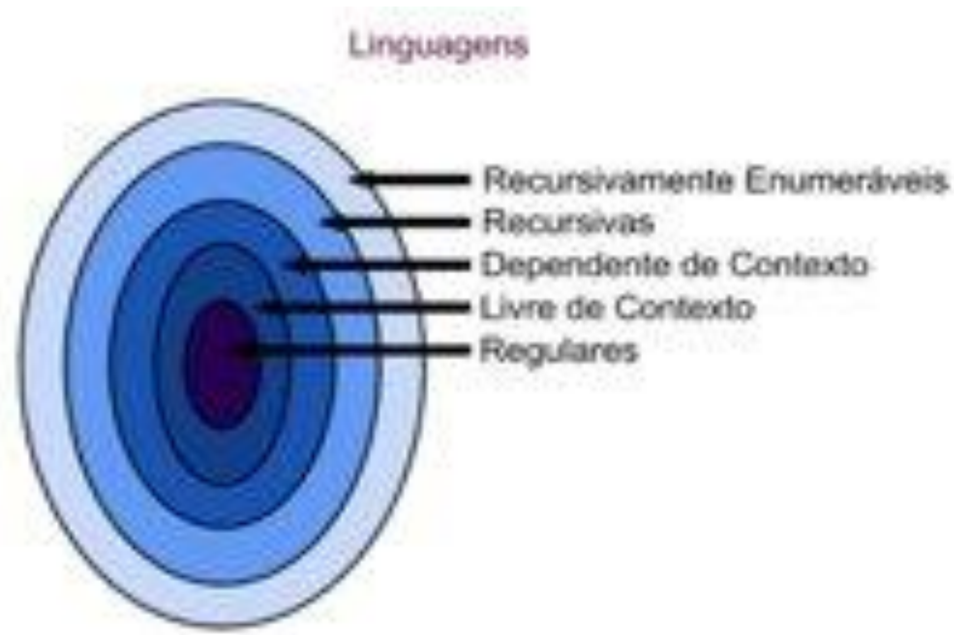
Lógica Computacional

Tema: Linguagens e Gramáticas

Prof. Dr. Ricardo Luis de Azevedo da Rocha

Hierarquia de Chomsky

- As linguagens podem ser classificadas segundo os tipos (linguagens são conjuntos):
  - Tipo 0 (Recursivamente Enumeráveis)
  - Tipo 1 (Sensíveis ao Contexto)
  - Tipo 2 (Livres de Contexto)
  - Tipo 3 (Regulares)
  - As recursivas compõem uma subclasse dentro das de tipo 0, para as quais existe algoritmo de reconhecimento.



Gramáticas

- Gramáticas permitem gerar as cadeias das linguagens, e são caracterizadas como quádruplas ordenadas, da seguinte forma:
  - G = (N, T, P, S)
    - V representa o vocabulário da gramática G ($V = N \cup T$)
    - T representa alguns dos elementos de V, que são os símbolos ou átomos da linguagem (símbolos terminais)
    - N representa o conjunto de símbolos não-terminais
    - P representa o conjunto de todas as leis de formação da gramática (que têm a forma $\alpha \rightarrow \beta$)
    - S é um elemento de N, cuja propriedade é ser o símbolo não-terminal que dá início ao processo de geração de sentenças (derivação)

Hierarquia de Chomsky para Gramáticas

- Gramáticas irrestritas (tipo 0): Nenhuma limitação é imposta, as regras de derivação são do tipo: $\alpha \rightarrow \beta$, com $\alpha \in V^* N V^*$ e $\beta \in V^*$. Ex:
  - G=({A,B,C}, {a,b}, {A→BC, BC→CBA, BA→b, C→a}, A)
- Gramáticas sensíveis ao contexto (tipo 1): impõe-se a restrição de que nenhuma substituição possa reduzir o comprimento da forma sentencial que se está aplicando, assim: $\alpha \rightarrow \beta$, com $|\alpha| \leq |\beta|$, onde $\alpha \in V^* N V^*$ e $\beta \in V^*$ (na verdade G é crescente).
  - Ex: G=({A,B,C,a,b}, {a,b}, {A→AB, AB→AC, C→aba}, A)
  - Obs. A dependência de contexto é definida pela existência de regras da forma uAv→uxv, ou seja, A é reescrito por x em presença do contexto u, v.

