Linguagens Formais e Autómatos / Compiladores

Gramáticas independentes do contexto (GIC)

Artur Pereira <artur@ua.pt>,
Miguel Oliveira e Silva <mos@ua.pt>

DETI, Universidade de Aveiro

Ano letivo de 2020-2021

Sumário

Gramáticas
Definição

Uma gramática é um quádruplo $G = (T, N, P, S)$, onde

- T é um conjunto finito não vazio de símbolos **terminais**;
- N, com $N \cap T = \emptyset$, é um conjunto finito não vazio de símbolos **não terminais**;
- P é um conjunto de **produções** (ou regras de rescrita), cada uma da forma $\alpha \to \beta$;
- $S \in N$ é o símbolo inicial.

- α e β são designados por **cabeça da produção** e **corpo da produção**, respetivamente.
- No caso geral $\alpha \in (N \cup T)^* \times N \times (N \cup T)^*$ e $\beta = (N \cup T)^*$.
- Em ANTLR:
  - os terminais são representados por ids começados por letra maíscula
  - os não terminais são representados por ids começados por letra minúscula

Gramáticas independentes do contexto – GIC

$\mathcal{D}$ Uma gramática $G = (T, N, P, S)$ diz-se **independente do contexto** (ou **livre de contexto**) se, para qualquer produção $(\alpha \to \beta) \in P$, as duas condições seguintes são satisfeitas

$$\begin{array}{lcl} \alpha & \in & N \\ \beta & \in & (T \cup N)^* \end{array}$$

- A linguagem gerada por uma gramática independente do contexto diz-se independente do contexto
- as gramáticas regulares são independentes do contexto
- As gramáticas independentes do contexto são fechadas sob as operações de reunião, concatenação e fecho
  - mas não o são sob as operações de intersecção e complementação.

- Note que: se $\beta \in T^* \cup T^* N$, então $\beta \in (T \cup N)^*$

Derivação

Exemplo

$\mathcal{Q}$ Considere, sobre o alfabeto $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, a gramática

$$S \rightarrow \varepsilon \mid \mathtt{a}\ B \mid \mathtt{b}\ A \mid \mathtt{c}\ S$$
$$A \rightarrow \mathtt{a}\ S \mid \mathtt{b}\ A\ A \mid \mathtt{c}\ A$$
$$B \rightarrow \mathtt{a}\ B\ B \mid \mathtt{b}\ S \mid \mathtt{c}\ B$$

e transforme o símbolo inicial S na palavra $\mathtt{aabcbc}$ por aplicação sucessiva de produções da gramática

$\mathcal{R}$

$$S \Rightarrow \mathtt{a}B \Rightarrow \mathtt{aa}BB \Rightarrow \mathtt{aab}SB \Rightarrow \mathtt{aabc}SB \Rightarrow \mathtt{aabc}B \Rightarrow \mathtt{aabcb}S \Rightarrow \mathtt{aabcbc}S \Rightarrow \mathtt{aabcbc}$$

- Acabou de se obter uma **derivação à esquerda** da palavra $\mathtt{aabcbc}$
- Cada passo dessa derivação é uma **derivação direta à esquerda**

- Quando há dois ou mais símbolos não terminais, opta-se por expandir primeiro o mais à esquerda

Derivação
Definições

$\mathcal{D}$ Dada uma palavra $\alpha A\beta$, com $A \in N$ e $\alpha, \beta \in (N \cup T)^*$, e uma produção $(A \to \gamma) \in P$, com $\gamma \in (N \cup T)^*$, chama-se **derivação direta** à rescrita de $\alpha A\beta$ em $\alpha\gamma\beta$, denotando-se

$$\alpha A\beta \Rightarrow \alpha\gamma\beta$$

$\mathcal{D}$ Dada uma palavra $\alpha A\beta$, com $A \in N$, $\alpha \in T^*$ e $\beta \in (N \cup T)^*$, e uma produção $(A \to \gamma) \in P$, com $\gamma \in (N \cup T)^*$, chama-se **derivação direta à esquerda** à rescrita de $\alpha A\beta$ em $\alpha\gamma\beta$, denotando-se

$$\alpha A\beta \overset{E}{\Rightarrow} \alpha\gamma\beta$$

$\mathcal{D}$ Dada uma palavra $\alpha A\beta$, com $A \in N$, $\alpha \in (N \cup T)^*$ e $\beta \in T^*$, e uma produção $(A \to \gamma) \in P$, com $\gamma \in (N \cup T)^*$, chama-se **derivação direta à direita** à rescrita de $\alpha A\beta$ em $\alpha\gamma\beta$, denotando-se

$$\alpha A\beta \overset{D}{\Rightarrow} \alpha\gamma\beta$$

Derivação
Definições

$\mathcal{D}$ Chama-se **derivação** a uma sucessão de zero ou mais derivações diretas, denotando-se

$$\alpha \Rightarrow^* \beta \qquad \equiv \qquad \alpha = \gamma_0 \Rightarrow \gamma_1 \Rightarrow \cdots \Rightarrow \gamma_n = \beta$$

onde n é o comprimento da derivação.

