Compiladores

Linguagens Regulares, Expressões Regulares e Gramáticas Regulares

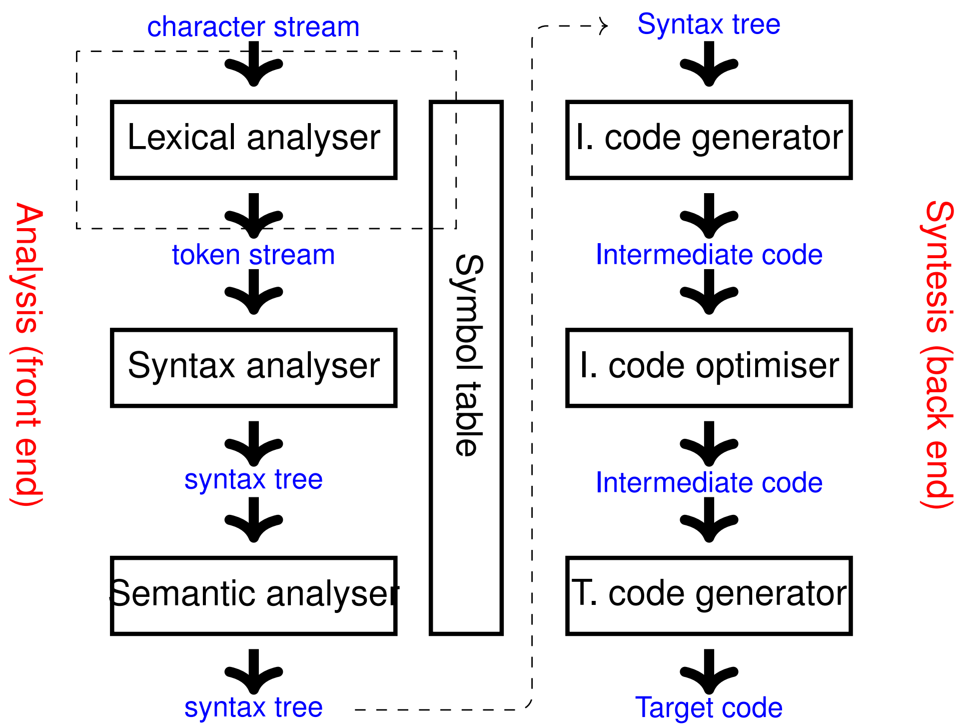
Artur Pereira <artur@ua.pt>,
Miguel Oliveira e Silva <mos@ua.pt

DETI, Universidade de Aveiro

Ano letivo de 2022-2023

Sumário

Papel da análise lexical



Papel da análise lexical

- Converte a sequência de caracteres numa sequência de *tokens*
- Um *token* é um tuplo `<token-name, attribute-value>`
  - `token-name` é um símbolo (abstrato) representando um tipo de entrada
  - `attribute-value` representa o valor corrente desse símbolo
- Exemplo:

  ```
  pos = pos + vel * 5;
  ```

  é convertido em

  ```
  <ID,"pos"> <=> <ID,"pos"> <+> <ID,"vel">
    <*> <INT,5>
  ```
- Tipicamente, alguns símbolos são descartados pelo analisador lexical
- O conjunto dos *tokens* corresponde a uma linguagem regular
  - os *tokens* são descritos usando expressões regulares e/ou gramáticas regulares
  - são reconhecidos usando autómatos finitos

Linguagem regular
Definição

A classe das **linguagens regulares** sobre o alfabeto A define-se indutivamente da seguinte forma:

1. O conjunto vazio, $\emptyset$, é uma linguagem regular (LR).
2. Qualquer que seja o $\mathtt{a} \in A$, o conjunto $\{\mathtt{a}\}$ é uma LR.

Note que:

- em $\mathtt{a} \in A$, $\mathtt{a}$ é uma letra do alfabeto
- em $\{\mathtt{a}\}$, $\mathtt{a}$ é uma palavra com apenas uma letra
- Numa analogia Java, o primeiro é um ’a’ e o segundo um ”a”

Linguagem regular
Definição

A classe das **linguagens regulares** sobre o alfabeto A define-se indutivamente da seguinte forma:

1. O conjunto vazio, $\emptyset$, é uma linguagem regular (LR).
2. Qualquer que seja o $\mathtt{a} \in A$, o conjunto $\{\mathtt{a}\}$ é uma LR.
3. Se L_1 e L_2 são LR, então a sua reunião $(L_1 \cup L_2)$ é uma LR.

Exemplo:

- Seja $L_1 = \{\mathtt{ab}, \mathtt{c}\}$, uma LR sobre o alfabeto $A = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- e $L_2 = \{\mathtt{bb}, \mathtt{c}\}$, outra LR sobre o mesmo alfabeto A
- então, $L_3 = L_1 \cup L_2 = \{\mathtt{ab}, \mathtt{bb}, \mathtt{c}\}$ é uma LR sobre o mesmo alfabeto A

Linguagem regular
Definição

A classe das **linguagens regulares** sobre o alfabeto A define-se indutivamente da seguinte forma:

1. O conjunto vazio, $\emptyset$, é uma linguagem regular (LR).
2. Qualquer que seja o $\mathtt{a} \in A$, o conjunto $\{\mathtt{a}\}$ é uma LR.
3. Se L_1 e L_2 são LR, então a sua reunião $(L_1 \cup L_2)$ é uma LR.
4. Se L_1 e L_2 são LR, então a sua concatenação $(L_1 \cdot L_2)$ é uma LR.

