Teoria da Computação

Aula Teórica 11:
Expressões Regulares

António Ravara

Departamento de Informática

10 de Abril de 2019

Palavra ou traço

Alfabeto

Um conjunto finito de símbolos diz-se um *alfabeto*. Exemplos:

- $\Sigma_{AFILE} = \{\texttt{open}, \texttt{read}, \texttt{write}, \texttt{close}\}$
- $DIGITS = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Palavra

- Uma sequência finita (eventualmente vazia) de símbolos de um alfabeto diz-se uma *palavra* (sobre esse alfabeto).
- ε é uma palavra sobre qualquer alfabeto;
- `close close write write` é uma palavra sobre Σ_{AFILE};
- `1 1 1 1` é uma palavra sobre $DIGITS$;
- `open 1 2 close` não é uma palavra, nem sobre Σ_{AFILE} nem sobre $DIGITS$.

Conjunto de palavras

Extensão de uma palavra com um símbolo

Dada uma palavra w sobre Σ e um símbolo $a \in \Sigma$, a sequência aw é a palavra sobre Σ obtida colocando a à frente do primeiro símbolo de w.
Exemplos:

- Se $w = \varepsilon$ e $a = 1$ então $aw = 1$;
- Se $w = 2\ 3$ e $a = 1$ então $aw = 1\ 2\ 3$.

$Words(\Sigma)$

O conjunto $Words(\Sigma)$ de todas as palavras sobre um alfabeto Σ define-se indutivamente como se segue:

$$\varepsilon \in Words(\Sigma)$$
$$w \in Words(\Sigma) \wedge a \in \Sigma \Rightarrow aw \in Words(\Sigma)$$

Linguagem

Propriedades

- Se Σ é não vazio, $Words(\Sigma)$ é infinitamente contável.
- Note-se que $Words(\Sigma) = \Sigma^*$.

$Lang(\Sigma)$

- Uma linguagem sobre Σ é um conjunto $\mathcal{L}_\Sigma \subseteq Words(\Sigma)$.
- Denote-se por $Lang(\Sigma) = \wp(Words(\Sigma))$ o conjunto de todas as linguagens sobre Σ.

Exemplos de linguagens

Nem todas as palavras pertencem a uma linguagem.

Seja $PRIMES \subseteq Words(DIGITS)$ o conjunto das palavras sobre $DIGITS$ que representam números primos

- $11 \in PRIMES$ mas $22 \notin PRIMES$,
- apesar de $11 \in Words(DIGITS)$ e $22 \in Words(DIGITS)$.

Considere-se $\mathcal{L}(AFILE)$, a linguagem das palavras aceites pelo AFD $AFILE$

- `open read close` $\in \mathcal{L}(AFILE)$ mas `open open` $\notin \mathcal{L}(AFILE)$,
- apesar de `open read close` $\in Words(\Sigma_{AFILE})$ e `open open` $\in Words(\Sigma_{AFILE})$.

Linguagem dos AFDs

- Vimos vários exemplos de linguagens ($PRIMES$, $Words(DIGITS)$, $\mathcal{L}(AFILE)$, $Words(\Sigma_{AFILE})$).
- Nem todas são linguagens aceites por algum AFD ($PRIMES$ não é).
- Como um AFD só tem um número finito de estados, não consegue aceitar todas as palavras de uma linguagem que precise de "memória" para saber o que já tratou, dada uma palavra arbitrária.
- Exemplos de linguagens que ultrapassam o poder expressivo dos AFDs:
  - $PRIMES$
  - $CAPICUAS$ (palavras que lidas da esquerda para a direita ou vice-versa denotam o mesmo natural).

Como caracterizar a linguagem dos AFDs?

Será possível definir precisamente que tipo de linguagens podem ser aceites por algum AFD?

Teorema de Kleene

Linguagens Regulares = Linguagens reconhecidas por AFDs

- Mas o que são "Linguagens Regulares"?
- São as denotadas por "Expressões Regulares".
- Uma expressão regular é uma forma concisa de identificar uma linguagem (regular).
  Define-se por indução o conjunto das expressões regulares (sobre dado alfabeto) e a linguagem denotada por dada expressão.

Expressões regulares

Definição

Dado um alfabeto Σ, o conjunto $RegExp(\Sigma)$ é definido indutivamente:

$$\emptyset \in RegExp(\Sigma)$$
$$\varepsilon \in RegExp(\Sigma)$$
$$a \in \Sigma \Rightarrow a \in RegExp(\Sigma)$$
$$E \in RegExp(\Sigma) \wedge F \in RegExp(\Sigma) \Rightarrow (EF) \in RegExp(\Sigma)$$
$$E \in RegExp(\Sigma) \wedge F \in RegExp(\Sigma) \Rightarrow (E + F) \in RegExp(\Sigma)$$
$$E \in RegExp(\Sigma) \Rightarrow (E^*) \in RegExp(\Sigma)$$

- Os três primeiros casos são a base: o vazio, a palavra vazia e qualquer símbolo do alfabeto (palavras de tamanho 1).
- Os "passos" são as operações *concatenação*, *soma* (ou escolha) e a *estrela* (fecho de Kleene).