$\mathcal{D}$ Chama-se **derivação à esquerda** a uma sucessão de zero ou mais derivações diretas à esquerda, denotando-se

$$\alpha \stackrel{E}{\Rightarrow}{}^* \beta \qquad \equiv \qquad \alpha = \alpha_0 \stackrel{E}{\Rightarrow} \alpha_1 \stackrel{E}{\Rightarrow} \cdots \stackrel{E}{\Rightarrow} \alpha_n = \beta$$

onde n é o comprimento da derivação.

$\mathcal{D}$ Chama-se **derivação à direita** a uma sucessão de zero ou mais derivações diretas à direita, denotando-se

$$\alpha \stackrel{D}{\Rightarrow}{}^* \beta \qquad \equiv \qquad \alpha = \gamma_0 \stackrel{D}{\Rightarrow} \gamma_1 \stackrel{D}{\Rightarrow} \cdots \stackrel{D}{\Rightarrow} \gamma_n = \beta$$

onde n é o comprimento da derivação.

Derivação
Exemplo

$\mathcal{Q}$ Considere, sobre o alfabeto $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, a gramática seguinte

$$
\begin{aligned}
S &\rightarrow \varepsilon \mid \mathtt{a}\ B \mid \mathtt{b}\ A \mid \mathtt{c}\ S \\
A &\rightarrow \mathtt{a}\ S \mid \mathtt{b}\ A\ A \mid \mathtt{c}\ A \\
B &\rightarrow \mathtt{a}\ B\ B \mid \mathtt{b}\ S \mid \mathtt{c}\ B
\end{aligned}
$$

Determine as derivações à esquerda e à direita da palavra $\mathtt{aabcbc}$

$\mathcal{R}$

à esquerda

$$
\begin{aligned}
S &\Rightarrow \mathtt{a}B \Rightarrow \mathtt{aa}BB \Rightarrow \mathtt{aab}SB \Rightarrow \mathtt{aabc}SB \\
&\Rightarrow \mathtt{aabc}B \Rightarrow \mathtt{aabcb}S \Rightarrow \mathtt{aabcbc}S \Rightarrow \mathtt{aabcbc}
\end{aligned}
$$

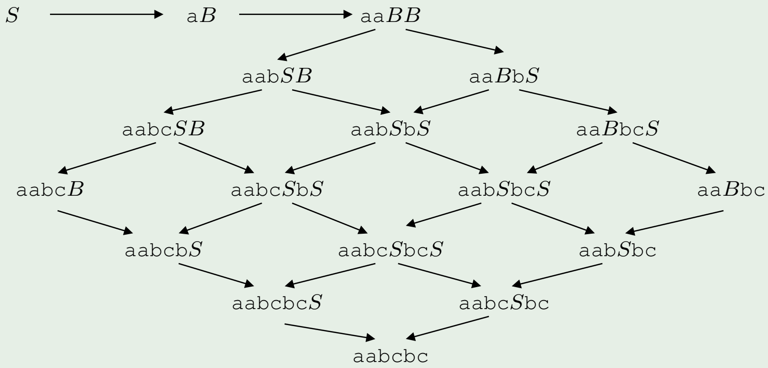
à direita

$$
\begin{aligned}
S &\Rightarrow \mathtt{a}B \Rightarrow \mathtt{aa}BB \Rightarrow \mathtt{aa}B\mathtt{b}S \Rightarrow \mathtt{aa}B\mathtt{bc}S \\
&\Rightarrow \mathtt{aa}B\mathtt{bc} \Rightarrow \mathtt{aab}S\mathtt{bc} \Rightarrow \mathtt{aabc}S\mathtt{bc} \Rightarrow \mathtt{aabcbc}
\end{aligned}
$$

- Note que se usou $\Rightarrow$ em vez de $\overset{D}{\Rightarrow}$ e $\overset{E}{\Rightarrow}$

Derivação
Alternativas de derivação

- O grafo seguinte capta as alternativas de derivação. Considera-se novamente a palavra `aabcbc` e a gramática anterior

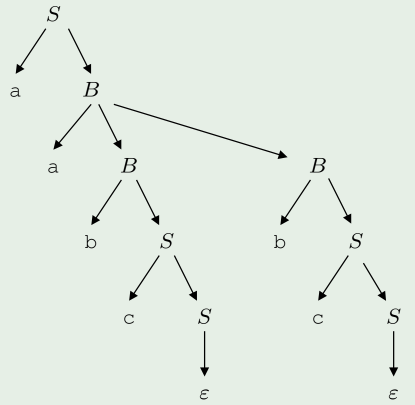


- Identifique os caminhos que correspondem às derivações à direita e à esquerda

Derivação
Árvore de derivação

$\mathcal{D}$ Uma **árvore de derivação** (*parse tree*) é uma representação de uma derivação onde os nós-ramos são elementos de N e os nós-folhas são elementos de T.