Exemplo:

- Seja $L_1 = \{\mathtt{ab}, \mathtt{c}\}$, uma LR sobre o alfabeto $A = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- e $L_2 = \{\mathtt{bb}, \mathtt{c}\}$, outra LR sobre o mesmo alfabeto A
- então, $L_3 = L_1 \cdot L_2 = \{\mathtt{abbb}, \mathtt{abc}, \mathtt{cbb}, \mathtt{cc}\}$ é uma LR sobre o mesmo alfabeto A

Linguagem regular
Definição

A classe das **linguagens regulares** sobre o alfabeto A define-se indutivamente da seguinte forma:

1. O conjunto vazio, $\emptyset$, é uma linguagem regular (LR).
2. Qualquer que seja o $\mathtt{a} \in A$, o conjunto $\{\mathtt{a}\}$ é uma LR.
3. Se L_1 e L_2 são LR, então a sua reunião $(L_1 \cup L_2)$ é uma LR.
4. Se L_1 e L_2 são LR, então a sua concatenação $(L_1 \cdot L_2)$ é uma LR.
5. Se L_1 é uma LR, então o seu fecho de Kleene $(L_1)^*$ é uma LR.

Exemplo:

- Seja $L_1 = \{\mathtt{ab}, \mathtt{c}\}$, uma LR sobre o alfabeto $A = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- então, $L_2 = {L_1}^* = \{\varepsilon, \mathtt{ab}, \mathtt{c}, \mathtt{abab}, \mathtt{abc}, \mathtt{cab}, \mathtt{cc}, \cdots\}$ é uma LR sobre o mesmo alfabeto

ACP (DETI/UA) Comp 2022/2023 Abril de 2023 7 / 38

Linguagem regular
Definição

A classe das **linguagens regulares** sobre o alfabeto A define-se indutivamente da seguinte forma:

1. O conjunto vazio, $\emptyset$, é uma linguagem regular (LR).
2. Qualquer que seja o $\mathtt{a} \in A$, o conjunto $\{\mathtt{a}\}$ é uma LR.
3. Se L_1 e L_2 são LR, então a sua reunião $(L_1 \cup L_2)$ é uma LR.
4. Se L_1 e L_2 são LR, então a sua concatenação $(L_1 \cdot L_2)$ é uma LR.
5. Se L_1 é uma LR, então o seu fecho de Kleene $(L_1)^*$ é uma LR.
6. Nada mais é LR.

Note que

- $\{\varepsilon\}$ é uma LR, uma vez que $\{\varepsilon\} = \emptyset^*$.

Definição de linguagem regular
exemplo #1

$\mathcal{Q}$ Mostre que a linguagem L, constituída pelo conjunto dos números binários começados em $\mathtt{1}$ e terminados em $\mathtt{0}$ é uma LR sobre o alfabeto $A = \{\mathtt{0}, \mathtt{1}\}$

$\mathcal{R}$

- pela regra 2 (elementos primitivos), $\{\mathtt{0}\}$ e $\{\mathtt{1}\}$ são LR
- pela regra 3 (união), $\{\mathtt{0}, \mathtt{1}\} = \{\mathtt{0}\} \cup \{\mathtt{1}\}$ é uma LR
- pela regra 5 (fecho), $\{\mathtt{0}, \mathtt{1}\}^*$ é uma LR
- pela regra 4 (concatenação), $\{\mathtt{1}\} \cdot \{\mathtt{0}, \mathtt{1}\}^*$ é uma LR
- pela regra 4, $(\{\mathtt{1}\} \cdot \{\mathtt{0}, \mathtt{1}\}^*) \cdot \{\mathtt{0}\}$ é uma LR
- logo, $L = \{\mathtt{1}\} \cdot \{\mathtt{0}, \mathtt{1}\}^* \cdot \{\mathtt{0}\}$ é uma LR

Expressões regulares
Definição

O conjunto das **expressões regulares** sobre o alfabeto A define-se indutivamente da seguinte forma:

1. $\emptyset$ é uma expressão regular (ER) que representa a LR $\{\}$.
2. Qualquer que seja o $a \in A$, a é uma ER que representa a LR $\{a\}$.
3. Se e_1 e e_2 são ER representando respetivamente as LR L_1 e L_2, então $(e_1|e_2)$ é uma ER representando a LR $L_1 \cup L_2$.
4. Se e_1 e e_2 são ER representando respetivamente as LR L_1 e L_2, então $(e_1 e_2)$ é uma ER representando a LR $L_1.L_2$.
5. Se e_1 é uma ER representando a LR L_1, então $(e_1)^*$ é uma ER representando a LR $(L_1)^*$.
6. Nada mais é expressão regular.

- É habitual representar-se por ε a ER $\emptyset^*$. Representa a linguagem $\{\varepsilon\}$.

Expressões regulares
Precedência dos operadores regulares

- Na escrita de expressões regulares assume-se a seguinte precedência dos operadores:
  - fecho (*)
  - concatenação
  - escolha (|).
- O uso destas precedências permite a queda de alguns parêntesis e consequentemente uma notação simplificada.

- Exemplo: a expressão regular

  $$e_1|e_2\, e_3*$$

  recorre a esta precedência para representar a expressão regular

  $$(e_1)|(e_2\,((e_3)*))$$

Expressões regulares
Exemplos

$\mathcal{Q}$ Determine uma ER que represente o conjunto dos números binários começados em `1` e terminados em `0`.

$\mathcal{R}$ `1(0|1)*0`

$\mathcal{Q}$ Determine uma ER que represente as sequências definidas sobre o alfabeto $A = \{\texttt{a}, \texttt{b}, \texttt{c}\}$ que satisfazem o requisito de qualquer `b` ter um `a` imediatamente à sua esquerda e um `c` imediatamente à sua direita.

$\mathcal{R}$ O `a` pode aparecer sozinho; o `c` também; o `b`, se aparecer, tem de ter um `a` à sua esquerda e um `c` à sua direita. Ou seja, pode considerar-se que as palavras da linguagem são sequências de 0 ou mais `a`, `c` ou `abc`.

`(a|abc|c)*`

$\mathcal{Q}$ Determine uma ER que represente as sequências binárias com um número par de zeros.

