SCC0217 – Linguagens de programação e compiladores
Exercícios de revisão sobre gramáticas

Guilherme de Abreu Barreto, 12543033
Hélio Nogueira Cardoso, 10310227
Theo da Mota dos Santos, 10691331
Laura Fernandes Camargos, 13692334
Sandy da Costa Dutra, 12544570

25 de março de 2025

Exercícios

1. Crie uma gramática para a linguagem ww^r, onde w é uma cadeia não vazia e w^r é o reverso de w.

   Solução:

   Gramática Formal $G = (V_n, V_t, P, S)$:

   - Não-terminais (V_n): $V_n = \{S\}$
   - Terminais (V_t): $V_t = \{a, b\}$
   - Produções (P):

     $$S \to aSa \mid bSb \mid aa \mid bb$$

     Onde:

     - $S \to aSa$ e $S \to bSb$: geram os casos recursivos
     - $S \to aa$ e $S \to bb$: casos base para strings mínimas
   - Símbolo Inicial: S

Características:

- Gera apenas cadeias de comprimento par
- Todas as cadeias geradas são da forma ww^r
- O comprimento mínimo é 2 (casos base)

Exemplos de derivação:

- Para $w = a$: $ww^r = aa$
  Derivação: $S \Rightarrow aa$
- Para $w = ab$: $ww^r = abba$
  Derivação: $S \Rightarrow aSa \Rightarrow abSba \Rightarrow abba$
- Para $w = aab$: $ww^r = aabbaa$
  Derivação: $S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow aabSbaa \Rightarrow aabbaa$

2. Crie uma gramática livre de contexto que represente o conjunto de expressões algébricas envolvendo os identificadores x e y e as operações de adição, subtração, multiplicação e divisão, além dos parênteses.

   Solução:

$$E \to E + T \mid E - T \mid T$$
$$T \to T * F \mid T/F \mid F$$
$$F \to (E) \mid x \mid y$$

3. Na resposta da questão anterior, existe precedência diferente entre os operadores utilizados? Se sim, fornecer a relação de precedência. Caso não exista precedência diferente, reescrever a gramática de forma que exista uma relação de precedência adequada entre os operadores.

   Solução: Sim, na gramática que apresentamos na resposta anterior existe precedência entre os operadores, seguindo a hierarquia matemática padrão. A gramática está estruturada para refletir a seguinte ordem de precedência:

   - Parênteses: ( ) (maior precedência, resolvidos primeiro)
   - Multiplicação e Divisão: *, / (associativos à esquerda)
   - Adição e Subtração: +, - (associativos à esquerda)

Fizemos isso impondo a precedência através de níveis hierárquicos de não-terminais: F (Fator), T (Termo) e E (Expressão). Caso não houvesse precedência teríamos uma gramática ambígua, por exemplo:

$$E \to E + E \mid E * E \mid (E) \mid x \mid y$$

que permite derivações distintas para `x + y * x`:

- `(x + y) * x` (incorreta, se esperamos prioridade de *)
- `x + (y * x)` (correta)

4. Prove, via construção de gramáticas, que uma linguagem livre de contexto é fechada sobre a operação de união, concatenação e fechamento.

   Solução:

   Em nossas demonstrações iremos assumir um par de linguagens livres de contexto, L_1 e L_2 com correspondentes gramáticas G_1 e G_2, a partir das quais serão construídas as novas gramáticas fechadas sob união ($L_1 \cup L_2$), concatenação ($L_1 \cdot L_2$) e fecho (L_1, também conhecido como *Estrela de Kleene* ou conjunto infinitamente contável).

   - Seja G_1 definido por $G_1 = (V_1, \sum_1, R_1, S_1)$, sendo:
     - V_1 um conjunto de símbolos não terminais;
     - $\sum_1$ um conjunto de símbolos terminais;
     - R_1 um conjunto de regras de geração;
     - S_1 o símbolo inicial.
   - e G_2 definido de maneira similar.

   Assumimos, sem perda de generalidade, que V_1 e V_2 são *disjuntos entre si.* Isto pois, doutra forma, podemos simplesmente renomear os símbolos não terminais em uma destas gramáticas para satisfazer esta condição.