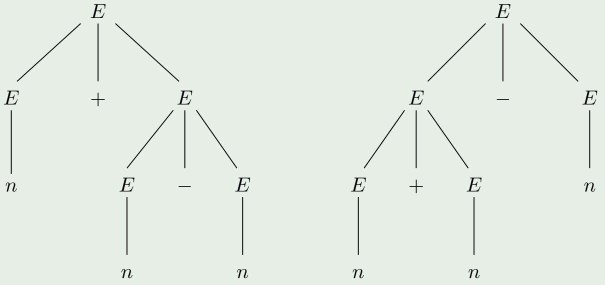
- A árvore de derivação da palavra `aabcbc` na gramática anterior é



Ambiguidade
Ilustração através de um exemplo

- Considere a gramática $S \rightarrow S$ + S | S - S | (S) | n
  e desenhe a árvore de derivação da palavra n+n−n

$\mathcal{R}$ Podem obter-se duas árvores de derivação diferentes



- Há duas interpretações diferentes para a palavra; há **ambiguidade**

Ambiguidade
Definição

- $\mathcal{D}$ Diz-se que uma palavra é derivada **ambiguamente** se possuir duas ou mais árvores de derivação distintas
- $\mathcal{D}$ Diz-se que uma gramática é **ambígua** se possuir pelo menos uma palavra gerada ambiguamente

- Frequentemente é possível definir-se uma gramática não ambígua que gera a mesma linguagem que uma ambígua
- No entanto, há gramáticas **inerentemente ambíguas**

  Por exemplo, a linguagem

  $$L = \{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^k \mid i = j \vee j = k\}$$

  não possui uma gramática não ambígua que a represente.

Ambiguidade
Remoção da ambiguidade

$\mathcal{R}$ Considere-se novamente a gramática

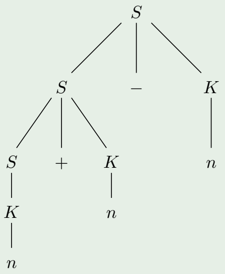
$$S \to S \texttt{ + } S \mid S \texttt{ - } S \mid \texttt{ (} S \texttt{) } \mid \texttt{n}$$

e obtenha-se uma gramática não ambígua equivalente

$\mathcal{R}$

$$S \to K \mid S \texttt{ + } K \mid S \texttt{ - } K$$
$$K \to \texttt{n} \mid \texttt{ (} S \texttt{) }$$

$\mathcal{Q}$ Desenhe a árvore de derivação da palavra `n+n-n` na nova gramática



Projeto de gramáticas
Exemplo #1, solução #1

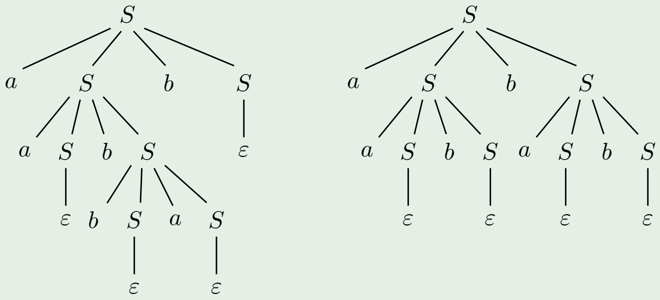
$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\texttt{a}, \texttt{b}\}$, determine uma gramática independente do contexto que represente a linguagem

$$L_1 \;=\; \{\omega \in T^* \;:\; \#(\texttt{a},\omega) = \#(\texttt{b},\omega)\}$$

$\mathcal{R}_1$

$$S \;\rightarrow\; \varepsilon \;|\; \texttt{a}\; S\; \texttt{b}\; S \;|\; \texttt{b}\; S\; \texttt{a}\; S$$

$\mathcal{Q}$ A gramática é ambígua? Analise a palavra `aabbab`



Projeto de gramáticas
Exemplo #1, solução #2

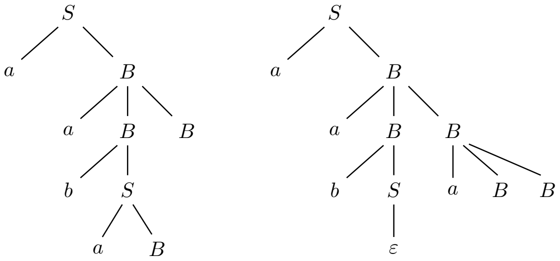
$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}\}$, determine uma gramática independente do contexto que represente a linguagem

$$L_1 \;=\; \{\omega \in T^* \;:\; \#(\mathtt{a}, \omega) = \#(\mathtt{b}, \omega)\}$$

$\mathcal{R}_2$

$$\begin{aligned} S &\to \varepsilon \mid \mathtt{a}\ B \mid \mathtt{b}\ A \\ A &\to \mathtt{a}\ S \mid \mathtt{b}\ A\ A \\ B &\to \mathtt{a}\ B\ B \mid \mathtt{b}\ S \end{aligned}$$

$\mathcal{Q}$ A gramática é ambígua?
Analise a palavra $\mathtt{aababb}$.



