1 Linguagens regulares

1.1 Alfabetos

Um alfabeto é denotado por Σ. Exemplos:

$$\Sigma = \{\, 0, 1 \,\} \qquad \Sigma = \{\, \mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e} \,\} \qquad \Sigma = \{\, \triangle, \mathrm{O}, \square, \mathrm{X} \,\}$$

1.2 Palavras

Uma palavra (ou cadeia) é uma sequência de zero ou mais símbolos do alfabeto.

Notação:

$$\begin{aligned}
&\lambda = \varnothing \\
&0^4 = 0000 \\
&\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\} \\
&\Sigma^* = \bigcup_{i \in \mathbb{N}} \Sigma^i \quad \text{conjunto de todas as possíveis palavras deste alfabeto.}
\end{aligned}$$

Concatenação:

$$x = 00 \qquad y = 11$$
$$xy = 0011$$

Reverso:

$$(xy)^{\mathrm{R}} = 1100$$

Observação: uma palavra w é um palíndromo se, e somente se $w^{\mathrm{R}} = w$.

1.3 Linguagens

Uma linguagem é um conjunto de palavras $L \subseteq \Sigma^*$.

Operações:

$$L_1 L_2 = \{\, xy \mid x \in L_1,\, y \in L_2 \,\}$$

$$\begin{aligned}
&L^0 = \{\, \lambda \,\} \\
&L^1 = L \\
&L^2 = LL \\
&L^* = \bigcup_{i \in \mathbb{N}} L^i \quad \text{Fecho de Kleene} \\
&L^+ = \bigcup_{i \in \mathbb{N}^*} L^i \\
&\varnothing^* = \{\, \lambda \,\} \\
&\varnothing^+ = \varnothing
\end{aligned}$$

Teorema: As linguagens regulares são fechadas sob as seguintes operações

- União
- Interseção
- Complemento
- Concatenação
- Fecho de Kleene

2 Autômatos finitos

2.1 Determinísticos

Um autômato finito determinístico é definido por:

Q	Um conjunto finito de estados.
Σ	Um alfabeto finito.
$\delta : Q \times \Sigma \to Q$	Uma função de transição.
$q_o \in Q$	Um estado inicial.
$F \subseteq Q$	Um conjunto de estados finais.

Notação:

$L(M) = A$ A linguagem reconhecida pelo autômato M.

$L(M : F = \varnothing) = \varnothing$

$\hat{\delta} : Q \times \Sigma^* \to Q$

$\hat{\delta}(e, w)$: aplicação sucessiva de δ aos símbolos de w.

Ainda, nos autômatos existe um estado especial, denonimado $\emptyset$, que aprisiona todas as transições omitidas.

2.1.1 Computação

Seja $M = (Q, \Sigma, \delta, q_0, F)$ um autômato finito determinístico, e $w \in \Sigma^*$.
Dizemos que M aceita w se existe uma **sequência** de estados $r_1, \ldots, r_n \in Q$ satisfazendo:

1. $r_0 = q_0$
2. $\forall\, i \in [0, n) : \delta(r_i,\, w_{i+1}) = r_{i+1}$
3. $r_n \in F$

Um autômato M reconhece uma linguagem L se $\forall\, w \in L : M$ aceita w.
Uma linguagem é regular se existe um autômato finito que a reconhece.

2.1.2 Minimização de estados

Dois estados e e e' são **equivalentes** se

$$\hat{\delta}(e, w) \in F \iff \hat{\delta}(e', w) \in F$$

O algoritmo de minimização, então, é:

1. Produza uma partição $P_0 = \{F,\, Q - F\}$ de Q, separando os estados finais dos não finais.
2. Para cada bloco de estados B na partição P_i, cada símbolo s do alfabeto Σ, e cada par de estados $(e,\, e')$ contidos no bloco B:
   (a) Sejam $d = \delta(e, s)$ e $d' = \delta(e', s)$ os estados para os quais o AFD transita quando lê o sımbolo s a partir dos estados e e e', respectivamente.
   (b) Se d e d' pertencem a blocos diferentes na partiçãoo P_i, então os estados e e e' não são equivalentes, e devem ser separados na partição P_{i+1}.
3. Se a partição P_{i+1} for diferente da partiçãoo P_i, repita o passo 2.
4. O autômato mínimo é construído de tal forma que seus estados são os blocos da última partição P produzida.