Expressões regulares

Exemplos

São expressões regulares:

- $(a(b+c))$
- $((a^*)b)$
- $((ab)((b+c)^*))$

Para simplificar, omitem-se os parênteses exteriores, considera-se o operador estrela mais forte que a concatenação, e esta que a soma. As expressões acima escrevem-se:

- $a(b+c)$
- a^*b
- $ab(b+c)^*$

Linguagem denotada por uma expressão regular

Uma expressão regular identifica (univocamente) uma linguagem.
A partir de um AFD não é tão claro ver a linguagem...

Considere o alfabeto $\{a, b\}$

- O conjunto das palavras só com um b é denotado por a^*ba^*
- O conjunto das palavras começadas por a é denotado por $a(a+b)^*$
- O conjunto das palavras com pelo menos dois bs é denotado por $a^*ba^*b(a+b)^*$
- O conjunto das palavras com só dois bs e não consecutivos é denotado por $a^*ba^*aba^*$
- O conjunto das palavras com um número ímpar de bs é denotado por $a^*ba^*(ba^*ba^*)^*$

Linguagem denotada por uma expressão regular

Relembre as operações de concatenação de conjuntos ($A \cdot B$) e fecho de Kleene de um conjunto (A^*), da AT9.

Definição

Dada uma expressão regular $E \in RegExp(\Sigma)$, o conjunto $\mathcal{L}(E)$ é definido indutivamente:

$$\mathcal{L}(\emptyset) = \emptyset$$
$$\mathcal{L}(\varepsilon) = \{\varepsilon\}$$
$$a \in \Sigma \Rightarrow \mathcal{L}(a) = \{a\}$$
$$\mathcal{L}(E) = L_E \wedge \mathcal{L}(F) = L_F \Rightarrow \mathcal{L}(EF) = L_E \cdot L_F$$
$$\mathcal{L}(E) = L_E \wedge \mathcal{L}(F) = L_F \Rightarrow \mathcal{L}(E + F) = L_E \cup L_F$$
$$\mathcal{L}(E) = L_E \Rightarrow \mathcal{L}(E^*) = (L_E)^*$$

Para qualquer conjunto A considera-se $A^+ \stackrel{\text{def}}{=} A^* \setminus \{\varepsilon\}$

Propriedades

Seja L uma linguagem
(conjunto eventualmente vazio de palavras sobre um alfabeto).

- Como ε é o elemento neutro da concatenação de palavras, $\{\varepsilon\}^* = \{\varepsilon\}$
- O vazio é o elemento absorvente da concatenação: $\emptyset \cdot L = \emptyset = L \cdot \emptyset$
- O fecho de Kleene do vazio é a linguagem ε $\emptyset^* = \{\varepsilon\}$
- O fecho de Kleene é idempotente: $(L^*)^* = L^*$
- Dados dois símbolos distintos a e b, tem-se $\{a, b\}^* = \{a\}^*(\{b\}\{a\}^*)^*$

Naturalmente, sendo as linguagens conjuntos, as propriedades usuais verificam-se (*e.g.* a união é comutativa e associativa, tem o vazio como elemento neutro, etc.).

Obtenção da linguagem denotada por uma expressão

Uma expressão regular define univocamente uma linguagem.
O contrário não é verdade: dada linguagem tem um número infinito contável de expressões que a define (por causa das equivalências: $a = a\,\varepsilon$, $\emptyset = \emptyset a$, $a + b = b + a$, etc.).
Exemplo: linguagem de $(a + b)^* a$ (ou de $(b + a)^* a$)

$$\begin{aligned}
\mathcal{L}((a+b)^*a) &= \mathcal{L}((a+b)^*) \cdot \mathcal{L}(a) \\
&= \mathcal{L}((a+b)^*) \cdot \{a\} \\
&= \mathcal{L}(a+b)^* \cdot \{a\} \\
&= (\mathcal{L}(a) \cup \mathcal{L}(b))^* \cdot \{a\} \\
&= (\{a\} \cup \{b\})^* \cdot \{a\} \\
&= \{a,b\}^* \cdot \{a\} \\
&= \{wa \mid w \in \{a,b\}^*\}
\end{aligned}$$

É o conjunto das palavras sobre $\{a, b\}$ que terminam em a

Linguagens regulares versus AFDs

Concisão

Uma expressão regular identifica univoca e concisamente uma linguagem.