- Falta expandir alguns nós

Projeto de gramáticas
Exemplo #1, solução #3

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}\}$, determine uma gramática independente do contexto que represente a linguagem

$$L_1 = \{\omega \in T^* \ : \ \#(\mathtt{a}, \omega) = \#(\mathtt{b}, \omega)\}$$

$\mathcal{R}_3$

$$S \to \varepsilon \mid \mathtt{a}\ B\ S \mid \mathtt{b}\ A\ S$$
$$A \to \mathtt{a} \mid \mathtt{b}\ A\ A$$
$$B \to \mathtt{a}\ B\ B \mid \mathtt{b}$$

$\mathcal{Q}$ A gramática é ambígua? Analise a palavra `aababb`

Projeto de gramáticas
Exemplo #2

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, determine uma gramática independente do contexto que represente a linguagem

$$L_2 = \{\omega \in T^* \ : \ \#(\mathtt{a}, \omega) = \#(\mathtt{b}, \omega)\}$$

$\mathcal{R}$

$$S \to \varepsilon \mid \mathtt{a}\ B\ S \mid \mathtt{b}\ A\ S \mid \mathtt{c}\ S$$
$$A \to \mathtt{a} \mid \mathtt{b}\ A\ A \mid \mathtt{c}\ A$$
$$B \to \mathtt{a}\ B\ B \mid \mathtt{b} \mid \mathtt{c}\ B$$

$\mathcal{Q}$ A gramática é ambígua?

Projeto de gramáticas
Exemplo #3, solução #1

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, determine uma gramática independente do contexto que represente a linguagem

$$L_3 = \{\omega \in T^* \; : \; \#(\mathtt{a}, \omega) = \#(\mathtt{b}, \omega) \wedge \forall_{i \leq |\omega|} \; \#(\mathtt{a}, \text{prefix}(i, \omega)) \geq \#(\mathtt{b}, \text{prefix}(i, \omega))\}$$

$\mathcal{R}_1$

$$S \rightarrow \varepsilon \mid \mathtt{a} \; S \; \mathtt{b} \; S \mid \mathtt{c} \; S$$

$\mathcal{Q}$ A gramática é ambígua? Analise a palavra `aababb`

- Esta linguagem faz-vos lembrar algo que conheçam?
- Solução inspirada na do exemplo 1.1 removendo a produção $S \rightarrow \mathtt{b} \; S \; \mathtt{a} \; S$

Projeto de gramáticas
Exemplo #3: solução #2

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, determine uma gramática independente do contexto que represente a linguagem

$$\begin{aligned} L_3 \ = \ \{\omega \in T^* \ : \ & \#(\mathtt{a}, \omega) = \#(\mathtt{b}, \omega) \wedge \\ & \forall_{i \le |\omega|} \ \#(\mathtt{a}, \text{prefix}(i, \omega)) \ge \#(\mathtt{b}, \text{prefix}(i, \omega))\} \end{aligned}$$

$\mathcal{R}_2$

$$\begin{aligned} S &\rightarrow \varepsilon \mid \mathtt{a} \ B \mid \mathtt{c} \ S \\ B &\rightarrow \mathtt{a} \ B \ B \mid \mathtt{b} \ S \mid \mathtt{c} \ B \end{aligned}$$

$\mathcal{Q}$ A gramática é ambígua? Analise a palavra $\mathtt{aababb}$

- Solução inspirada na do exemplo 1.2 removendo a produção $S \rightarrow \mathtt{b} \ A$ e as começadas por A

Projeto de gramáticas
Exemplo #3: solução #3

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, determine uma gramática independente do contexto que represente a linguagem

$$L_3 = \{\omega \in T^* \ : \ \#(\mathtt{a},\omega) = \#(\mathtt{b},\omega) \wedge \\ \forall_{i \le |\omega|} \ \#(\mathtt{a}, \text{prefix}(i,\omega)) \ge \#(\mathtt{b}, \text{prefix}(i,\omega))\}$$

$\mathcal{R}_3$

$$S \to \varepsilon \mid \mathtt{a} \ B \ S \mid \mathtt{c} \ S$$

$$B \to \mathtt{a} \ B \ B \mid \mathtt{b} \mid \mathtt{c} \ B$$

$\mathcal{Q}$ A gramática é ambígua? Analise a palavra $\mathtt{aababb}$

- Solução inspirada na do exemplo 1.3 removendo a produção $S \to \mathtt{b} \ A \ S$ e as começadas por A

Projeto de gramáticas
Exercício

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{(}, \mathtt{)}, +, \star\}$, determine uma gramática independente do contexto que represente a linguagem

$$L = \{\, \omega \in T^* \; : \; \omega \text{ representa uma expressão regular sobre o alfabeto } \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}\}$$

$\mathcal{R}$ Em ANTLR, poder-se-ia fazer

$$\begin{aligned} S &\to E \\ E &\to E \;{'\star'} \\ &\mid \; E \;\; E \\ &\mid \; E \;{'+'}\; E \\ &\mid \; {'\mathtt{('}} \; E \; {'\mathtt{)}'} \\ &\mid \; {'\mathtt{a}'} \mid {'\mathtt{b}'} \mid {'\mathtt{c}'} \end{aligned}$$

mas em geral não, porque, em geral, as alternativas estão todas ao mesmo nível

- Como escrever a gramática de modo à precedência ser imposta por construção?

