Teoria da Computação

Engenharia Informática

Rodrigo Santos

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

II Semestre - 2023/2024

Contents

1 Demonstrações

1.1 $(A_i)_{i\in\mathbb{N}} = A_1, A_2, \ldots$ uma sequencia de conjuntos contáveis. Então $\bigcup\limits_{i\in\mathbb{N}} A_i$ também é contável.

Se cada conjunto A_i é contável, Então existe um função injetiva $g_i : A_i \to \mathbb{N}$ para cada $i \in \mathbb{N}$.Definimos a função $f : \bigcup\limits_{i\in\mathbb{N}} A_i \to \mathbb{N}$ tal que $f(x) = g_i(x)$ se $x \in A_i$. f é injetiva pois g_i é injetiva para todo $i \in \mathbb{N}$. Logo, $\bigcup\limits_{i\in\mathbb{N}} A_i$ é contável.

1.2 Se $L1$ e $L2$ são regulares, então $L1 \cap L2$ também é regular.

Como L_1 e L_2 são regulares, então existem autómatos finitos deterministas $(AFD's)$ $M_1 = (S_1, \Sigma, \delta_1, s_1, F_1)$ e $M_2 = (S_2, \Sigma, \delta_2, s_2, F_2)$ completos, que aceitam L_1 e L_2 respetivamente. Vamos construir um AFD $M = (S, \Sigma, \delta, s, F)$ que aceita $L_1 \cap L_2$. Vamos seguir a estratégia em que dado um input w, simulamos as computações de M_1 e M_2 em w, lado-a-lado, e aceitamos w se ambas as simulações aceitarem w. Para isso, precisamos de saber os estados atuais de M_1 e M_2 a cada momento da computação de w. Definimos então $S = S_1 \times S_2$ em que cada par representa um estado atual possível de M. Se M_1 está em q_1 e M_2 está em q_2, então o estado atual de M é o tuplo $(q_1, q_2) \in S$ com $q_1 \in S_1$ e $q_2 \in S_2$. Se M está em $(q_1, q_2) \in S$ e lê o simbolo $a \in \Sigma$ então temos que atualizar o estado q_1 para $\delta_1(q_1, a)$ e o estado q_2 para $\delta_2(q_2, a)$. Assim, definimos a função de transição δ como $\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$. Para o estado inicial de M queremos escolher o par (s_1, s_2) (estados iniciais de M_1 e M_2 respetivamente), ou seja $s = (s_1, s_2)$. Falta agora definir F. Queremos aceitar w qualquer, se ambas as computações em simultâneo aceitarem w (Ou seja terminem num estado final de M_1 e num estado final de M_2). Assim vamos escolher o conjunto dos pares (q_1, q_2) em que $q_1 \in F_1$ e $q_2 \in F_2$, ou seja $F = F_1 \times F_2$. Assim, o AFD $M = (S, \Sigma, \delta, s, F)$ aceita $L_1 \cap L_2$, mas precisamos de demonstrar que $L(M) = L_1 \cap L_2$. Seja $w \in \Sigma^*$ qualquer. A sequência de estados gerada por w em M_1 e M_2 pode ser descrita como $r_0^i, r_1^i, \ldots, r_n^i$ para $i \in 1, 2$. Se $w \in L_1 \cap L_2$ então por definição temos $r_n^1 \ \wedge \ r_n^2$ pertencentes a F_1 e F_2 respetivamente. Concluimos então que $\delta(w) = (r_n^1, r_n^2) \in F$. Como w é arbitrário, então $L_1 \cap L_2 \subseteq L(M)$. Vamos ao complementar, se $w \notin L(M)$, então temos $r_n^1 \notin F_1 \ \wedge \ r_n^2 \notin F_2$. O que se traduz para $(r_n^1, r_n^2) \notin F$, e portanto $w \notin L(M)$ que leva a $L(M) \subseteq L_1 \cap L_2$. Concluimos então que $L(M) = L_1 \cap L_2$ e portanto $L_1 \cap L_2$ é regular.

2 Exercícios

2.1 Problem set 1

2.1.1 Exercício 1

(a) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Para demonstrar a igualdade temos que provar que $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$ e $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$. Vamos começar por provar que $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$. Seja $x \in A \cup (B \cap C)$. Então $x \in A$ ou $x \in B \cap C$. Se $x \in A$ então $x \in A \cup B$ e $x \in A \cup C$. Se $x \in B \cap C$ então $x \in B$ e $x \in C$. Assim, $x \in A \cup B$ e $x \in A \cup C$. Portanto, $x \in (A \cup B) \cap (A \cup C)$. Vamos agora provar que $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$. Seja $x \in (A \cup B) \cap (A \cup C)$. Então $x \in A \cup B$ e $x \in A \cup C$. Assim, $x \in A$ ou $x \in B$ e $x \in A$ ou $x \in C$. Se $x \in A$ então $x \in A \cup (B \cap C)$. Se $x \in B$ e $x \in C$ então $x \in B \cap C$. Assim, $x \in A \cup (B \cap C)$. Portanto, $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$. Concluímos então que $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.

(b) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Para demonstrar a igualdade temos que provar que $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$ e $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$. Vamos começar por provar que $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$. Seja $x \in A \cap (B \cup C)$. Então $x \in A$ e $x \in B \cup C$. Assim, $x \in A$ e $x \in B$ ou $x \in A$ e $x \in C$. Portanto, $x \in A \cap B$ ou

2.2 Problem set 3

2.2.1 Exercício 5

Sejam $L1$ e $L2$ linguagens regulares sobre o mesmo alfabeto Σ. Mostre que $L1 \cap L2$ também é regular.

