

AUTOMI E LINGUAGGI FORMALI  
ESERCIZI DI PREPARAZIONE AGLI ESAMI

**Prima parte**

1. Dati i linguaggi  $A$  e  $B$ , lo *shuffle perfetto* di  $A$  e  $B$  è il linguaggio

$$\{w \mid w = a_1b_1a_2b_2 \dots a_kb_k, \text{ dove } a_1a_2 \dots a_k \in A \text{ e } b_1b_2 \dots b_k \in B, \text{ ogni } a_i, b_i \in \Sigma\}$$

Mostrare che la classe dei linguaggi regolari è chiusa rispetto allo shuffle perfetto, cioè che se  $A$  e  $B$  sono linguaggi regolari allora anche il loro shuffle perfetto è un linguaggio regolare.

2. Sia  $A$  un linguaggio, e sia  $DROPOUT(A)$  come il linguaggio contenente tutte le stringhe che possono essere ottenute togliendo un simbolo da una stringa di  $A$ :

$$DROPOUT(A) = \{xz \mid xyz \in A \text{ dove } x, z \in \Sigma^* \text{ e } y \in \Sigma\}.$$

Mostrare che la classe dei linguaggi regolari è chiusa rispetto all'operazione  $DROPOUT$ , cioè che se  $A$  è un linguaggio regolare allora  $DROPOUT(A)$  è un linguaggio regolare.

3. Per una stringa  $w = w_1w_2 \dots w_n$ , l'*inversa* di  $w$  è la stringa  $w^R = w_n \dots w_2w_1$ . Per ogni linguaggio  $A$ , sia  $A^R = \{w^R \mid w \in A\}$ . Mostrare che se  $A$  è regolare allora lo è anche  $A^R$ .
4. Sia  $A/b = \{w \mid wb \in A\}$ . Mostrare che se  $A$  è un linguaggio regolare e  $b \in \Sigma$ , allora  $A/b$  è regolare.
5. Sia  $A/B = \{w \mid wx \in A \text{ per qualche } x \in B\}$ . Mostrare che se  $A$  è un linguaggio regolare e  $B$  un linguaggio qualsiasi, allora  $A/B$  è regolare.
6. Il pumping lemma afferma che ogni linguaggio regolare ha una lunghezza del pumping  $p$ , tale che ogni stringa del linguaggio può essere iterata se ha lunghezza maggiore o uguale a  $p$ . La *lunghezza minima del pumping* per un linguaggio  $A$  è il più piccolo  $p$  che è una lunghezza del pumping per  $A$ . Per ognuno dei seguenti linguaggi, dare la lunghezza minima del pumping e giustificare la risposta.

- |                            |   |                   |
|----------------------------|---|-------------------|
| (a) $110^*$                | (g) $10(11^*0)^*0$                          | (m) $\varepsilon$ |
| (b) $1^*0^*1^*$            | (h) $101101$                                | (n) $1^*01^*01^*$ |
| (c) $0^*1^*0^*1^* + 10^*1$ | (i) $\{w \in \Sigma^* \mid w \neq 101101\}$ | (o) $1011$        |
| (d) $(01)^*$               | (j) $0001^*$                                | (p) $\Sigma^*$    |
| (e) $\emptyset$            | (k) $0^*1^*$                                |                   |
| (f) $0^*01^*01^*$          | (l) $001 + 0^*1^*$                          |                   |

7. Sia  $\Sigma = \{0, 1\}$ , e considerate il linguaggio

$$D = \{w \mid w \text{ contiene un ugual numero di occorrenze di } 01 \text{ e di } 10\}$$

Mostrare che  $D$  è un linguaggio regolare.

8. Sia  $\Sigma = \{0, 1\}$ .

- Mostrare che il linguaggio  $A = \{0^k u 0^k \mid k \geq 1 \text{ e } u \in \Sigma^*\}$  è regolare.
- Mostrare che il linguaggio  $B = \{0^k 1 u 0^k \mid k \geq 1 \text{ e } u \in \Sigma^*\}$  non è regolare.

9. Dimostrare che i seguenti linguaggi non sono regolari.

- (a)  $\{0^n 1^m 0^m \mid m, n \geq 0\}$   
 (b)  $\{0^n 1^m \mid n \neq m\}$   
 (c)  $\{w \in \{0, 1\}^* \mid w \text{ non è palindroma}\}$   
 (d)  $\{wtw \mid w, t \in \{0, 1\}^+\}$ , dove  $\{0, 1\}^+$  è l'insieme di tutte le stringhe binarie di lunghezza maggiore o uguale a 1

10. Per ogni linguaggio  $A$ , sia  $SUFFIX(A) = \{v \mid uv \in A \text{ per qualche stringa } u\}$ . Mostrare che la classe dei linguaggi context-free è chiusa rispetto all'operazione di  $SUFFIX$ .