Considere o alfabeto Σ_{AFILE}

- A linguagem aceite pelo AFD *AFILE* é denotada pela expressão regular $(\texttt{open}(\texttt{read} + \texttt{write})^*\texttt{close})^*$
- Mas como verificar se uma palavra pertence ao conjunto denotado pela expressão?
- O AFD, como modelo operacional, permite fazer isso facilmente: executa-se a função de transição estendida!

Como verificar se dada palavra pertence a uma linguagem?

Análise de sub-palavras

$w \in \mathcal{L}(E)$ com $E \in RegExp(\Sigma)$, se:

- $E = E_1 E_2$, $w = w_1 w_2$ e $w_1 \in \mathcal{L}(E_1)$ e $w_2 \in \mathcal{L}(E_2)$
- $E = E_1 + E_2$, $w \in \mathcal{L}(E_1)$ ou $w \in \mathcal{L}(E_2)$
- $E = F^*$, $w \in \mathcal{L}_n(F)$, para algum n.

Exemplos

- $\varepsilon \in \mathcal{L}(a^* + b^*)$, pois $\varepsilon \in \mathcal{L}(a^*)$ porque $\varepsilon \in \mathcal{L}_0(a)$
- $\varepsilon \in \mathcal{L}(a^* b^*)$, pois $\varepsilon = \varepsilon\varepsilon$, $\varepsilon \in \mathcal{L}(a^*)$ e $\varepsilon \in \mathcal{L}(b^*)$
- $a \in \mathcal{L}(a^*)$, pois $a \in \mathcal{L}_1(a)$
- $bb \in \mathcal{L}(a^*) \cdot \mathcal{L}(b^*)$, pois $bb = \varepsilon bb$, $\varepsilon \in \mathcal{L}(a^*)$ e $bb \in \mathcal{L}(b^*)$ porque $bb \in \mathcal{L}_2(b)$

Um exemplo mais elaborado

Note-se que $E^* = \varepsilon + E^+$ e $E^+ = E\,E^* = E^*E$

$abc \in \mathcal{L}((a+b)^*(c+d))$?

- $abc \in \mathcal{L}((a+b)^*(c+d))$ porque $ab \in \mathcal{L}((a+b)^*)$ e $c \in \mathcal{L}(c+d)$
- $c \in \mathcal{L}(c+d)$ porque $\mathcal{L}(c+d) = \mathcal{L}(c) \cup \mathcal{L}(d)$ e $c \in L(c) = \{c\}$
- $ab \in \mathcal{L}((a+b)^*)$ porque $ab \in \mathcal{L}(\varepsilon + (a+b)^+)$ uma vez que $ab \in \mathcal{L}((a+b)^+)$
- $ab \in \mathcal{L}((a+b)^+)$ porque $a \in \mathcal{L}((a+b))$ e $b \in \mathcal{L}((a+b)^*)$
- $b \in \mathcal{L}((a+b)^*)$ porque $b \in \mathcal{L}(\varepsilon + (a+b)^+)$ uma vez que $b \in \mathcal{L}((a+b)^+)$
- $b \in \mathcal{L}((a+b)(a+b)^*)$ porque $b = b\varepsilon$, $b \in \mathcal{L}(a+b)$ e $\varepsilon \in \mathcal{L}((a+b)^*)$

Como verificar que dada palavra não pertence a uma linguagem?

- Tem que se mostrar que não há nenhuma decomposição que permita provar que pertence...
- Note-se que, para um alfabeto Σ, se dado símbolo $a \notin \Sigma$, então $w_1 a w_2 \notin \Sigma^*$, para $\{w_1, w_2\} \subseteq \Sigma^*$.

Exemplos

- $\varepsilon \notin \mathcal{L}(a^+)$ pois $\mathcal{L}(a^+) = \mathcal{L}(a) \cdot \mathcal{L}(a^*)$, $\varepsilon = \varepsilon\varepsilon$, mas $\varepsilon \notin \mathcal{L}(a)$
- $abc \notin \mathcal{L}((a+b)^+)$, pois considerando:
  - $(a+b)^+ = (a+b)^*(a+b)$ e apesar de $ab \in \mathcal{L}((a+b)^*)$, tem-se que $c \notin \mathcal{L}(a+b) = \{a, b\}$
  - $(a+b)^+ = (a+b)(a+b)^*$ e apesar de $a \in \mathcal{L}(a+b)$, tem-se que $bc \notin \mathcal{L}((a+b)^*) = \{a, b\}^*$