- Está a usar-se o operador $+$ em vez do $|$

Projeto de gramáticas
Exercício (cont.)

$\mathcal{R}$ Em geral

$$
\begin{array}{rcl}
S & \rightarrow & E \\
E & \rightarrow & E \ '+' \ T \\
 & | & T \\
T & \rightarrow & T \ F \\
 & | & F \\
F & \rightarrow & F \ '\star' \\
 & | & O \\
O & \rightarrow & '\texttt{(}' \ E \ '\texttt{)}' \\
 & | & '\texttt{a}' \mid '\texttt{b}' \mid '\texttt{c}'
\end{array}
$$

- Uma expressão é vista como uma 'soma' de termos
- Um termo é visto como um 'produto' (concatenação) de fatores
- Um fator é visto como um 'fecho' de operandos
- Um operando ou é um elemento base ou uma expressão entre parêntesis

---

- Está a usar-se o operador + em vez do |

Reunião de GIC
Exemplo

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, determine uma gramática independente do contexto que represente a linguagem

$$L = \{ \omega \in T^* : \#(\mathtt{a},\omega) = \#(\mathtt{b},\omega) \lor \#(\mathtt{a},\omega) = \#(\mathtt{c},\omega) \}$$

$\mathcal{R}$

$L_1 = \{ \omega \in T^* : \#(\mathtt{a},\omega) = \#(\mathtt{b},\omega) \}$	$S_1 \to \varepsilon \mid \mathtt{a}\ S_1\ \mathtt{b}\ S_1$ $\mid \mathtt{b}\ S_1\ \mathtt{a}\ S_1 \mid \mathtt{c}\ S_1$
$L_2 = \{ \omega \in T^* : \#(\mathtt{a},\omega) = \#(\mathtt{c},\omega) \}$	$S_2 \to \varepsilon \mid \mathtt{a}\ S_2\ \mathtt{c}\ S_2$ $\mid \mathtt{b}\ S_2 \mid \mathtt{c}\ S_2\ \mathtt{a}\ S_2$
$L = L_1 \cup L_2$	$S \to S_1 \mid S_2$ $S_1 \to \varepsilon \mid \mathtt{a}\ S_1\ \mathtt{b}\ S_1$ $\mid \mathtt{b}\ S_1\ \mathtt{a}\ S_1 \mid \mathtt{c}\ S_1$ $S_2 \to \varepsilon \mid \mathtt{a}\ S_2\ \mathtt{c}\ S_2$ $\mid \mathtt{b}\ S_2 \mid \mathtt{c}\ S_2\ \mathtt{a}\ S_2$

- Para esta linguagem, mesmo que as gramáticas de L_1 e L_2 não sejam ambíguas, a de L será ambígua. Porquê?

Operações sobre GICs
Reunião

$\mathcal{D}$ Sejam $G_1 = (T_1, N_1, P_1, S_1)$ e $G_2 = (T_2, N_2, P_2, S_2)$ duas gramáticas independentes do contexto quaisquer, com $N_1 \cap N_2 = \emptyset$.

A gramática $G = (T, N, P, S)$ onde

$$\begin{aligned} T &= T_1 \cup T_2 \\ N &= N_1 \cup N_2 \cup \{S\} \quad \text{com} \quad S \notin (N_1 \cup N_2) \\ P &= \{S \to S_1, S \to S_2\} \cup P_1 \cup P_2 \end{aligned}$$

é independente do contexto e gera a linguagem $L = L(G_1) \cup L(G_2)$

- As novas produções $S \to S_i$, com $i = 1, 2$, permitem que G gere a linguagem $L(G_i)$
- Esta definição é idêntica à que foi dada para a operação de reunião nas gramáticas regulares

Concatenação de GIC
Exemplo

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, determine uma gramática independente do contexto que represente a linguagem

$$L = \{\, \omega_1\omega_2 \;:\; \omega_1, \omega_2 \in T^* \;\wedge \#(\mathtt{a}, \omega_1) = \#(\mathtt{b}, \omega_1) \;\wedge\; \#(\mathtt{a}, \omega_2) = \#(\mathtt{c}, \omega_2) \,\}$$

