Um ALL para reconhecer a LSC

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

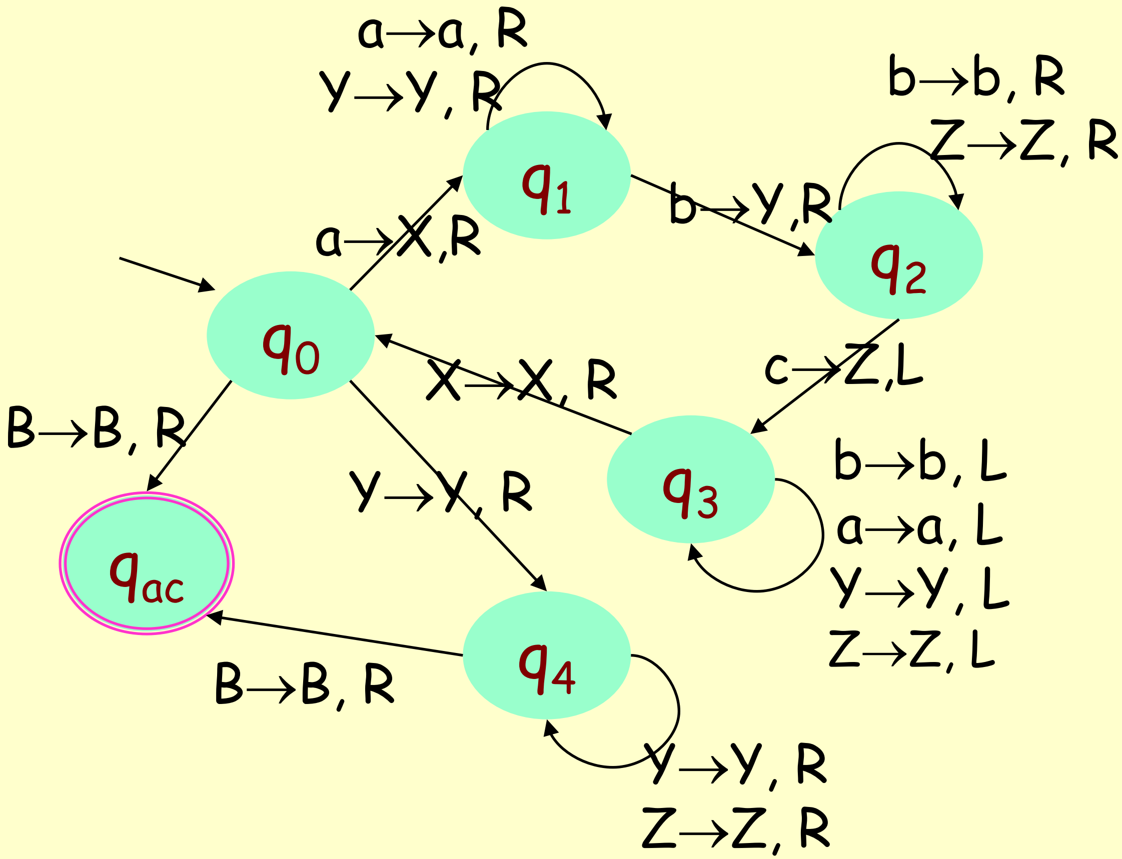
Exemplos:

Pertence à L: aaabbbccc

Não Pertence à L: aaabbcccc

Ideia: em cada passo, reconhecer um a, um b e um c, substituindo-os por X, Y e Z, respectivamente.

1. $Q = \{q_0, q_1, q_2, q_3, q_4, q_{ac}\}$
2. $\Sigma = \{a, b, c\}$
3. $\Gamma = \{a, b, c, B, X, Y, Z\}$
4. δ a seguir.
5. q_0 – o estado inicial
6. $F = \{q_{ac}\}$


a→a, R
Y→Y, R
b→b, R
Z→Z, R
q1
b→Y,R
q2
a→X,R
q0
X→X, R
c→Z,L
B→B, R
q3
b→b, L
Y→Y, R
a→a, L
qac
Y→Y, L
Z→Z, L
B→B, R
q4
Y→Y, R
Z→Z, R

A seguir...

Máquinas de Turing (MT) e Linguagens Recursivamente Enumeráveis (LRE)

- Máquinas de Turing são ALLs sem limitação do uso da fita, ou seja, com memória ilimitada;
- As Linguagens Recursivamente Enumeráveis são geradas por Gramáticas Irrestritas;
- As LREs coincidem com o conjunto de funções computáveis, ou seja, funções que podem ser resolvidas por um computador (qualquer).