   1. Fechamento sob união ($L_1 \cup L_2$)

   Seja uma dada gramática $G_{uniao} = (V_{uniao}, \sum_{uniao}, R_{uniao}, S_{uniao}$ definida tal que:

- $V_{uniao} = V_1 \cup V_2 \cup \{S_{uniao}\}$;
- $\sum_{uniao} = \sum_1 \cup \sum_2$;
- $R_{uniao} = R_1 \cup R_2 \cup \{S_{uniao} \to S_1 \mid S_2\}$;
- S_{uniao} é o novo símbolo inicial

Temos que G_{uniao} é a união das gramáticas G_1 e G_2 e constitui-se enquanto gramática da linguagem L_{uniao}

Prova:

L_{uniao} é equivalente a $L_1 \cup L_2$ *se, e somente se* toda palavra w presente em L_1 ou L_2 está presente em L_{uniao}.

- **Prova direta:**
  - Se a palavra w estiver em L_1: Então existe uma derivação $S_1 \to \cdots \to w$ em G_1. Tido que R_1 é subconjunto de R_{uniao} e $S_{uniao} \to S_1$ está em R_{uniao}, possuímos uma derivação $S_{uniao} \to S_1 \to \cdots \to w$ em G_{uniao}. Logo, w está em L_{uniao};
  - De maneira análoga, se w estiver em L_2, também estará em L_{uniao}. $\square$
- **Prova da recíproca:**
  - Se a palavra w está em L_{uniao}: então existe uma derivação $S_{uniao} \to \cdots \to w$ em G_{uniao}. Tido que a derivação imediata de S_{uniao} é $S_{uniao} \to S_1 \mid S_2$, se w estiver em S_1, o restante das derivações seguem regras descritas em R_1 e portanto w tem de estar em L_1; Doutra forma está em S_2 e de maneira análoga estará em L_2. $\blacksquare$

2. Fechamento sobre concatenação ($L_1 \cdot L_2$):

Seja uma dada gramática $G_{concat} = (V_{concat}, \sum_{concat}, R_{concat}, S_{concat}$ definida tal que:

- $V_{concat} = V_1 \cup V_2 \cup \{S_{concat}\}$;
- $\sum_{concat} = \sum_1 \cup \sum_2$;
- $R_{concat} = R_1 \cup R_2 \cup \{S_{concat} \to S_1 S_2\}$;
- S_{concat} é o novo símbolo inicial

Temos que G_{concat} é a concatenação das gramáticas G_1 e G_2 e constitui-se enquanto gramática da linguagem L_{concat}

- **Prova direta:** se a palavra w está em $L_1 \cdot L_2$, então $w = xy$, onde x encontra-se em L_1 e y encontra-se em L_2. Isto é, existem derivações tais que $S_1 \to \cdots \to x$ em G_1 e $S_2 \to \cdots \to y$ em G_2. Em G_{concat} tem-se que $S_{concat} \to S_1S_2 \to xS_2 \to xy = w$. Logo w está em L_{concat}. $\square$
- Prova da recíproca: se a palavra w está em L_{concat} há uma derivação tal que $S_{concat} \to w$ Como $S_{concat} \to S_1S_2$ e S_1 segue as regras de derivação estabelecidas em R_1, tem-se que $w = xS_2$. O análogo se segue para S_2, R_2 e y. Portanto $w = xy$ e w está em $L_1 \cdot L_2$. $\blacksquare$

Fechamento sob estrela de Kleene (L^*):

Seja uma dada gramática $G^* = (V^*, \sum^*, R^*, S^*$ definida tal que:

- $V^* = V_1 \cup \{S^*\}$;
- $\sum^* = \sum_1$;
- $R^* = R_1 \cup \{S^* \to S_1S^* \mid \lambda\}$;
- S^* é o novo símbolo inicial

Temos que G é a concatenação das gramáticas G_1 uma ou mais vezes e constitui-se enquanto gramática da linguagem L

Prova

Se a palavra w está em L^*, então ou w é λ, ou então uma concatenação de um ou mais caracteres terminais de L_1 (isto é, $w = x_1x_2 \ldots x_n$, onde cada x_i está em L_1). Se $w = \lambda$, temos que $S^* \to \lambda$ é uma derivação. Senão, podemos derivar w infinitamente pelo uso repetido da regra $S \to S_1S^*$, gerando $S_1S_1 \ldots S_1S^*$, com cada S_1 leavando a um x_i correspondente e $S^* \to \lambda$ $\blacksquare$.

5. Sobre gramáticas em geral, considere as seguintes questões:

   a) Forneça a definição de gramática.