$\mathcal{R}$ $\texttt{(1*01*01*)*|1*} = \texttt{1*(01*01*)*}$

Expressões regulares
Propriedades da operação de escolha

- A operação de escolha goza das propriedades:
  - comutativa: $e_1 \mid e_2 = e_2 \mid e_1$
  - associativa: $e_1 \mid (e_2 \mid e_3) = (e_1 \mid e_2) \mid e_3 = e_1 \mid e_2 \mid e_3$
  - idempotência: $e_1 \mid e_1 = e_1$
  - existência de elemento neutro: $e_1 \mid \emptyset = \emptyset \mid e_1 = e_1$

- Exemplo:
  - comutativa: $\texttt{a} \mid \texttt{ab} = \texttt{ab} \mid \texttt{a}$
  - associativa: $\texttt{a} \mid (\texttt{b} \mid \texttt{ca}) = (\texttt{a} \mid \texttt{b}) \mid \texttt{ca} = \texttt{a} \mid \texttt{b} \mid \texttt{ca}$
  - idempotência: $\texttt{ab} \mid \texttt{ab} = \texttt{ab}$
  - não há interesse prático em fazer uma união com o conjunto vazio

Expressões regulares
Propriedades da operação de concatenação

- A operação de concatenação goza das propriedades:
  - associativa: $e_1(e_2e_3) = (e_1e_2)e_3 = e_1e_2e_3$
  - existência de elemento neutro: $e_1\varepsilon = \varepsilon e_1 = e_1$
  - existência de elemento absorvente: $e_1\,\emptyset = \emptyset\, e_1 = \emptyset$
  - **não goza da propriedade comutativa**

- Exemplo: seja $e_1 = \mathtt{a}$, $e_2 = \mathtt{bc}$, e $e_3 = \mathtt{c}$
  - associativa: $\mathtt{a}(\mathtt{bc}\,\mathtt{c}) = (\mathtt{a}\,\mathtt{bc})\mathtt{c} = \mathtt{a}\,\mathtt{bc}\,\mathtt{c}$

Expressões regulares
Propriedades distributivas

- A combinação das operações de concatenação e escolha gozam das propriedades:
  - distributiva à esquerda da concatenação em relação à escolha:

    $e_1(e_2 \mid e_3) = e_1e_2 \mid e_1e_3$
  - distributiva à direita da concatenação em relação à escolha:

    $(e_1 \mid e_2)e_3 = e_1e_3 \mid e_2e_3$

- Exemplo:
  - distributiva à esquerda da concatenação em relação à escolha:

    $\mathtt{ab}\,(\mathtt{a} \mid \mathtt{cc}) = \mathtt{aba} \mid \mathtt{abcc}$
  - distributiva à direita da concatenação em relação à escolha:

    $(\mathtt{ab} \mid \mathtt{a})\,\mathtt{cc} = \mathtt{abcc} \mid \mathtt{acc}$

Expressões regulares
Propriedades da operação de fecho de Kleene

- A operação de fecho goza das propriedades:
  - $(e^*)^* = e^*$
  - $({e_1}^* \,|\, {e_2}^*)^* = (e_1 \,|\, e_2)^*$
  - $(e_1 \,|\, {e_2}^*)^* = (e_1 \,|\, e_2)^*$
  - $({e_1}^* \,|\, e_2)^* = (e_1 \,|\, e_2)^*$
- Mas atenção:
  - $(e_1 \,|\, e_2)^* \neq {e_1}^* \,|\, {e_2}^*$
  - $(e_1\, e_2)^* \neq {e_1}^*\, {e_2}^*$

- Exemplo:
  - $\mathtt{b}\,(\mathtt{a}^*)^* = \mathtt{b}\,\mathtt{a}^*$
  - $(\mathtt{a}^* \,|\, \mathtt{b}^*)^* = (\mathtt{a} \,|\, \mathtt{b})^*$
  - $(\mathtt{a} \,|\, \mathtt{b}^*)^* = (\mathtt{a} \,|\, \mathtt{b})^*$
  - $(\mathtt{a}^* \,|\, \mathtt{b})^* = (\mathtt{a} \,|\, \mathtt{b})^*$
  - $(\mathtt{a} \,|\, \mathtt{b})^* \neq \mathtt{a}^* \,|\, \mathtt{b}^*$
  - $(\mathtt{a}\,\mathtt{b})^* \neq \mathtt{a}^*\,\mathtt{b}^*$

Expressões regulares
Exemplos

$\mathcal{Q}$ Sobre o alfabeto $A = \{\mathtt{0}, \mathtt{1}\}$ construa uma expressão regular que represente a linguagem

$$L = \{\omega \in A^* \;:\; \#(\mathtt{0}, \omega) = 2\}$$

$\mathcal{R}$ $\mathtt{1^*01^*01^*}$

$\mathcal{Q}$ Sobre o alfabeto $A = \{\mathtt{a}, \mathtt{b}, \cdots, \mathtt{z}\}$ construa uma expressão regular que represente a linguagem

$$L = \{\omega \in A^* \;:\; \#(\mathtt{a}, \omega) = 3\}$$

$\mathcal{R}$ $(\mathtt{b}|\mathtt{c}|\cdots|\mathtt{z})^*\mathtt{a}(\mathtt{b}|\mathtt{c}|\cdots|\mathtt{z})^*\mathtt{a}(\mathtt{b}|\mathtt{c}|\cdots|\mathtt{z})^*\mathtt{a}(\mathtt{b}|\mathtt{c}|\cdots|\mathtt{z})^*$

- Na última resposta, onde estão as reticências (...) deveriam estar todas as letras entre $\mathtt{d}$ e $\mathtt{y}$. Parece claro que faz falta uma forma de simplificar este tipo de expressões

Expressões regulares
Extensões notacionais comuns

- uma ou mais ocorrências:

  $e^+ = e.e^*$
- uma ou nenhuma ocorrência:

  $e? = (e|\varepsilon)$
- um símbolo do sub-alfabeto dado:

  $[a_1 a_2 a_3 \cdots a_n] = (a_1 \mid a_2 \mid a_3 \mid \cdots \mid a_n)$
- um símbolo do sub-alfabeto dado:

  $[a_1 - a_n] = (a_1 \mid \cdots \mid a_n)$
- um símbolo do alfabeto fora do conjunto dado:

  $[\hat{}a_1 a_2 a_3 \cdots a_n]$, $[\hat{}a_1 - a_n]$

Em ANTLR:

- `x..y` é equivalente a `[x-y]`
- `~[abc]` é equivalente a `[^abc]`

Expressões regulares
Outras extensões notacionais

- n ocorrências de:

  $e\{n\} = \underbrace{e.e.\cdots.e}_{n}$
- de n_1 a n_2 ocorrências:

  $e\{n_1, n_2\} = \underbrace{e.e.\cdots.e}_{n_1, n_2}$
- n ou mais ocorrências:

  $e\{n\,,\} = \underbrace{e.e.\cdots.e}_{n,}$

- . representa um símbolo qualquer
- ^ representa palavra vazia no início de linha
- $ representa palavra vazia no fim de linha
- $\backslash<$ representa palavra vazia no início de palavra
- $\backslash>$ representa palavra vazia no fim de palavra

Em ANTLR:

- Pode ser feito através de predicados semânticos

Expressões regulares
Exemplos de extensões notacionais

$\mathcal{Q}$ Sobre o alfabeto $A = \{\texttt{0}, \texttt{1}\}$ construa uma expressão regular que reconheça a linguagem

$$L = \{\omega \in A^* \;:\; \#(\texttt{0}, \omega) = 2\}$$

$\mathcal{R}$ $\texttt{1}^*\texttt{0}\texttt{1}^*\texttt{0}\texttt{1}^* = (\texttt{1}^*\texttt{0})\,(\texttt{1}^*\texttt{0})\,\texttt{1}^* = (\texttt{1}^*\texttt{0})\{2\}\,\texttt{1}^*$

$\mathcal{Q}$ Sobre o alfabeto $A = \{\texttt{a}, \texttt{b}, \cdots, \texttt{z}\}$ construa uma expressão regular que reconheça a linguagem

$$L = \{\omega \in A^* \;:\; \#(\texttt{a}, \omega) = 3\}$$

$\mathcal{R}$ $(\texttt{b}|\texttt{c}|\cdots|\texttt{z})^*\texttt{a}(\texttt{b}|\texttt{c}|\cdots|\texttt{z})^*\texttt{a}(\texttt{b}|\texttt{c}|\cdots|\texttt{z})^*\texttt{a}(\texttt{b}|\texttt{c}|\cdots|\texttt{z})^*$

$$= ([\texttt{b-z}]^*\texttt{a})\,([\texttt{b-z}]^*\texttt{a})\,([\texttt{b-z}]^*\texttt{a})\,[\texttt{b-z}]^*$$

$$= ([\texttt{b-z}]^*\texttt{a})\{3\}[\texttt{b-z}]^*$$

Gramáticas regulares
Introdução

- Exemplo de gramática regular

$$\begin{aligned} S &\to \texttt{a}\ X \\ X &\to \texttt{a}\ X \\ &\mid\ \texttt{b}\ X \\ &\mid\ \varepsilon \end{aligned}$$

- Exemplo de gramática não regular

$$\begin{aligned} S &\to \texttt{a}\ S\ \texttt{a} \\ &\mid\ \texttt{b}\ S\ \texttt{b} \\ &\mid\ \texttt{a} \end{aligned}$$

- Letras minúsculas representam símbolos terminais e letras maísculas representam símbolos não terminais (o contrário do ANTLR)
- Nas gramáticas regulares os símbolos não terminais apenas podem aparecer no fim

Gramáticas regulares
Definição

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

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

- A linguagem gerada por uma gramática regular é regular
  - Logo, é possível converter-se uma gramática regular numa expressão regular que represente a mesma linguagem e vice-versa

Gramáticas regulares
Operações sobre gramáticas regulares

- As gramáticas regulares são fechadas sob as operações de
  - reunião
  - concatenação
  - fecho
  - intersecção
  - complementação
- As operações de intersecção e complementação serão abordadas mais adiante através de autómatos finitos

Reunião de gramáticas regulares
Exemplo

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

$$L = L_1 \cup L_2$$

sabendo que

$$L_1 = \{\omega\mathtt{a} \,:\, \omega \in T^*\} \qquad L_2 = \{\mathtt{a}\omega \,:\, \omega \in T^*\}$$

$\mathcal{R}$

$$\begin{array}{rcl} S_1 & \rightarrow & \mathtt{a}\ S_1 \\ & \mid & \mathtt{b}\ S_1 \\ & \mid & \mathtt{c}\ S_1 \\ & \mid & \mathtt{a} \end{array} \qquad \begin{array}{rcl} S_2 & \rightarrow & \mathtt{a}\ X_2 \\ X_2 & \rightarrow & \mathtt{a}\ X_2 \\ & \mid & \mathtt{b}\ X_2 \\ & \mid & \mathtt{c}\ X_2 \\ & \mid & \varepsilon \end{array}$$

- Comece-se por obter as gramáticas regulares que representam L_1 e L_2.

Reunião de gramáticas regulares
Exemplo

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

$$L = L_1 \cup L_2$$

sabendo que

$$L_1 = \{\omega\mathtt{a} \,:\, \omega \in T^*\} \qquad L_2 = \{\mathtt{a}\omega \,:\, \omega \in T^*\}$$

$\mathcal{R}$

$$\begin{array}{rcl} S_1 & \rightarrow & \mathtt{a}\ S_1 \\ & \mid & \mathtt{b}\ S_1 \\ & \mid & \mathtt{c}\ S_1 \\ & \mid & \mathtt{a} \end{array} \qquad \begin{array}{rcl} S_2 & \rightarrow & \mathtt{a}\ X_2 \\ X_2 & \rightarrow & \mathtt{a}\ X_2 \\ & \mid & \mathtt{b}\ X_2 \\ & \mid & \mathtt{c}\ X_2 \\ & \mid & \varepsilon \end{array} \qquad \begin{array}{rcl} S & \rightarrow & S_1 \mid S_2 \\ S_1 & \rightarrow & \mathtt{a}\ S_1 \mid \mathtt{b}\ S_1 \mid \mathtt{c}\ S_1 \\ & \mid & \mathtt{a} \\ S_2 & \rightarrow & \mathtt{a}\ X_2 \\ X_2 & \rightarrow & \mathtt{a}\ X_2 \mid \mathtt{b}\ X_2 \mid \mathtt{c}\ X_2 \\ & \mid & \varepsilon \end{array}$$

- E acrescentam-se as transições $S \rightarrow S_1$ e $S \rightarrow S_2$ que permitem escolher as palavras de L_1 e de L_2, sendo S o novo símbolo inicial.