2.2 Não determinísticos

Um autômato finito não determinístico é definido por:

Q	Um conjunto finito de estados.
Σ	Um alfabeto finito.
$\delta : Q \times \Sigma \to \mathcal{P}(Q)$	Uma função de transição.
$I \subseteq Q$	Um conjunto de estados iniciais.
$F \subseteq Q$	Um conjunto de estados finais.

Sendo $a \in \Sigma$ um símbolo, e $w \in \Sigma^*$ uma palavra, define-se a função de transição estendida:

$$\begin{aligned}
&\hat{\delta} : Q \times \Sigma^* \to \mathcal{P}(Q)\\
&\hat{\delta}(\emptyset,\, w) = \{\emptyset\}\\
&\hat{\delta}(X,\, \lambda) = X\\
&\hat{\delta}(X,\, aw) = \hat{\delta}\left(\bigcup_{l \in X} \delta(l,\, a),\, w\right)
\end{aligned}$$

Teorema: Todo AFN possui um AFD equivalente.
Por construção:

$$\begin{aligned}
&Q = \mathcal{P}(Q_{\text{afn}})\\
&\Sigma = \Sigma_{\text{afn}}\\
&\delta(X, a) = \bigcup_{l \in X} \delta_{\text{afn}}(l,\, a)\\
&q_o = I_{\text{afn}}\\
&F = \{X \subseteq Q_{\text{afn}} \mid X \cap F \neq \varnothing\}
\end{aligned}$$

2.2.1 Transições λ

Um autômato finito não determinístico com transições λ introduz a possibilidade de transições sem a consumação de símbolos.

$$\begin{aligned}
&Q = Q_{\text{afn}}\\
&\Sigma = \Sigma_{\text{afn}}\\
&\delta : Q \times \Sigma_\lambda \to \mathcal{P}(Q)\\
&I = I_{\text{afn}}\\
&F = F_{\text{afn}}
\end{aligned}$$

Onde $\Sigma_\lambda = \Sigma \cup \{\lambda\}$.

Os estados para os quais se transita sem consumir símbolos é definido pelo fecho λ:

$$\mathcal{F}_\lambda : \mathcal{P}(Q) \to \mathcal{P}(Q)$$

Teorema: O fecho lambda de um estado é pelo menos o próprio estado.

$$\forall\, X \in Q : X \in \mathcal{F}_\lambda(\{X\})$$

Assim, define-se a função de transição estendida:

$$\begin{aligned}
&\hat{\delta} : Q \times \Sigma_\lambda^* \to \mathcal{P}(Q) \\
&\hat{\delta}(\varnothing, w) = \varnothing \\
&\hat{\delta}(X, \lambda) = \mathcal{F}_\lambda(X) \\
&\hat{\delta}(X, ay) = \hat{\delta}\left(\bigcup_{Y \in \mathcal{F}_\lambda(X)} \delta(Y, a), \;\; y \right)
\end{aligned}$$

Teorema: Todo AFNλ possui um AFN equivalente.
Por construção:

$$\begin{aligned}
&Q = Q_{\text{afn}\lambda} \\
&\Sigma = \Sigma_{\text{afn}\lambda} \\
&\delta = \mathcal{F}_\lambda \circ \delta_{\text{afn}\lambda} \\
&I = \mathcal{F}_\lambda\left(I_{\text{afn}\lambda}\right) \\
&F = F_{\text{afn}\lambda}
\end{aligned}$$

2.3 Com pilha

Um autômato finito com pilha não determinístico é definido por:

Q	Um conjunto finito de estados.
Σ	Um alfabeto finito.
Γ	Um alfabeto de pilha finito.
$\delta : Q \times \Sigma_\lambda \times \Gamma_\lambda \to \mathcal{P}\left(\Gamma^* \times Q\right)$	Uma função de transição.
$I \subseteq Q$	Um conjunto de estados iniciais.
$F \subseteq Q$	Um conjunto de estados finais.