$\mathcal{R}$

$L_1 = \{\, \omega \in T^* \;:\; \#(\mathtt{a}, \omega) = \#(\mathtt{b}, \omega) \,\}$	$S_1 \rightarrow \varepsilon \mid \mathtt{a}\ S_1\ \mathtt{b}\ S_1$ $\mid \mathtt{b}\ S_1\ \mathtt{a}\ S_1 \mid \mathtt{c}\ S_1$
$L_2 = \{\, \omega \in T^* \;:\; \#(\mathtt{a}, \omega) = \#(\mathtt{c}, \omega) \,\}$	$S_2 \rightarrow \varepsilon \mid \mathtt{a}\ S_2\ \mathtt{c}\ S_2$ $\mid \mathtt{b}\ S_2 \mid \mathtt{c}\ S_2\ \mathtt{a}\ S_2$
$L = L_1 \cdot L_2$	$S \rightarrow S_1\ S_2$ $S_1 \rightarrow \varepsilon \mid \mathtt{a}\ S_1\ \mathtt{b}\ S_1$ $\mid \mathtt{b}\ S_1\ \mathtt{a}\ S_1 \mid \mathtt{c}\ S_1$ $S_2 \rightarrow \varepsilon \mid \mathtt{a}\ S_2\ \mathtt{c}\ S_2$ $\mid \mathtt{b}\ S_2 \mid \mathtt{c}\ S_2\ \mathtt{a}\ S_2$

Operações sobre gramáticas:
Concatenação

$\mathcal{D}$ Sejam $G_1 = (T_1, N_1, P_1, S_1)$ e $G_2 = (T_2, N_2, P_2, S_2)$ duas gramáticas independentes do contexto quaisquer, com $N_1 \cap N_2 = \emptyset$.

A gramática $G = (T, N, P, S)$ onde

$$\begin{aligned} T &= T_1 \cup T_2 \\ N &= N_1 \cup N_2 \cup \{S\} \quad \text{com} \quad S \notin (N_1 \cup N_2) \\ P &= \{S \to S_1 S_2\} \cup P_1 \cup P_2 \end{aligned}$$

é independente do contexto e gera a linguagem $L = L(G_1) \cdot L(G_2)$

- A nova produção $S \to S_1 S_2$ justapõe palavras de $L(G_2)$ às de $L(G_1)$
- Esta definição é **diferente** da que foi dada para a operação de concatenação nas gramáticas regulares

Fecho de Kleene de GIC
Exemplo

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, determine uma gramática independente do contexto que represente a linguagem

$$L = \{\omega \in T^* : \#(\mathtt{a}, \omega) \geq \#(\mathtt{b}, \omega)\}$$

$\mathcal{R}$

$X = \{\omega \in T^* : \#(\mathtt{a}, \omega) = \#(\mathtt{b}, \omega)\}$	$X \to \varepsilon \mid \mathtt{a}\ B \mid \mathtt{b}\ A \mid \mathtt{c}\ X$ $A \to \mathtt{a}\ X \mid \mathtt{b}\ A\ A \mid \mathtt{c}\ A$ $B \to \mathtt{a}\ B\ B \mid \mathtt{b}\ X \mid \mathtt{c}\ B$
$A = \{\omega \in T^* : \#(\mathtt{a}, \omega) = \#(\mathtt{b}, \omega) + 1\}$	Basta usar o A anterior como símbolo inicial
$L = X \cup A^*$	$S \to \varepsilon \mid A\ S \mid X$ $X \to \varepsilon \mid \mathtt{a}\ B \mid \mathtt{b}\ A \mid \mathtt{c}\ X$ $A \to \mathtt{a}\ X \mid \mathtt{b}\ A\ A \mid \mathtt{c}\ A$ $B \to \mathtt{a}\ B\ B \mid \mathtt{b}\ X \mid \mathtt{c}\ B$

- O fecho de A inclui a palavra vazia mas não as outras palavras com $\#_a = \#_b$

Operações sobre gramáticas
Fecho de Kleene

Seja $G_1 = (T_1, N_1, P_1, S_1)$ uma gramática independente do contexto qualquer.
A gramática $G = (T, N, P, S)$ onde

$$\begin{aligned} T &= T_1 \\ N &= N_1 \cup \{S\} \quad \text{com} \quad S \notin N_1 \\ P &= \{S \to \varepsilon, S \to S_1 S\} \cup P_1 \end{aligned}$$

é independente do contexto e gera a linguagem $L = (L(G_1))^*$

- A produção $S \to \varepsilon$, per si, garante que $L^0(G_1) \subseteq L(G)$
- As produções $S \to S_1 S$ e $S \to \varepsilon$ garantem que $L^i(G_1) \subseteq L(G)$, para qualquer $i > 0$
- Esta definição é **diferente** da que foi dada para a operação de fecho nas gramáticas regulares