   Solução: É uma maneira de listar de forma finita uma linguagem (finita ou infinita). Uma definição formal de gramática é uma tupla $G = (V_n, V_t, P, S)$: onde:

- V_n: conjunto de símbolos não terminais da gramática
- V_t: conjunto de símbolos terminais da gramática, os quais constituem as sentenças da linguagem, com $V_n \cap V_t = \emptyset$
- P: regra de produção, responsáveis por produzir as sentenças da linguagem
- S: símbolo inicial da gramática, por onde se começa a derivação de sentenças

b) Baseado em a), forneça uma gramática para a linguagem $0^n 1^{2n} 0^m$, para n e m estritamente positivos.

Solução:

Gramática Formal $G = (V_n, V_t, P, S)$:

- Não-terminais (V_n): $V_n = \{S, A, B\}$
  - S: Símbolo inicial
  - A: Gera o padrão $0^n 1^{2n}$
  - B: Gera o padrão 0^m
- Terminais (V_t): $V_t = \{0, 1\}$
- Produções (P):

$$S \to AB \quad \text{(Combina as partes fixas)}$$
$$A \to 0A11 \mid 011 \quad \text{(Gera } 0^n 1^{2n} \text{ com } n \geq 1\text{)}$$
$$B \to 0B \mid 0 \quad \text{(Gera } 0^m \text{ com } m \geq 1\text{)}$$

- Símbolo Inicial: S

c) Classifique sua gramática na hierarquia de Chomsky.

Solução: A gramática é livre de contexto (Tipo 2), pois todas as produções são da forma $A \to \alpha$ com A um não-terminal e α uma cadeia de terminais e não-terminais. Sendo assim, no lado esquerdo da regra há apenas um símbolo não-terminal.

6. Escreva uma gramática que represente a linguagem $0^n 1^m$, para:

a) $n < m$

Solução:

A linguagem $L = \{0^n1^m \mid n < m\}$ pode ser gerada pela seguinte gramática:

$$\begin{aligned}
S &\to AB \\
A &\to 0A1 \mid B \\
B &\to 1B \mid 1
\end{aligned}$$

A segunda regra ($A \to 0A1 \mid B$) garante que cada 0 adicionado à esquerda é balanceado com um 1 à direita, ao mesmo tempo, permite que se "escape"para a terceira regra, que coloca mais indefinidos 1 (pelo menos 1), o que garante que a quantidade de 1 será maior que a de 0.

b) $n > m$

Solução:

Para a linguagem $L = \{0^n1^m \mid n > m\}$ a gramática é semelhante:

$$\begin{aligned}
S &\to AB \\
B &\to 0B1 \mid A \\
A &\to 0A \mid 0
\end{aligned}$$

A segunda regra balanceia os 0s e 1s, até que se "escape"para a regra 3, a qual coloca 0s de maneira indefinida (pelo menos 1).

7. Qual gramática é utilizada para descrever linguagens de programação. Justifique sua resposta.

   Solução: Cada fase da análise em compiladores (léxica, sintática e semântica) lida com diferentes aspectos da linguagem de programação e se utiliza de diferentes formalismos. A análise léxica, que identifica palavras-chave e identificadores, é feita com Linguagens Regulares, já a sintática, que verifica a hierarquia do código, é feita com Linguagens Livres de Contexto. Por fim, a análise semântica, que avalia o significado das expressões, é feita com Linguagens Sensíveis ao Contexto.

8. Dada uma linguagem livre de contexto L, é possível dizer que $L - \{\lambda\}$ também é livre de contexto?

   Solução: Sim, a classe das linguagens livres de contexto é fechada sob a operação de diferença com um conjunto finito. Portanto, se L é livre de contexto, então $L - \{\lambda\}$ também é livre de contexto.

Provando: Seja $G = (V_n, V_t, P, S)$ uma GLC que gera L. Podemos construir uma gramática G' para $L - \{\lambda\}$ da seguinte forma:

(a) **Caso 1:** Se $\lambda \notin L$, então $G' = G$ (nada precisa ser feito)

(b) **Caso 2:** Se $\lambda \in L$, construímos $G' = (V_n \cup \{S'\}, V_t, P', S')$ onde:

- S' é um novo símbolo inicial ($S' \notin V_n$)
- $P' = P \cup \{S' \to \alpha \mid S \to \alpha \in P \text{ e } \alpha \neq \lambda\}$
- Removemos todas as produções que derivam diretamente λ do novo símbolo inicial