ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 11

Algoritmo CYK de análise sintática e
Cap. 2.2 – Autômato com Pilha

Profa. Ariane Machado Lima
ariane.machado@usp.br

Forma Normal de Chomsky

- **Definição 2.8:**
  Uma GLC está na Forma Normal de Chomsky se:

  a) Toda regra de substituição é da forma

  $$A \rightarrow BC \quad \text{ou } A \rightarrow a$$

  onde B,C são variáveis, a é símbolo terminal;

  b) A variável inicial S não pode aparecer no lado direito de nenhuma regra;

  c) Somente a variável inicial pode ter a regra

  $$S \rightarrow \varepsilon \ .$$

TEOREMA 2.9

Qualquer linguagem livre-do-contexto é gerada por uma gramática livre-do-contexto na forma normal de Chomsky.

Algoritmo CYK para análise sintática

- Problema:
  - Dado uma GLC G e uma cadeia w, decidir se G gera a cadeia w
  - Redefinição do problema em termos de linguagem:
    - A_{GLC} = { (H, x) | H é uma GLC que gera a cadeia x }
    - Dada uma Problema: (G, W) $\in A_{GLC}$?

Algoritmo CYK para análise sintática

- Se G for uma gramática qualquer:
  - Testar todas as possíveis derivações: loop infinito
- Se G estiver na Forma Normal de Chomsky:
  - Teorema: qualquer derivação de uma cadeia não vazia *w* tem exatamente 2*n*-1 passos, onde $n = |w|$.
    - Ideia da demonstração:
      - n-1 passos para gerar a cadeia com n variáveis
      - n passos para substituir as variáveis por terminais
    - Método de força bruta (para n>0): custo exponencial
    - Teste todas as derivações possíveis com 2n-1 passos.
    - Se alguma das derivações gerar w, aceite; senão, rejeite

Algoritmo CYK para análise sintática

- Algoritmo CYK (Cocke–Younger–Kasami)
  - G deve estar na Forma Normal de Chomsky
  - Complexidade: Polinomial - $O(n^3 m)$ onde n é o tamanho da cadeia e m é o número de regras de G

Algoritmo CYK para análise sintática

Exemplo:

Gramática original:

$S \rightarrow aSb \mid bSa \mid SS \mid \varepsilon$

- Conversão para FNC:

1) Criação de nova variável inicial:

$S_0 \rightarrow S$

$S \rightarrow aSb \mid bSa \mid SS \mid \varepsilon$

2) Eliminação de substituições ε :

$S_0 \rightarrow S \mid \varepsilon$

$S \rightarrow aSb \mid bSa \mid SS \mid ab \mid ba \mid S$

3) Eliminação de regras unitárias:

$S_0 \rightarrow \varepsilon \mid aSb \mid bSa \mid SS \mid ab \mid ba$

$S \rightarrow aSb \mid bSa \mid SS \mid ab \mid ba$

4) Padronização:

$S_0 \rightarrow \varepsilon \mid AT \mid BU \mid SS \mid AB \mid BA$

$S \rightarrow AT \mid BU \mid SS \mid AB \mid BA$

$T \rightarrow SB$

$U \rightarrow SA$

$A \rightarrow a$

$B \rightarrow b$

Algoritmo CYK para análise sintática

- Algoritmo CYK (Cocke–Younger–Kasami)
  - G deve estar na Forma Normal de Chomsky
  - Complexidade: Polinomial - $O(n^3 m)$ onde n é o tamanho da cadeia e m é o número de regras de G
- **Programação dinâmica**: uso de soluções de subproblemas menores para resolver subproblemas maiores (até chegar à solução do problema original)
- Tabela $n \times n$:
  - Para $i \leq j$, a entrada (i,j) da tabela contém todas as variáveis que geram a subcadeia $w_i \, w_{i+1} \ldots w_j$
  - Tratam-se subcadeias de tamanhos crescentes (começando de 1)

Algoritmo CYK para análise sintática

$D =$ "On input $w = w_1 \cdots w_n$:

1. If $w = \varepsilon$ and $S \to \varepsilon$ is a rule, *accept*. [[handle $w = \varepsilon$ case]]
2. For $i = 1$ to n: [[examine each substring of length 1]]
3. For each variable A:
4. Test whether $A \to \mathtt{b}$ is a rule, where $\mathtt{b} = w_i$.
5. If so, place A in $table(i, i)$.
6. For $l = 2$ to n: [[l is the length of the substring]]
7. For $i = 1$ to $n - l + 1$: [[i is the start position of the substring]]
8. Let $j = i + l - 1$, [[j is the end position of the substring]]
9. For $k = i$ to $j - 1$: [[k is the split position]]
10. For each rule $A \to BC$:
11. If $table(i, k)$ contains B and $table(k + 1, j)$ contains C, put A in $table(i, j)$.
12. If S is in $table(1, n)$, *accept*. Otherwise, *reject*."