11. Dimostrare che i seguenti linguaggi sono context-free. Salvo quando specificato diversamente, l'alfabeto è  $\Sigma = \{0, 1\}$ .
- $\{w \mid w \text{ contiene almeno tre simboli uguali a } 1\}$
  - $\{w \mid \text{la lunghezza di } w \text{ è dispari}\}$
  - $\{w \mid w \text{ inizia e termina con lo stesso simbolo}\}$
  - $\{w \mid \text{la lunghezza di } w \text{ è dispari e il suo simbolo centrale è } 0\}$
  - $\{w \mid w = w^R, \text{ cioè } w \text{ è palindroma}\}$
  - $\{w \mid w \text{ contiene un numero maggiore di } 0 \text{ che di } 1\}$
  - Il complemento di  $\{0^n 1^n \mid n \geq 0\}$
  - Sull'alfabeto  $\Sigma = \{0, 1, \#\}$ ,  $\{w\#x \mid w^R \text{ è una sottostringa di } x \text{ e } w, x \in \{0, 1\}^*\}$
  - $\{x\#y \mid x, y \in \{0, 1\}^* \text{ e } x \neq y\}$
  - $\{xy \mid x, y \in \{0, 1\}^* \text{ e } |x| = |y| \text{ ma } x \neq y\}$
  - $\{a^i b^j \mid i \neq j \text{ e } 2i \neq j\}$
12. Se  $A$  e  $B$  sono linguaggi, definiamo  $A \circ B = \{xy \mid x \in A, y \in B \text{ e } |x| = |y|\}$ . Mostrare che se  $A$  e  $B$  sono linguaggi regolari, allora  $A \circ B$  è un linguaggio context-free.
13. Dimostrare che se  $G$  è una CFG in forma normale di Chomsky, allora per ogni stringa  $w \in L(G)$  di lunghezza  $n \geq 1$ , ogni derivazione di  $w$  richiede esattamente  $2n - 1$  passi.

## Seconda parte

14. Un *automa a coda* è simile ad un automa a pila con la differenza che la pila viene sostituita da una coda. Una *coda* è un nastro che permette di scrivere solo all'estremità sinistra del nastro e di leggere solo all'estremità destra. Ogni operazione di scrittura (*push*) aggiunge un simbolo all'estremità sinistra della coda e ogni operazione di lettura (*pull*) legge e rimuove un simbolo all'estremità destra. Come per un PDA, l'input è posizionato su un nastro a sola lettura separato, e la testina sul nastro di lettura può muoversi solo da sinistra a destra. Il nastro di input contiene una cella con un blank che segue l'input, in modo da poter rilevare la fine dell'input. Un automa a coda accetta l'input entrando in un particolare stato di accettazione in qualsiasi momento. Mostra che un linguaggio può essere riconosciuto da un automa deterministico a coda se e solo se è Turing-riconoscibile.
15. Una *Macchina di Turing a sola scrittura* è una TM a nastro singolo che può modificare ogni cella del nastro al più una volta (inclusa la parte di input del nastro). Mostrare che questa variante di macchina di Turing è equivalente alla macchina di Turing standard.
16. Chiamiamo  $k$ -PDA un automa a pila dotato di  $k$  pile.
- Mostrare che i 2-PDA sono più potenti degli 1-PDA
  - Mostrare che i 2-PDA riconoscono esattamente la classe dei linguaggi Turing-riconoscibili
  - Mostrare che i 3-PDA *non* sono più potenti dei 2-PDA
17. Sia  $ALL_{DFA} = \{\langle A \rangle \mid A \text{ è un DFA e } L(A) = \Sigma^*\}$ . Mostrare che  $ALL_{DFA}$  è decidibile.
18. Sia  $A_{\varepsilon CFG} = \{\langle G \rangle \mid G \text{ è una CFG che genera } \varepsilon\}$ . Mostrare che  $A_{\varepsilon CFG}$  è decidibile.
19. Sia  $S_{REX} = \{\langle R, S \rangle \mid R, S \text{ sono espressioni regolari tali che } L(R) \subseteq L(S)\}$ . Mostrare che  $S_{REX}$  è decidibile.
20. Sia  $X = \{\langle M, w \rangle \mid M \text{ è una TM a nastro singolo che non modifica la porzione di nastro che contiene l'input } w\}$ .  $X$  è decidibile? Dimostrare la vostra risposta.
21. Sia  $E_{TM} = \{\langle M \rangle \mid G \text{ è una TM tale che } L(M) = \emptyset\}$ . Mostrare che  $\overline{E_{TM}}$ , il complemento di  $E_{TM}$ , è Turing-riconoscibile.
22. Mostrare che se  $A$  è Turing-riconoscibile e  $A \leq_m \overline{A}$ , allora  $A$  è decidibile.
23. Sia  $A$  un linguaggio. Dimostrare che  $A$  è Turing-riconoscibile *se e solo se* esiste un linguaggio decidibile  $B$  tale che  $A = \{x \mid \text{esiste } y \text{ tale che } \langle x, y \rangle \in B\}$ .
24.  $A \leq_m B$  e  $B$  è un linguaggio regolare implica che  $A$  è un linguaggio regolare? Perché sì o perché no?
25. Mostrare che se  $A$  è Turing-riconoscibile e  $A \leq_m \overline{A}$ , allora  $A$  è decidibile.