Reunião de gramáticas regulares
Algoritmo

$\mathcal{D}$ Sejam $G_1 = (T_1, N_1, P_1, S_1)$ e $G_2 = (T_2, N_2, P_2, S_2)$ duas gramáticas regulares quaisquer, com $N_1 \cap N_2 = \emptyset$. A gramática $G = (T, N, P, S)$ onde

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

é regular e gera a linguagem $L = L(G_1) \cup L(G_2)$.

- Para $i = 1, 2$, a nova produção $S \to S_i$ permite que G gere a linguagem $L(G_i)$

Concatenação de gramáticas regulares
Exemplo

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

$$L = L_1 \cdot L_2$$

sabendo que

$$L_1 = \{\omega\mathtt{a} \,:\, \omega \in T^*\} \qquad L_2 = \{\mathtt{a}\omega \,:\, \omega \in T^*\}$$

$\mathcal{R}$

$$\begin{array}{ll} S_1 \to \mathtt{a}\ S_1 & \\ \quad\ \mid\ \mathtt{b}\ S_1 & \\ \quad\ \mid\ \mathtt{c}\ S_1 & \\ \quad\ \mid\ \mathtt{a} & \end{array} \qquad \begin{array}{l} S_2 \to \mathtt{a}\ X_2 \\ X_2 \to \mathtt{a}\ X_2 \\ \quad\ \mid\ \mathtt{b}\ X_2 \\ \quad\ \mid\ \mathtt{c}\ X_2 \\ \quad\ \mid\ \varepsilon \end{array}$$

- Comece-se por obter as gramáticas regulares que representam L_1 e L_2.

Concatenação de gramáticas regulares
Exemplo

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

$$L = L_1 \cdot L_2$$

sabendo que

$$L_1 = \{\omega\mathtt{a} \, : \, \omega \in T^*\} \qquad L_2 = \{\mathtt{a}\omega \, : \, \omega \in T^*\}$$

$\mathcal{R}$

$$\begin{array}{lll}
S_1 \to \mathtt{a}\ S_1 & S_2 \to \mathtt{a}\ X_2 & S_1 \to \mathtt{a}\ S_1 \mid \mathtt{b}\ S_1 \mid \mathtt{c}\ S_1 \\
\quad \mid \mathtt{b}\ S_1 & X_2 \to \mathtt{a}\ X_2 & \quad \mid \mathtt{a}\ S_2 \\
\quad \mid \mathtt{c}\ S_1 & \quad \mid \mathtt{b}\ X_2 & S_2 \to \mathtt{a}\ X_2 \\
\quad \mid \mathtt{a} & \quad \mid \mathtt{c}\ X_2 & X_2 \to \mathtt{a}\ X_2 \mid \mathtt{b}\ X_2 \mid \mathtt{c}\ X_2 \\
 & \quad \mid \varepsilon &
\end{array}$$

- A seguir substitui-se $S_1 \to \mathtt{a}$ por $S_1 \to \mathtt{a}\ S_2$, de modo a impor que a segunda parte das palavras têm de pertencer a L_2

Concatenação de gramáticas regulares
Algoritmo

$\mathcal{D}$ Sejam $G_1 = (T_1, N_1, P_1, S_1)$ e $G_2 = (T_2, N_2, P_2, S_2)$ duas gramáticas regulares quaisquer, com $N_1 \cap N_2 = \emptyset$. A gramática $G = (T, N, P, S)$ onde

$$\begin{array}{rcl}
T & = & T_1 \cup T_2 \\
N & = & N_1 \cup N_2 \\
P & = & \{A \to \omega S_2 \, : \, (A \to \omega) \in P_1 \ \wedge \ \omega \in {T_1}^*\} \\
 & & \cup \{A \to \omega \, : \, (A \to \omega) \in P_1 \ \wedge \ \omega \in {T_1}^* N_1\} \\
 & & \cup P_2 \\
S & = & S_1
\end{array}$$

é regular e gera a linguagem $L = L(G_1) \cdot L(G_2)$.

- As produções da primeira gramática do tipo $\beta \in T^*$ ganham o símbolo inicial da segunda gramática no fim
- As produções da primeira gramática do tipo $\beta \in T^* N$ mantêm-se inalteradas
- As produções da segunda gramática mantêm-se inalteradas

Fecho de gramáticas regulares
Exemplo

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

$$L = {L_1}^*$$

sabendo que

$$L_1 = \{\omega\mathtt{a} \;:\; \omega \in T^*\}$$

$\mathcal{R}$

$$\begin{aligned} S_1 &\to \mathtt{a}\ S_1 \\ &\mid \mathtt{b}\ S_1 \\ &\mid \mathtt{c}\ S_1 \\ &\mid \mathtt{a} \end{aligned}$$

- Começa-se pela obtenção da gramática regular que representa L_1.

Fecho de gramáticas regulares
Exemplo

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

$$L = {L_1}^*$$

sabendo que

$$L_1 = \{\omega\mathtt{a} \;:\; \omega \in T^*\}$$

$\mathcal{R}$

$$\begin{aligned} S_1 &\to \mathtt{a}\ S_1 \\ &\mid \mathtt{b}\ S_1 \\ &\mid \mathtt{c}\ S_1 \\ &\mid \mathtt{a} \end{aligned}$$

$$\begin{aligned} S &\to \varepsilon \mid S_1 \\ S_1 &\to \mathtt{a}\ S_1 \mid \mathtt{b}\ S_1 \mid \mathtt{c}\ S_1 \\ &\mid \mathtt{a}\ S \end{aligned}$$

- Acrescentando-se a transição $S \to S_1$ e substituindo-se $S_1 \to \mathtt{a}$ por $S_1 \to \mathtt{a}\ S$, permite-se iterações sobre S_1
- Acrescentando-se $S \to \varepsilon$, permite-se 0 ou mais iterações