Hierarquia de Chomsky

- Gramáticas livres de contexto (tipo 2): impõe-se mais uma restrição às regras, tornando-as da forma: $A \rightarrow \alpha$, onde $A \in N$, $\alpha \in V^*$.
  - Ex: G=({A,B,C,a,b}, {a,b}, {A$\rightarrow$BC, B$\rightarrow$CB, B$\rightarrow$b, C$\rightarrow$a}, A)
- Gramáticas regulares (tipo 3): impõe-se mais uma restrição às regras, tornando-as da forma: $A \rightarrow \alpha B$ ou (XOR) $A \rightarrow B\alpha$, e $A \rightarrow \alpha$, onde $\alpha \in T^*$; $A, B \in N$.
  - Ex: G=({S,a,b}, {a,b}, {S$\rightarrow$aS}, S$\rightarrow$b}, S)

Exemplos

- $L_1 = a^n b^n$ é Linguagem Livre de Contexto
  - $G_1=(\{S\},\{a,b\},P_1,S)$
  - $P_1=\{S \rightarrow ab, S \rightarrow aSb\}$
    - G_1 é de que tipo?
  - $G_2=(\{S,A,B\},\{a,b\},P_2,\{S\})$
  - $P_2=\{S \rightarrow ab, S \rightarrow aA, A \rightarrow Bb, A \rightarrow b, B \rightarrow aA\}$
    - G_2 é de que tipo?

Qual é a diferença entre as duas gramáticas?

Derivação

- Define-se $u \Rightarrow v$ como uma reescrita do termo u pelo termo v (ou um passo de derivação):
- $u \Rightarrow_G v \Leftrightarrow \exists x, y \in V^*, \alpha \rightarrow \beta \in P \mid u = x\alpha y, v = x\beta y$
- Uma derivação é uma sequência de passos da forma $u \Rightarrow_G^* v$
- $S_G = \{u \in V^* \mid S \Rightarrow_G^* u\}$,
- $L(G) = \{w \in T^* \mid S \Rightarrow_G^* w\}$, ou (outra forma) $L(G) = S_G \cap T^*$

Exemplo - Derivação

- Linguagem $a^n b^n c^n$, $n > 0$.
- $G=(\{S,B,C\},\{a,b,c\},P,S)$
- $P=\{(1) S \rightarrow aBC,\ (2) S \rightarrow aSBC,\ (3) CB \rightarrow BC,\ (4) aB \rightarrow ab,\ (5) bB \rightarrow bb,\ (6) bC \rightarrow bc,\ (7) cC \rightarrow cc\}$

$S \Rightarrow_{(1)} aBC \Rightarrow_{(4)} abC \Rightarrow_{(6)} abc$

$S \Rightarrow_{(2)} aSBC \Rightarrow_{(1)} aaBCBC \Rightarrow_{(3)} aaBBCC \Rightarrow_{(4)} aabBCC \Rightarrow_{(5)} aabbCC \Rightarrow_{(6)} aabbcC \Rightarrow_{(7)} aabbcc = a^2b^2c^2$

Cadeia Vazia (ε)