Em cada transição, o elemento do topo da pilha é retirado para a função de transição, que por sua vez devolve uma sequência de elementos a serem empilhados, além do estado transitado.

Um AFPN aceita uma palavra se ao consumí-la, encerra-se em um estado final **com a pilha vazia**.

3 Expressões regulares

Uma expressão regular pode ser uma das seguintes formas, cada qual com a linguagem correspondente:

λ	$\{\lambda\}$
$\varnothing$	$\varnothing$
a	$\{a\}$
$R_1 + R_2$	$L(R_1) \cup L(R_2)$
$R_1 R_2$	$L(R_1) \cdot L(R_2)$
R^*	$L(R)^*$

Operações:

$$\begin{aligned}
&R^+ = RR^* \\
&R^0 = \lambda \\
&R^n = RR^{(n-1)}
\end{aligned}$$

4 Linguagens irregulares

Nas linguagens regulares, têm se o **lema do bombardeamento**:
Se L é uma linguagem regular, então

$$\begin{aligned}
&\exists\, k \in \mathbb{N}^* : \\
&\quad \forall\, z \in L, |z| \geq k : \\
&\quad\quad \exists\, u, v, w : \\
&\quad\quad\quad 1.\ z = uvw \\
&\quad\quad\quad 2.\ |uv| \leq k \\
&\quad\quad\quad 3.\ v \neq \lambda \\
&\quad\quad\quad 4.\ \forall\, i \in \mathbb{N}^* : \big(uv^i w\big) \in L
\end{aligned}$$

O lema pode ser utilizado para provar que uma dada linguagem não é regular.

5 Linguagens livres de contexto

Uma linguagem livre de contexto é uma linguagem que pode ser denotada por uma gramática livre de contexto.

5.1 Gramáticas livres de contexto

Uma gramática livre de contexto é definida por:

V	Um conjunto finito de variáveis.
Σ	Um alfabeto finito.
R	Um conjunto de regras.
$S \in V$	Uma variável inicial.

As regras são constituídas da seguinte forma:

1. O lado esquerdo de uma regra é constituído por uma única variável.
2. O lado direito é constituído por uma combinação de terminais e variáveis.

Por convenção a variável inicial é a variável alvo da primeira regra.

Exemplo:

$$\begin{aligned}
G = \big(\{A, B\}, \{0, 1, 5\},\ R,\ A\big) \\
R: \quad A \to 0A1 \\
A \to B \\
B \to 5
\end{aligned}$$

Lema: Para toda GLC, existe um AFPN que a reconhece.

$$\begin{aligned}
&G = (V, \Sigma, R, S) \\
&M = \big(\{i, f\},\ \Sigma,\ (V \cup \Sigma),\ \delta,\ \{i\},\ \{f\}\big)
\end{aligned}$$

$$\begin{aligned}
\delta(i, \lambda, \lambda) &= \big\{[f, S]\big\} \\
\delta(f, \lambda, X) &= \big\{[f, \beta] \,\big|\, (X \to \beta) \in R\big\} \\
\delta(f, a, a) &= \big\{[f, \lambda]\big\}
\end{aligned}$$

Lema: Para todo AFPN, existe uma GLC equivalente.

Nas linguagens livres de contexto, têm se o **lema do bombardeamento**:
Se L é uma linguagem livre de contexto, então

$$\begin{aligned}
&\exists\, k \in \mathbb{N}^* : \\
&\quad \forall\, w \in L, |w| \geq k : \\
&\qquad \exists\, u, v, x, y, z : \\
&\qquad\quad 1.\ w = uvxyz \\
&\qquad\quad 2.\ |vxy| \leq k \\
&\qquad\quad 3.\ vy \neq \lambda \\
&\qquad\quad 4.\ \forall\, i \in \mathbb{N}^* : \left(uv^ixy^iz\right) \in L
\end{aligned}$$

O lema pode ser utilizado para provar que uma dada linguagem não é livre de contexto.

5.2 Propriedade de fechamento

A classe das linguagens livre de contexto são fechadas sob as seguintes operações

- União
- Concatenação
- Fecho de Kleene