Fecho de gramáticas regulares
Algoritmo

$\mathcal{D}$ Seja $G_1 = (T_1, N_1, P_1, S_1)$ uma gramática regular qualquer. A gramática $G = (T, N, P, S)$ onde

$$\begin{aligned} T &= T_1 \\ N &= N_1 \cup \{S\} \quad \text{com} \quad S \notin N_1 \\ P &= \{S \to \varepsilon, S \to S_1\} \\ &\quad \cup \{A \to \omega S \;:\; (A \to \omega) \in P_1 \;\wedge\; \omega \in {T_1}^*\} \\ &\quad \cup \{A \to \omega \;:\; (A \to \omega) \in P_1 \;\wedge\; \omega \in {T_1}^* N_1\} \end{aligned}$$

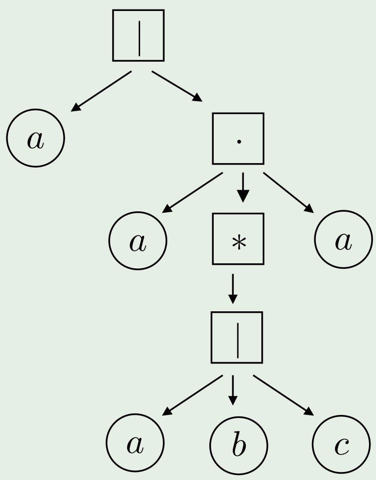
é regular e gera a linguagem $L = (L(G_1))^*$.

- As novas produções $S \to \varepsilon$ e $S \to S_1$ garantem que $(L(G_1))^n \subseteq L(G)$, para qualquer $n \geq 0$
- As produções que só têm terminais ganham o novo símbolo inicial no fim
- As produções que terminam num não terminal mantêm-se inalteradas

Conversão de uma ER em uma GR
exemplo

$\mathcal{Q}$ Construa uma GR equivalente à ER $e = \mathtt{a}|\mathtt{a}(\mathtt{a}|\mathtt{b}|\mathtt{c})^*\mathtt{a}$.

$\mathcal{R}$

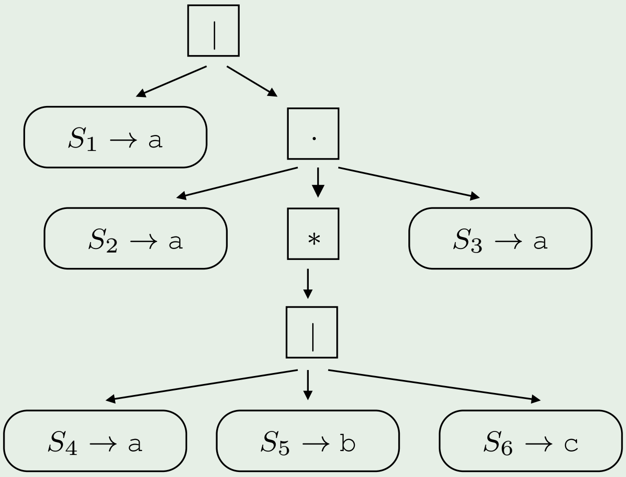


- Coloque-se de forma arbórea

Conversão de uma ER em uma GR
exemplo

$\mathcal{Q}$ Construa uma GR equivalente à ER $e = \mathtt{a}|\mathtt{a}(\mathtt{a}|\mathtt{b}|\mathtt{c})^*\mathtt{a}$.

$\mathcal{R}$

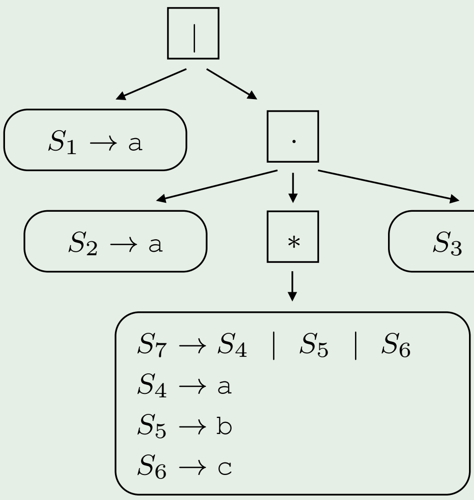


- Após converter as folhas (elementos primitivos) em GR

Conversão de uma ER em uma GR
exemplo

$\mathcal{Q}$ Construa uma GR equivalente à ER $e = \mathtt{a}|\mathtt{a}(\mathtt{a}|\mathtt{b}|\mathtt{c})^*\mathtt{a}$.

$\mathcal{R}$

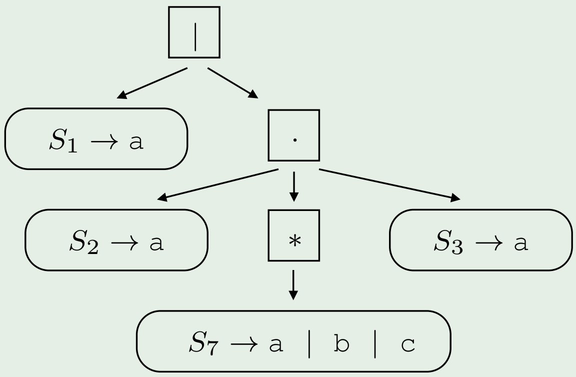


- Após aplicar a escolha (reunião) de baixo

Conversão de uma ER em uma GR
exemplo

$\mathcal{Q}$ Construa uma GR equivalente à ER $e = \mathtt{a}|\mathtt{a}(\mathtt{a}|\mathtt{b}|\mathtt{c})^*\mathtt{a}$.