26. Sia  $J = \{w \mid w = 0x \text{ per qualche } x \in A_{TM} \text{ oppure } w = 1y \text{ per qualche } y \in \overline{A_{TM}}\}$ . Mostrare che sia  $J$  che  $\bar{J}$  non sono Turing-riconoscibili.

27. Sia  $J = \{w \mid w = 0x \text{ per qualche } x \in A_{TM} \text{ oppure } w = 1y \text{ per qualche } y \in \overline{A_{TM}}\}$ . Mostrare che sia  $J$  che  $\bar{J}$  non sono Turing-riconoscibili.

28. Un circuito Hamiltoniano in un grafo  $G$  è un ciclo che attraversa ogni vertice di  $G$  esattamente una volta. Stabilire se un grafo contiene un circuito Hamiltoniano è un problema NP-completo.

(a) Un *circuito Toniano* in un grafo  $G$  è un ciclo che attraversa *almeno la metà* dei vertici del grafo (senza ripetere vertici). Il *problema del circuito Toniano* è il problema di stabilire se un grafo contiene un circuito quasi Hamiltoniano. Dimostrare che il problema è NP-completo.

(b) Un *circuito quasi Hamiltoniano* in un grafo  $G$  è un ciclo che attraversa esattamente una volta tutti i vertici del grafo *tranne uno*. Il *problema del circuito quasi Hamiltoniano* è il problema di stabilire se un grafo contiene un circuito quasi Hamiltoniano. Dimostrare che il problema del circuito quasi Hamiltoniano è NP-completo.

29. Considera i seguenti problemi:

SETPARTITIONING =  $\{\langle S \rangle \mid S \text{ è un insieme di numeri interi che può essere suddiviso in due sottoinsiemi disgiunti } S_1, S_2 \text{ tali che la somma dei numeri in } S_1 \text{ è uguale alla somma dei numeri in } S_2\}$

SUBSETSUM =  $\{\langle S, t \rangle \mid S \text{ è un insieme di numeri interi, ed esiste } S' \subseteq S \text{ tale che la somma dei numeri in } S' \text{ è uguale a } t\}$

(a) Mostra che entrambi i problemi sono in NP.

(b) Mostra che SETPARTITIONING è NP-Hard usando SUBSETSUM come problema di riferimento.

(c) Mostra che SUBSETSUM è NP-Hard usando SETPARTITIONING come problema di riferimento.

30. Considerate la seguente variante del problema SETPARTITIONING, che chiameremo QUASIPARTITIONING: dato un insieme di numeri interi  $S$ , stabilire se può essere suddiviso in due sottoinsiemi disgiunti  $S_1$  e  $S_2$  tali che la somma dei numeri in  $S_1$  è uguale alla somma dei numeri in  $S_2$  *meno 1*. Dimostrare che il problema QUASIPARTITIONING è NP-completo.

31. “Colorare” i vertici di un grafo significa assegnare etichette, tradizionalmente chiamate “colori”, ai vertici del grafo in modo tale che nessuna coppia di vertici adiacenti condivida lo stesso colore. Il problema  $k$ -COLOR è il problema di trovare una colorazione di un grafo non orientato usando  $k$  colori diversi.

(a) Mostrare che il problema  $k$ -COLOR è in NP per ogni valore di  $k$

(b) Mostrare che 2-COLOR è in P

(c) Mostrare che 3-COLOR  $\leq_P$   $k$ -COLOR per ogni  $k > 3$

(d) Considerate la seguente variante del problema, che chiameremo ALMOST3COLOR: dato un grafo non orientato  $G$  con  $n$  vertici, stabilire se è possibile colorare i vertici di  $G$  con tre colori diversi, in modo che ci siano al più  $n/2$  coppie di vertici adiacenti dello stesso colore. Dimostrare che il problema ALMOST3COLOR è NP-completo.