Símbolos produtivos e improdutivos
Exemplo de ilustração

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$, considere a gramática

$$
\begin{aligned}
S &\rightarrow \mathtt{a}\ A\ \mathtt{b} \mid \mathtt{b}\ B \\
A &\rightarrow \mathtt{c}\ C \mid \mathtt{b}\ B \mid \mathtt{d} \\
B &\rightarrow \mathtt{d}\ D \mid \mathtt{b} \\
C &\rightarrow A\ C \mid B\ D \mid S\ D \\
D &\rightarrow A\ D \mid B\ C \mid C\ S \\
E &\rightarrow \mathtt{a}\ A \mid \mathtt{b}\ B \mid \varepsilon
\end{aligned}
$$

- Tente expandir (através de uma derivação) o símbolo não terminal A para uma sequência apenas com símbolos terminais ($S \Rightarrow^* u$, com $u \in T^*$)
  - $A \Rightarrow \mathtt{d}$
- Faça o mesmo com o símbolo C
  - Não consegue
- A é um símbolo **produtivo**; C é um símbolo **improdutivo**

Símbolos produtivos e improdutivos
Definição de símbolo produtivo

- Seja $G = (T, N, P, S)$ uma gramática qualquer
- Um símbolo não terminal A diz-se **produtivo** se for possível expandi-lo para uma expressão contendo apenas símbolos terminais
- Ou seja, A é produtivo se

$$A \Rightarrow^+ u \quad \wedge \quad u \in T^*$$

- Caso contrário, diz-se que A é **improdutivo**
- Uma gramática é improdutiva se o seu símbolo inicial for improdutivo

- Na gramática

$$\begin{aligned} S &\rightarrow \texttt{a b} \mid \texttt{a } S \texttt{ b} \mid X \\ X &\rightarrow \texttt{c } X \end{aligned}$$

  - S é produtivo, porque $S \Rightarrow \texttt{ab} \quad \wedge \quad \texttt{ab} \in T^*$
  - X é improdutivo, porque $X \Rightarrow \texttt{c}X \Rightarrow \texttt{cc}X \Rightarrow^* \texttt{c} \cdots \texttt{c}X$

Símbolos produtivos
Algoritmo de cálculo

- O conjunto dos símbolos produtivos, N_p, pode ser obtido por aplicação sucessiva das seguintes regras construtivas

$$\textbf{if}\ (A \to \alpha) \in P\ \ \textbf{and}\ \alpha \in T^*\ \ \textbf{then}\ A \in N_p$$
$$\textbf{if}\ (A \to \alpha) \in P\ \ \textbf{and}\ \alpha \in (T \cup N_p)^*\ \ \textbf{then}\ A \in N_P$$

- Algoritmo de cálculo:

```
Np ← ∅                                    # no fim, ficará com todos os símbolos produtivos
Pp ← orded(P)                             # lista ordenada de produções a processar
repeat
    nothingAdded ← true
    (A → α) ← headOf(Pp)                  # retira o primeiro elemento de Pp
    if A ∈ Np then                        # A já foi marcado como produtivo
        continue
    else if α ∈ (T ∪ Np)* then            # se todos são terminais ou produtivos
        Np ← Np ∪ {A}                     # então A é produtivo
        nothingAdded ← false
    else
        tailOf(Pp) ← (A → α)              # para ser processado mais tarde
until nothingAdded or Np = N
```

Símbolos acessíveis e inacessíveis
Exemplo de ilustração

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$, considere a gramática

$$
\begin{aligned}
S &\to \mathtt{a}\ A\ \mathtt{b} \mid \mathtt{b}\ B \\
A &\to \mathtt{c}\ C \mid \mathtt{b}\ B \mid \mathtt{d} \\
B &\to \mathtt{d}\ D \mid \mathtt{b} \\
C &\to A\ C \mid B\ D \mid S\ D \\
D &\to A\ D \mid B\ C \mid C\ S \\
E &\to \mathtt{a}\ A \mid \mathtt{b}\ B \mid \varepsilon
\end{aligned}
$$

- Tente alcançar (através de uma derivação) o símbolo não terminal C a partir do símbolo inicial (S) ($S \Rightarrow^* \alpha\, C\, \beta$, com $\alpha, \beta \in (T \cup N)^*$)
  - $S \Rightarrow \mathtt{b}\, B \Rightarrow \mathtt{b}\,\mathtt{d}\, D \Rightarrow \mathtt{b}\,\mathtt{d}\, B\, C$
- Faça o mesmo com o símbolo E
  - Não consegue
- C é um símbolo **acessível**; E é um símbolo **inacessível**

Símbolos acessíveis e inacessíveis
Definição de símbolo acessível

- Seja $G = (T, N, P, S)$ uma gramática qualquer
- Um símbolo terminal ou não terminal x diz-se **acessível** se for possível expandir S (o símbolo inicial) para uma expressão que contenha x
- Ou seja, x é acessível se

$$S \Rightarrow^* \alpha\, x\, \beta$$

- Caso contrário, diz-se que x é **inacessível**

- Na gramática

$$\begin{aligned} S &\rightarrow \varepsilon \mid \mathtt{a}\ S\ \mathtt{b} \mid \mathtt{c}\ C\ \mathtt{c} \\ C &\rightarrow \mathtt{c}\ S\ \mathtt{c} \\ D &\rightarrow \mathtt{d}\ X\ \mathtt{d} \\ X &\rightarrow C\ C \end{aligned}$$

