Gramáticas

Teoria da Computação

Professor: Rennan Dantas

Universidade Federal do Ceará
Campus de Crateús

18 de abril de 2023

[0] Slides baseados no livro LEWIS, Harry R.; PAPADIMITRIOU, Christos H. Elements of the Theory of Computation. ACM SIGACT News, v. 29, n. 3, p. 62-78, 1998.

Introdução

- Máquinas de Turing e suas extensões agem basicamente como reconhecedores de linguagens
- Agora vamos conhecer dispositivos geradores de linguagens: expressões regulares e linguagens livres de contexto são exemplos
- Esses dois mecanismos proveem uma caracterização alternativa às classes de linguagens definidas pelos reconhecedores de linguagem

Gramática livre de contexto

- Alfabeto V:
  - Símbolos terminais Σ
  - Símbolos não terminais $V - \Sigma$
- Conjunto finito de regras da forma $A \to u$: A não terminal e $u \in V^*$
- Inicia a operação por um não terminal S e repetidamente substitui o que está no lado esquerdo das regras pelo que está no lado direito até que nenhum outra substituição possar ser feita

Gramática irrestrita

- Na gramática irrestrita, as mesmas convenções são aplicadas exceto que o lado esquerdo das regras não precisam consistir de um único não terminal
- Ao invés disso, o lado esquerdo das regras podem consistir em qualquer string de terminais e não terminais contendo pelo menos um não terminal
- O produto final é uma string contendo apenas terminais como nas gramáticas livres de contexto

Gramáticas

Definição

Uma gramática irrestrita é uma quádrupla $G = (V, \Sigma, R, S)$ onde

- V é um alfabeto;
- $\Sigma \subseteq V$ é o conjunto de símbolos terminais e $V - \Sigma$ é chamado de conjunto de símbolos não terminais
- $S \in V - \Sigma$ é o símbolo inicial
- R é o conjunto das regras, um subconjunto finito de $(V^*(V - \Sigma)V^*)xV^*$

Definição

- Escrevemos $u \to v$ se $(u, v) \in R$
- Escrevemos $u \Rightarrow_G v$ se e somente se, para algum $w_1, w_2 \in V^*$ e alguma regra $u' \to v' \in R$, $u = w_1 u' w_2$ e $v = w_1 v' w_2$
- Como usual, $\Rightarrow_G^*$ é reflexivo, fechamento transitivo de $\Rightarrow_G$
- Uma string $w \in \Sigma^*$ é gerada por G se e somente se $S \Rightarrow_G^* w$
- $L(G)$, a linguagem gerada por G é o conjunto de todas as strings em Σ^* geradas por G.
- Derivação é a sequência da forma $w_0 \Rightarrow_G w_1 \Rightarrow_G \dots \Rightarrow_G w_n$

Exemplo

$$
\begin{aligned}
V &= \{S, a, b, c, A, B, C, T_a, T_b, T_c\},\\
\Sigma &= \{a, b, c\}, \text{and}\\
R &= \{S \to ABCS,\\
&\qquad S \to T_c,\\
&\qquad CA \to AC,\\
&\qquad BA \to AB,\\
&\qquad CB \to BC,\\
&\qquad CT_c \to T_c c,\\
&\qquad CT_c \to T_b c,\\
&\qquad BT_b \to T_b b,\\
&\qquad BT_b \to T_a b,\\
&\qquad AT_a \to T_a a,\\
&\qquad T_a \to e\}.
\end{aligned}
$$

Figura: Fonte: LEWIS, Harry R.; PAPADIMITRIOU, Christos H. Elements of the Theory of Computation. ACM SIGACT News, v. 29, n. 3, p. 62-78, 1998.

Gramáticas

Teorema

Uma linguagem é gerada por uma gramática se e somente se ela é recursivamente enumerável.

Ideia da prova - Ida

- Seja G uma gramática
- Devemos encontrar uma máquina de Turing M que semidecide a linguagem gerada por G
- Desenvolveremos uma máquina não-determinística com:
  - Uma fita para armazenar a entrada original w
  - Uma fita para tentar reconstruir a derivação de w a partir do símbolo inicial da gramática G
  - M procede em passos não determinísticos
  - Exemplo de uma regra $u \to v$: o cabeçote para não deterministicamente sobre um símbolo e verifica se os próximos $|u|$ símbolos formam u. Se sim, apaga-se u e escreve-se w no lugar fazendo o shift apropriado. Caso contrário, a computação entra em loop.

Teorema

Uma linguagem é gerada por uma gramática se e somente se ela é recursivamente enumerável.

Ideia da prova - Volta

- Seja M uma máquina de Turing que semidecide uma linguagem L
- Devemos construir uma gramática irrestrita G que gera a linguagem L semidecidida por M
- Intuitivamente, as derivações de G irão simular as computações de M de maneira reversa
- Exemplo: Se $\delta(q, a) = (p, b)$, então G tem a regra $bp \rightarrow aq$
- Exemplo: Se $\delta(q, a) = (p, \rightarrow)$, então G tem a regra $abp \rightarrow aqb$

Definição

Seja $G = \{V, \Sigma, R, S\}$ uma gramática irrestrita e seja $f : \Sigma^* \to \Sigma^*$ uma função. Dizemos que G computa f se para todos $w, v \in \Sigma^*$ temos:

$$SwS \Rightarrow_G^* v \text{ se e somente se } v = f(w)$$

A função $f : \Sigma^* \to \Sigma^*$ é chamada gramaticalmente computável se e somente se existe uma gramática G que a computa

Teorema

A função $f : \Sigma^* \rightarrow \Sigma^*$ é recursiva se e somente se é gramaticalmente computável

Próxima Aula

O que vem por aí?

- Funções numéricas

Gramáticas

Teoria da Computação

Professor: Rennan Dantas

Universidade Federal do Ceará
Campus de Crateús

18 de abril de 2023

[0] Slides baseados no livro LEWIS, Harry R.; PAPADIMITRIOU, Christos H. Elements of the Theory of Computation. ACM SIGACT News, v. 29, n. 3, p. 62-78, 1998.