$\mathcal{R}$

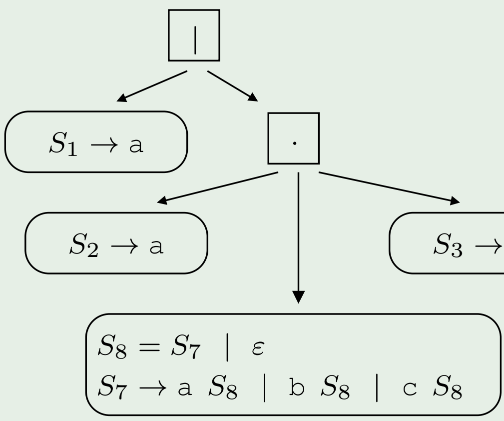


- Simplificando

Conversão de uma ER em uma GR
exemplo

$\mathcal{Q}$ Construa uma GR equivalente à ER $e = \mathtt{a}|\mathtt{a}(\mathtt{a}|\mathtt{b}|\mathtt{c})^*\mathtt{a}$.

$\mathcal{R}$

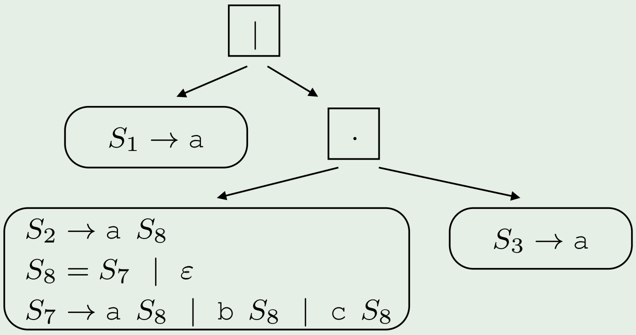


- Após aplicar o fecho

Conversão de uma ER em uma GR
exemplo

$\mathcal{Q}$ Construa uma GR equivalente à ER $e = \texttt{a}|\texttt{a}(\texttt{a}|\texttt{b}|\texttt{c})^*\texttt{a}$.

$\mathcal{R}$

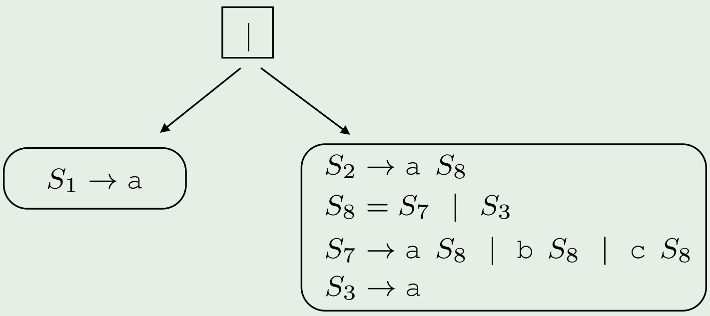


- Após aplicar a concatenação da esquerda

Conversão de uma ER em uma GR
exemplo

$\mathcal{Q}$ Construa uma GR equivalente à ER $e = \texttt{a}|\texttt{a}(\texttt{a}|\texttt{b}|\texttt{c})^*\texttt{a}$.

$\mathcal{R}$



- Após aplicar a concatenação da direita

Conversão de uma ER em uma GR
exemplo

$\mathcal{Q}$ Construa uma GR equivalente à ER $e = \mathtt{a}|\mathtt{a}(\mathtt{a}|\mathtt{b}|\mathtt{c})^*\mathtt{a}$.

$\mathcal{R}$

$$
\begin{aligned}
&S \to S_1 \ | \ S_2 \\
&S_1 \to \mathtt{a} \\
&S_2 \to \mathtt{a} \ S_8 \\
&S_8 \to S_7 \ | \ S_3 \\
&S_7 \to \mathtt{a} \ S_8 \ | \ \mathtt{b} \ S_8 \ | \ \mathtt{c} \ S_8 \\
&S_3 \to \mathtt{a}
\end{aligned}
$$

e simplificando

$$
\begin{aligned}
&S \to \mathtt{a} \ | \ \mathtt{a} \ S_8 \\
&S_8 \to \mathtt{a} \ S_8 \ | \ \mathtt{b} \ S_8 \ | \ \mathtt{c} \ S_8 \ | \ \mathtt{a}
\end{aligned}
$$

- Finalmente após aplicar escolha (reunião) de cima

Conversão de uma ER em uma GR
Abordagem

- Dada uma expressão regular qualquer ela é:
  - ou um elemento primitivo;
  - ou uma expressão do tipo e^*, sendo e uma expressão regular qualquer;
  - ou uma expressão do tipo $e_1.e_2$, sendo e_1 e e_2 duas expressões regulares quaisquer;
  - ou uma expressão do tipo $e_1|e_2$, sendo e_1 e e_2 duas expressões regulares quaisquer;
- Identificando-se as GR equivalentes às ER primitivas, tem-se o problema resolvido, visto que se sabe como fazer a reunião, a concatenação e o fecho de GR.

expressão regular	gramática regular
ε	$S \to \varepsilon$
$\mathtt{a}$	$S \to \mathtt{a}$

Conversão de uma ER em uma GR
Algoritmo de conversão

1. Se a ER é do tipo primitivo, a GR correspondente pode ser obtido da tabela anterior.
2. Se é do tipo e^*, aplica-se este mesmo algoritmo na obtenção de uma GR equivalente à expressão regular e e, de seguida, aplica-se o fecho de GR.
3. Se é do tipo $e_1.e_2$, aplica-se este mesmo algoritmo na obtenção de GR para as expressões e_1 e e_2 e, de seguida, aplica-se a concatenação de GR.
4. Finalmente, se é do tipo $e_1|e_2$, aplica-se este mesmo algoritmo na obtenção de GR para as expressões e_1 e e_2 e, de seguida, aplica-se a reunião de GR.

- Na realidade, o algoritmo corresponde a um processo de decomposição arbórea a partir da raiz seguido de um processo de construção arbórea a partir das folhas.