- $A \rightarrow \varepsilon \Rightarrow$ Gramática Irrestrita
- <u>Lidando com ε</u>: Seja G uma gramática tal que $\exists\ A \in N$, para o qual $A \rightarrow \varepsilon \in P$, $A \neq S$, e $\varepsilon \notin L(G)$.
- <u>Teorema</u> 1: $\exists\ G' \mid L(G') = L(G)$ e $A \rightarrow \varepsilon \notin P'$
  - <u>Dem</u>: Em P, $\exists\ (B \rightarrow uAv) \mid u,v \in V^* \wedge \{A \rightarrow \varepsilon, A \rightarrow w\} \subseteq P$ seja $P' = P - \{A \rightarrow \varepsilon\} \cup \{B \rightarrow uv \mid B \rightarrow uAv \in P\}$
- Caso a cadeia vazia ε pertença à Linguagem: (Se $\varepsilon \in L(G) \Rightarrow S \rightarrow \varepsilon$)
  - Construir G′ tal que: $N' = N \cup \{S'\}$, $P' = P \cup \{S' \rightarrow u \mid (S \rightarrow u) \in P\}$
  - Em G′, S′ não é recursivo. Utilizando-se o Teorema 1 em G′ da qual a regra $S' \rightarrow \varepsilon$ foi eliminada, gera-se G″ na qual não há regras da forma $A \rightarrow \varepsilon$. Gera-se então G‴ de tal forma que $P''' = P'' \cup \{S' \rightarrow \varepsilon\}$

Gramáticas Sensíveis ao Contexto e Recursividade

- Teorema 2: Toda gramática sensível de contexto na forma descrita uma linguagem recursiva (reconhecida por algoritmo).
  - Dem.: Define-se o seguinte algoritmo cujas entradas são a cadeia w a ser verificada e a gramática G sem regras vazias.
  - IF ($w = \varepsilon$) THEN $\varepsilon \in L(G)$? ($\exists\ S \rightarrow \varepsilon$?)
  - ELSE $|w| > 0$
  - Construir conjuntos T_i contendo formas sentenciais α geradas por G | $|\alpha| \leq |w|$ até que $T_{i+1} = T_i$
    - $T_0 = \{S\}$
    - $T_{i+1} = T_i \cup \{\beta \in V^+ \mid \alpha \Rightarrow \beta,\ \alpha \in T_i \land |\beta| \leq |w|\}$

Continuando a Demonstração

- Como a linguagem é crescente, a quantidade de cadeias geradas é finita e limitada a $t = (1 + k + k^2 + ... + k^{|w|})$, onde $k = |V|$. Assim o algoritmo termina em número máximo de passos limitado por $(k^{|w|+1} + 1)/(k-1)$. QED

- Ex.: $a^2bc \in L(G)$?
  - $T_0 = \{S\}$
  - $T_1 = \{aBC, aSBC$ (já possui tamanho > 4 ⇒ não pode ser utilizada), $S\}$
  - $T_2=\{abcS\}\cup T_1$
  - $T_3=\{abc\}\cup T_2$
  - $T_4=T_3$.
  - Como $a^2bc \notin T3 \Rightarrow a^2bc \notin L(G)$

Operações Regulares

- Teorema 3: A classe de linguagens recursivamente enumeráveis é fechada em relação às operações regulares (U, $\circ$, *).
  - Dem.: Sejam $G_i = (N_i, T_i, P_i, S_i)$, i = 1,2 duas gramáticas irrestritas, onde $N1 \cap N2 = \phi$.
    - União: $L(G) = L(G_1) \cup L(G_2)$, $G = (N_1 \cup N_2 \cup \{S\}, T_1 \cup T_2, P_1 \cup P_2 \cup \{S \rightarrow S_1, S \rightarrow S_2, S\}, S)$, $S \notin N_1 \cup N_2$.
    - Concatenação: $L(G) = L(G_1) \circ L(G_2)$, $G = (N_1 \cup N_2 \cup \{S\}, T_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \rightarrow S_1 \circ S_2\}, S)$, $S \notin N_1 \cup N_2$.
    - Estrela de Kleene: $L(G) = L(G_1)^*$, $G = (N_1 \cup \{S\}, T_1, P_1 \cup \{S \rightarrow S_1S, S \rightarrow \varepsilon\}, S)$.
  - Não foi feita qualquer observação sobre a forma das regras das gramáticas, logo vale para todos os tipos.