Algoritmo CYK para análise sintática

Exemplo (cont):

- Aplicação algoritmo CYK:

Grámática na FNC:

$S_0 \rightarrow \varepsilon \mid AT \mid BU \mid SS \mid AB \mid BA$
$S \rightarrow AT \mid BU \mid SS \mid AB \mid BA$
$T \rightarrow SB$
$U \rightarrow SA$
$A \rightarrow a$
$B \rightarrow b$

Cadeia:

a b a a b b

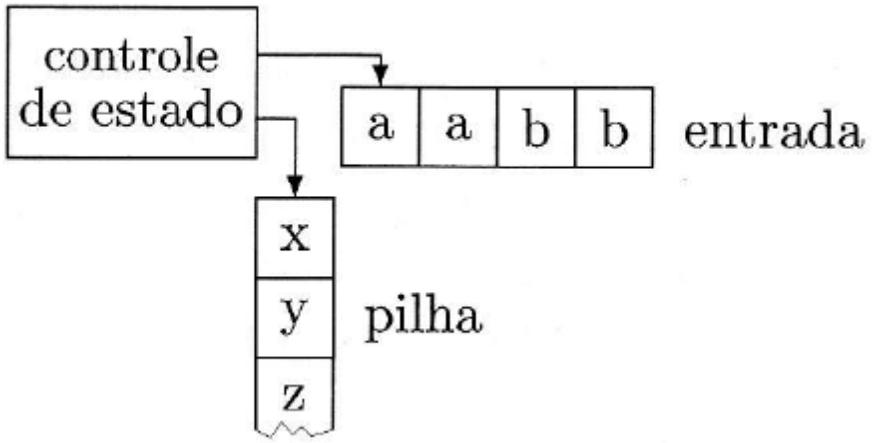
Tabela:

	a	b	a	a	b	b
a	A	S_0, S	U	⊘	⊘	S_0, S
b		B	S_0, S	U	S_0, S	T
a			A	⊘	⊘	S_0, S
a				A	S_0, S	T
b					B	⊘
b						B

Cap 2.2 – Autômato com pilha (AP)

Cap 2.2 – Autômato com pilha (AP)

- Autômato finito com uma memória adicional (leitura e escrita DO TOPO da pilha)



- Lembram de B = $\{0^n1^n \mid n >= 0\}$?

Cap 2.2 – Autômato com pilha (AP)

- Determinísticos e não-determinísticos
- NÃO são equivalentes
  - Autômatos a pilha não determinísticos reconhecem mais linguagens
- Autômatos a pilha não-determinísticos são equivalentes a gramáticas livres de contexto (são os que estudaremos aqui)

Definição formal
Autômato com Pilha (AP) não determinístico

DEFINIÇÃO 2.13

Um ***autômato com pilha*** é uma 6-upla $(Q, \Sigma, \Gamma, \delta, q_0, F)$, onde Q, Σ, Γ e F são todos conjuntos finitos, e

1. Q é o conjunto de estados,
2. Σ é o alfabeto de entrada,
3. Γ é o alfabeto de pilha,
4. $\delta: Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow \mathcal{P}(Q \times \Gamma_\varepsilon)$ é a função de transição,
5. $q_0 \in Q$ é o estado inicial, e
6. $F \subseteq Q$ é o conjunto de estados de aceitação.

Exemplo

$\{0^n 1^n | n \geq 0\}$. Suponha que M_1 seja $(Q, \Sigma, \Gamma, \delta, q_1, F)$

Exemplo

$\{0^n 1^n | n \geq 0\}$. Suponha que M_1 seja $(Q, \Sigma, \Gamma, \delta, q_1, F)$

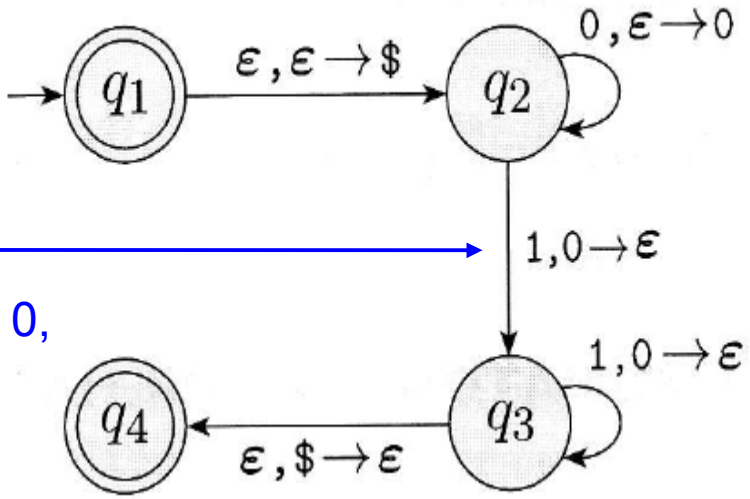
$Q = \{q_1, q_2, q_3, q_4\}$,

$\Sigma = \{0,1\}$,

$\Gamma = \{0, \$\}$,

$F = \{q_1, q_4\}$, e

Se estou em q2
e o próximo símbolo é 1
e o topo da pilha é 0, desempilho o 0,
empilho o épsilon e vou para q3



δ é dada pela tabela abaixo, na qual entradas em branco significam $\emptyset$.

Entrada:	0			1			ε		
Pilha:	0	$	ε	0	$	ε	0	$	ε
q_1									$\{(q_2, \$)\}$
q_2			$\{(q_2, 0)\}$	$\{(q_3, \varepsilon)\}$					
q_3				$\{(q_3, \varepsilon)\}$				$\{(q_4, \varepsilon)\}$	
q_4									