Conversão de uma GR em uma ER
Exemplo

$\mathcal{Q}$ Obtenha uma ER equivalente à gramática regular seguinte

$$S \to \mathtt{a}\ S \ |\ \mathtt{c}\ S\ |\ \mathtt{aba}\ X$$
$$X \to \mathtt{a}\ X\ |\ \mathtt{c}\ X\ |\ \varepsilon$$

$\mathcal{R}$ Abordagem admitindo expressões regulares nas produções das gramáticas

$$E \to \varepsilon\ S$$
$$S \to \mathtt{a}\ S\ |\ \mathtt{c}\ S\ |\ (\mathtt{aba})\ X$$
$$X \to \mathtt{a}\ X\ |\ \mathtt{c}\ X\ |\ \varepsilon\ \varepsilon$$

- *acrescentou-se um novo símbolo inicial de forma a garantir que não aparece do lado direito*

$$E \to \varepsilon\ S$$
$$S \to (\mathtt{a}|\mathtt{c})\ S\ |\ (\mathtt{aba})\ X$$
$$X \to (\mathtt{a}|\mathtt{c})\ X\ |\ \varepsilon\ \varepsilon$$

- *transformou-se* $S \to \mathtt{a}\ S$ *e* $S \to \mathtt{c}\ S$ *em* $S \to (\mathtt{a}|\mathtt{c})\ S$
- *fez-se algo similar com o X*

$$E \to \varepsilon\ (\mathtt{a}|\mathtt{c})^*\ (\mathtt{aba})\ X$$
$$X \to (\mathtt{a}|\mathtt{c})\ X\ |\ \varepsilon\ \varepsilon$$

- *transformaram-se as produções* $E \to \varepsilon\ S$, $S \to (\mathtt{a}|\mathtt{c})\ S$ *e* $S \to \mathtt{aba}\ X$ *em* $E \to (\mathtt{a}|\mathtt{c})^*\mathtt{aba}\ X$
- *Note que o* $(\mathtt{a}|\mathtt{c})$ *passou a* $(\mathtt{a}|\mathtt{c})^*$

$$E \to \varepsilon\ (\mathtt{a}|\mathtt{c})^*\ (\mathtt{aba})\ (\mathtt{a}|\mathtt{c})^*\ \varepsilon$$

- *repetiu-se com o* X, *obtendo-se a ER desejada:* $(\mathtt{a}|\mathtt{c})^*\mathtt{aba}(\mathtt{a}|\mathtt{c})^*$

Conversão de uma GR em uma ER
Exemplo

$\mathcal{Q}$ Obtenha uma ER equivalente à gramática regular seguinte

$$S \rightarrow \texttt{a}\ S \mid \texttt{c}\ S \mid \texttt{aba}\ X$$
$$X \rightarrow \texttt{a}\ X \mid \texttt{c}\ X \mid \varepsilon$$

$\mathcal{R}$ Abordagem transformando a gramática num conjunto e triplos

$\{(E, \varepsilon, S),$
$(S, \texttt{a}, S), (S, \texttt{c}, S), (S, \texttt{aba}, X),$
$(X, \texttt{a}, X), (X, \texttt{c}, X), (X, \varepsilon, \varepsilon)\}$

- *converte-se a gramática num conjunto de triplos, acrescentando um inicial*

$\{(E, \varepsilon, S), (S, (\texttt{a}|\texttt{c}), S), (S, \texttt{aba}, X),$
$(X, (\texttt{a}|\texttt{c}), X), (X, \varepsilon, \varepsilon)\}$

- *transformou-se* $(S, \texttt{a}, S), (S, \texttt{c}, S)$ *em* $(S, (\texttt{a}|\texttt{c}), S)$
- *fez-se algo similar com o X*

$\{(E, (\texttt{a}|\texttt{c})^*\texttt{aba}, X),$
$(X, (\texttt{a}|\texttt{c}), X), (X, \varepsilon, \varepsilon)\}$

- *transformou-se o triplo de triplos* $(E, \varepsilon, S), (S, (\texttt{a}|\texttt{c}), S), (S, \texttt{aba}, X)$ *em* $(E, (\texttt{a}|\texttt{c})^*\texttt{aba}, X)$
- *Note que o* $(\texttt{a}|\texttt{c})$ *passou a* $(\texttt{a}|\texttt{c})^*$

$\{(E, (\texttt{a}|\texttt{c})^*\texttt{aba}(\texttt{a}|\texttt{c})^*, \varepsilon)\}$

- *repetiu-se com o* X*, obtendo-se a ER desejada:* $(\texttt{a}|\texttt{c})^*\texttt{aba}(\texttt{a}|\texttt{c})^*$

Conversão de uma GR em uma ER
Algoritmo

- Uma expressão regular e que represente a mesma linguagem que a gramática regular G pode ser obtida por um processo de transformações de equivalência.
- Primeiro, converte-se a gramática $G = (T, N, P, S)$ no conjunto de triplos seguinte:

$$\begin{array}{rcl} \mathcal{E} & = & \{(E, \varepsilon, S)\} \\ & \cup & \{(A, \omega, B) \ : \ (A \rightarrow \omega B) \in P \ \wedge \ B \in N\} \\ & \cup & \{(A, \omega, \varepsilon) \ : \ (A \rightarrow \omega) \in P \ \wedge \omega \in T^*\} \end{array}$$

  com $E \notin N$.
- A seguir, removem-se, por transformações de equivalência, um a um, todos os símbolos de N, até se obter um único triplo da forma (E, e, ε).
- O valor de e é a expressão regular pretendida.

Conversão de uma GR em uma ER
Algoritmo de remoção dos símbolos de N

1. Substituir todos os triplos da forma (A, α_i, A), com $A \in N$, por um único (A, ω_2, A), onde $\omega_2 = \alpha_1 \mid \alpha_2 \mid \cdots \mid \alpha_m$
2. Substituir todos os triplos da forma (A, β_i, B), com $A, B \in N$, por um único (A, ω_1, B), onde $\omega_1 = \beta_1 \mid \beta_2 \mid \cdots \mid \beta_n$
3. Substituir cada triplo de triplos da forma $(A, \omega_1, B), (B, \omega_2, B), (B, \omega_3, C)$, com $A, B, C \in N$, pelo triplo $(A, \omega_1 {\omega_2}^* \omega_3, C)$
4. Repetir os passos anteriores enquanto houver símbolos intermédios

---

- Note que, se não existir qualquer triplo do tipo (A, α_i, A), ω_2 representa o conjunto vazio e consequentemente ${\omega_2}^* = \varepsilon$