  - D, $\mathtt{d}$, e X são inacessíveis
  - Os restantes são acessíveis

Símbolos acessíveis
Algoritmo de cálculo

- O conjunto dos seus símbolos acessíveis, V_A, pode ser obtido por aplicação das seguintes regras construtivas

  $S \in V_A$
  $\texttt{if } A \to \alpha B \beta \in P \texttt{ and } A \in V_A \texttt{ then } B \in V_A$

- Algoritmo de cálculo:

$V_A \leftarrow \{S\}$ # *no fim, ficará com todos os símbolos acessíveis*
$N_A \leftarrow \{S\}$ # *conjunto de símbolos não terminais acessíveis a processar*
repeat
  $X \leftarrow$ `elementOf(`N_A`)` # *retira um elemento qualquer de* N_A
  foreach $(X \to \alpha) \in P$ **do**
    foreach x **in** α **do**
      if $x \notin V_A$ **then** # *se ainda não está marcado como acessível*
        $V_A \leftarrow V_A \cup \{x\}$ # *passa a estar*
        if $x \in N$ **then** # *se adicinalmente é não terminal*
          $N_A \leftarrow N_A \cup \{x\}$ # *terá de ser processado*
until $N_A = \emptyset$

Gramáticas limpas
Algoritmo de limpeza

- Numa gramática, os símbolos inacessíveis e os símbolos improdutivos são **símbolos inúteis**
- Se tais símbolos forem removidos obtém-se uma gramática equivalente
- Diz-se que uma gramática é **limpa** se não possuir símbolos inúteis
- Para limpar uma gramática deve-se:
  - começar por a expurgar dos símbolos improdutivos
  - só depois remover os inacessíveis

Gramáticas limpas
Exemplo #1

$\mathcal{Q}$ Sobre o conjunto de terminais $T = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$, determine uma gramática limpa equivalente à gramática seguinte

$$
\begin{aligned}
S &\rightarrow \mathtt{a}\ A\ \mathtt{b} \mid \mathtt{b}\ B \\
A &\rightarrow \mathtt{c}\ C \mid \mathtt{b}\ B \mid \mathtt{d} \\
B &\rightarrow \mathtt{d}\ D \mid \mathtt{b} \\
C &\rightarrow A\ C \mid B\ D \mid S\ D \\
D &\rightarrow A\ D \mid B\ C \mid C\ S \\
E &\rightarrow \mathtt{a}\ A \mid \mathtt{b}\ B \mid \varepsilon
\end{aligned}
$$

- Cálculo dos símbolos produtivos

1 Inicialmente $N_p \leftarrow \emptyset$
2 $A \rightarrow \mathtt{d} \ \wedge \ \mathtt{d} \in T^* \quad \Longrightarrow \quad N_p \leftarrow N_p \cup \{A\}$
3 $B \rightarrow \mathtt{b} \ \wedge \ \mathtt{b} \in T^* \quad \Longrightarrow \quad N_p \leftarrow N_p \cup \{B\}$
4 $E \rightarrow \varepsilon \ \wedge \ \varepsilon \in T^* \quad \Longrightarrow \quad N_p \leftarrow N_p \cup \{E\}$
5 $S \rightarrow \mathtt{a}\, A\, \mathtt{b} \ \wedge \ \mathtt{a}, A, \mathtt{b} \in (T \cup N_p)^* \quad \Longrightarrow \quad N_p \leftarrow N_p \cup \{S\}$
6 Nada mais se consegue acrescentar a $N_p \quad \Longrightarrow \quad C$ e D são improdutivos

Gramáticas limpas
Exemplo #1, cont.

- Gramática após a remoção dos símbolos improdutivos

$$\begin{aligned} S &\to \mathtt{a}\ A\ \mathtt{b} \mid \mathtt{b}\ B \\ A &\to \mathtt{b}\ B \mid \mathtt{d} \\ B &\to \mathtt{b} \\ E &\to \mathtt{a}\ A \mid \mathtt{b}\ B \mid \varepsilon \end{aligned}$$

- Cálculo dos símbolos não terminais acessíveis sobre a nova gramática

1. S é acessível, porque é o inicial
2. sendo S acessível, de $S \to \mathtt{a}\ A\ \mathtt{b}$, tem-se que A é acessível
3. sendo S acessível, de $S \to \mathtt{b}\ B$, tem-se que B é acessível
4. de A só se chega a B, que já foi marcado como acessível
5. de B não se chega a nenhum não terminal
6. Logo E não é acessível, pelo que a gramática limpa é

$$\begin{aligned} S &\to \mathtt{a}\ A\ \mathtt{b} \mid \mathtt{b}\ B \\ A &\to \mathtt{b}\ B \mid \mathtt{d} \\ B &\to \mathtt{b} \end{aligned}$$