Como L_1 e L_2 são regulares, então existem autómatos finitos deterministas $(AFD's)$ $M_1 = (S_1, \Sigma, \delta_1, s_1, F_1)$ e $M_2 = (S_2, \Sigma, \delta_2, s_2, F_2)$ completos, que aceitam L_1 e L_2 respetivamente. Vamos construir um AFD $M = (S, \Sigma, \delta, s, F)$ que aceita $L_1 \cap L_2$. Vamos seguir a estratégia em que dado um input w, simulamos as computações de M_1 e M_2 em w, lado-a-lado, e aceitamos w se ambas as simulações aceitarem w. Para isso, precisamos de saber os estados atuais de M_1 e M_2 a cada momento da computação de w. Definimos então $S = S_1 \times S_2$ em que cada par representa um estado atual possível de M. Se M_1 está em q_1 e M_2 está em q_2, então o estado atual de M é o tuplo $(q_1, q_2) \in S$ com $q_1 \in S_1$ e $q_2 \in S_2$. Se M está em $(q_1, q_2) \in S$ e lê o simbolo $a \in \Sigma$ então temos que atualizar o estado q_1 para $\delta_1(q_1, a)$ e o estado q_2 para $\delta_2(q_2, a)$. Assim, definimos a função de transição δ como $\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$. Para o estado inicial de M queremos escolher o par (s_1, s_2) (estados iniciais de M_1 e M_2 respetivamente), ou seja $s = (s_1, s_2)$. Falta agora definir F. Queremos aceitar w qualquer, se ambas as computações em simultâneo aceitarem w (Ou seja terminem num estado final de M_1 e num estado final de M_2). Assim vamos escolher o conjunto dos pares (q_1, q_2) em que $q_1 \in F_1$ e $q_2 \in F_2$, ou seja $F = F_1 \times F_2$. Assim, o AFD $M = (S, \Sigma, \delta, s, F)$ aceita $L_1 \cap L_2$, mas precisamos de demonstrar que $L(M) = L_1 \cap L_2$. Seja $w \in \Sigma^*$ qualquer. A sequência de estados gerada por w em M_1 e M_2 pode ser descrita como $r_0^i, r_1^i, \ldots, r_n^i$ para $i \in 1, 2$. Se $w \in L_1 \cap L_2$ então por definição temos $r_n^1 \;\wedge\; r_n^2$ pertencentes a F_1 e F_2 respetivamente. Concluimos então que $\delta(w) = (r_n^1, r_n^2) \in F$. Como w é arbitrário, então $L_1 \cap L_2 \subseteq L(M)$. Vamos ao complementar, se $w \notin L(M)$, então temos $r_n^1 \notin F_1 \ \wedge \ r_n^2 \notin F_2$. O que se traduz para $(r_n^1, r_n^2) \notin F$, e portanto $w \notin L(M)$ que leva a $L(M) \subseteq L_1 \cap L_2$. Concluimos então que $L(M) = L_1 \cap L_2$ e portanto $L_1 \cap L_2$ é regular.

2.2.2 Exercício 6

Dada uma string $w = w_1 w_2 \dots w_n \in \Sigma^*$ definimos o seu reverso $rev(w) = w_n w_{n-1} \dots w_2 w_1$. Para uma linguagem $L \subseteq \Sigma^*$, definimos $rev(L) = \{rev(w) \mid w \in L\}$. Mostre que se L é regular então $rev(L)$ também é regular.

Com um AFD não é fácil de definir o reverso de uma linguagem, mas com um AFN é possível. Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFN que aceita a linguagem L.

2.2.3 Exercício 7

Seja $L_n = \{0^k \mid \textbf{k é múltiplo de n}\}$. Mostre que L_n é regular para qualquer $n \in \mathbb{N}^+$

Fixamos $n \in \mathbb{N}^+$ qualquer. Descrevemos o automato finito determinista (AFD) que reconhece a linguagem L_n como, $M = (S, \Sigma, \delta, s, F)$. O conjunto de estados $S = \{q_0, q_1, \dots, q_{n-1}\}$, o alfabeto $\Sigma = \{0\}$, o estado inicial $s = q_0$ e $F = \{q_0\}$. Falta então definir δ. $\delta(qi, 0) = q_{(i+1) \ mod \ n}$. Vamos mostrar que M reconhece L_n. Seja $w = 0^k \in L_n$, então k é múltiplo de n e $k = n \cdot m$ para algum $m \in \mathbb{N}$. Então, $\delta(q_0, 0) = q_0$, $\delta(q_0, 0) = q_0$, ..., $\delta(q_0, 0) = q_0$. Portanto, $q_0 \in F$ e M aceita w. Seja $w = 0^k \in \Sigma^* - L_n$, então k não é múltiplo de n e $k = n \cdot m + r$ para algum $m \in \mathbb{N}$ e $r \in \{1, 2, \dots, n-1\}$. Então, $\delta(q_0, 0) = q_r$ e $q_r \notin F$. Portanto, M rejeita w. Portanto, M reconhece L_n e L_n é regular.

2.2.4 Exercício 8

Para uma linguagem $L \subseteq \Sigma^*$ definimos a operação:

noPrefix$(L) = \{w \in L \mid$ **nenhum prefixo próprio de**w**pertence a** $L\}$

Mostre que se L é regular então noPrefix(L) também é regular.

Seja $M = (S, \Sigma, \delta, s, F)$ um AFD com função de transição total que aceita a linguagem L, ou seja $L = L(M)$. Queremos construir um AFD $M' = (S', \Sigma, \delta', s', F')$ que aceita a linguagem noPrefix(L). Para tal queremos construir M' de modo a obtermos apenas transições que vão para estados finais e não ter nenhuma transição que vem de um estado final.