

1 Gramáticas e Linguagens Livres de Contexto

Fundamental importância em Ciência da Computação, pois: [MENEZES, 2005]

- Compreende um universo mais amplo de linguagens (comparativamente com o das regulares), tratando adequadamente, questões como parênteses balanceados, construções bloco-estruturadas, entre outras, típicas de linguagens de programação como Pascal, C, Java, etc.;
- Os algoritmos reconhedores e geradores que implementam as linguagens livres de contexto são relativamente simples e possuem uma eficiência razoável;
- Exemplos típicos de aplicações dos conceitos e resultados referentes às linguagens livres do contexto são centrados em *linguagens artificiais* e, em especial, nas linguagens de programação. Em particular, destacam-se *analísadores sintáticos, tradutores de linguagens e processadores de texto* em geral.

Pela hierarquia de Chomsky, a classe das gramáticas livres de contextos contém a classe das Linguagens Regulares. Assim uma linguagem regular pode ser representada por uma gramática regular, mas o contrário não é válido.

1.1 Exemplo Informal

Palíndromo é um string lido da mesma forma, da esquerda para a direita ou da direita para esquerda, como *radar* ou *roma é amor*, ou w é palíndromo se e somente se $w = w^R$.

Exemplo: para o alfabeto $\{0,1\}$, as strings 0110, 11011 e ϵ são palíndromos, mas 011 ou 0101 não são.

Note que se w é um palíndromo, então $0w0$ e $1w1$ também são palíndromos, uma gramática livre de contexto é uma notação formal para expressar tais definições recursivas de linguagens. Uma gramática consiste em uma ou mais variáveis que representam classes de strings, isto é, linguagens.

Uma gramática livre de contexto para os palíndromos:

1. $P \rightarrow \epsilon$
2. $P \rightarrow 0$
3. $P \rightarrow 1$
4. $P \rightarrow 0P0$
5. $P \rightarrow 1P1$

1.2 Definição

Há quatro componentes importantes em uma descrição gramatical de uma linguagem: [HOPCROFT, 2004]

1. Existe um conjunto finito de símbolos que formam os strings da linguagem que está sendo definida. Em nosso exemplo esse conjunto era $\{0,1\}$. Chamamos esse alfabeto de *terminais* ou de *símbolos terminais*.
2. Existe um conjunto finito de *variáveis*, também chamada às vezes *não terminais* ou *categorias sintáticas*. Cada variável representa uma linguagem; isto é, um conjunto de strings. Em nosso exemplo anterior, havia apenas uma variável P .
3. Uma das variáveis representa a linguagem que está sendo definida; ela é chamada de *símbolo de início*. Outras variáveis representam classes auxiliares de strings que são usadas para ajudar a definir a linguagem do símbolo de início. Em nosso exemplo, P (a única variável) é o símbolo de início.
4. Existe um conjunto finito de *produções* ou *regras* que representam a definição recursiva de uma linguagem. Cada produção consiste em:
 - a. Uma variável que está sendo (parcialmente) definida pela produção. Essa variável frequentemente é chamada de *cabeça (head)* da produção.
 - b. O símbolo de produção \rightarrow .
 - c. Uma string e zero ou mais terminais e variáveis. Essa string, chamado de *corpo* da produção, representa um modo de formar strings na linguagem da variável da cabeça. Para fazê-lo, deixamos os terminais inalterados e substituímos cada variável do corpo por qualquer string conhecido por estar na linguagem dessa variável.

Esses quatro componentes formam uma *gramática livre de contexto*, simplesmente *gramática*, ou ainda a sigla GLC ou CFG (do inglês). Assim uma CFG G é representada por $G = (V,T,P,S)$, onde V é o conjunto de variáveis, T é o conjunto de terminais, P é o conjunto de produções e S , o símbolo de início.

1.3 Derivações que utilizam uma gramática

Abordagem mais convencional: do corpo para a cabeça: tomam-se os strings que sabemos estarem na linguagem de cada uma das variáveis do corpo, concatenamos esses strings na ordem apropriada com quaisquer terminais que apareçam no corpo e

inferimos que a string resultante está na linguagem da variável que aparece na cabeça. Esse procedimento é chamado *referência recursiva*.

Uma outra abordagem é definir a linguagem para uma gramática, na qual usamos as produções da cabeça para o corpo, expandimos o símbolo da cabeça usando uma de suas produções. Expandimos ainda mais o string resultante substituindo uma de suas variáveis pelo corpo de uma de suas produções e assim sucessivamente, até derivarmos uma string consistindo inteiramente de terminais. A linguagem da gramática é formada por todas as strings de terminais que puderem ser obtidos dessa maneira. Esse procedimento é chamado *derivação*.

Para restringir o número de escolhas na hora de derivar uma string, podemos exigir, que em cada etapa, a variável mais à esquerda ou mais à direita seja a substituída por um de seus corpos de produção. Essas derivações são chamadas de *derivação mais à esquerda* e *derivação mais à direita* respectivamente.

1.4 Linguagem de uma gramática

Se $G=(V,T,P,S)$ é uma GLC, a *linguagem* de G , denotada por $L(G)$, é o conjunto de strings terminais que têm derivações desde o símbolo de início.

Se uma linguagem L é a linguagem de alguma gramática livre de contexto, então L é dita uma *linguagem livre de contexto*, ou LLC (ou ainda CFL do inglês).