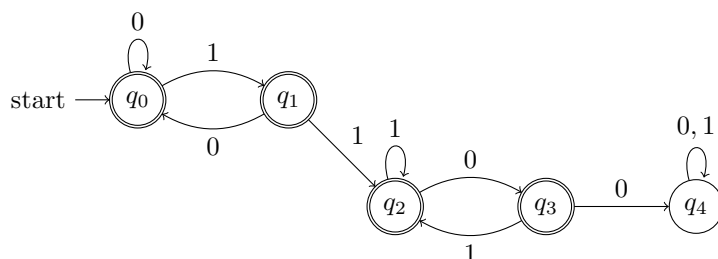


1. Definire un automa a stati finiti (di qualsiasi tipologia) che riconosca il linguaggio

$$L_1 = \{w \in \{0, 1\}^* \mid \text{ogni occorrenza di } 00 \text{ compare prima di ogni occorrenza di } 11\}$$



2. Definire una grammatica context-free che generi il linguaggio

$$L_2 = \{w \in \{0, 1\}^* \mid \text{il numero di } 0 \text{ è il doppio del numero di } 1\}$$

$$S \rightarrow 0S0S1 \mid 0S1S0 \mid 1S0S0 \mid SS \mid \varepsilon$$

3. Fornisci una descrizione a livello implementativo di una TM deterministica a nastro singolo che decide il linguaggio

$$L_3 = \{ww \mid w \in \{0, 1\}^*\}$$

M = “su input x , dove x è una stringa:

1. Scorre la stringa x per controllare se il numero di simboli è pari o dispari. Se è dispari, rifiuta.
2. Divide la stringa x in due parti uguali. Per farlo marca il primo simbolo di x con un pallino, poi va alla fine della stringa e marca l'ultimo simbolo con un pallino. Continua a marcare un carattere all'inizio e uno alla fine della stringa muovendosi a zig zag.
3. L'ultimo carattere marcato è la posizione di inizio della seconda metà della stringa. Toglie tutte le altre marcature e ritorna all'inizio della stringa.
4. sostituisce il simbolo all'inizio della stringa con $\#$, poi procede a destra e controlla se il simbolo marcato con un pallino è uguale al primo simbolo. Se sono diversi rifiuta.
5. Se i due simboli sono uguali, sostituisce il simbolo marcato con $\#$ e sposta la marcatura al simbolo successivo.
6. Torna all'inizio della stringa: se ci sono ancora simboli diversi da $\#$, ripeti da 4, altrimenti accetta.”

4. Fornisci un verificatore polinomiale per il seguente problema:

$$\text{DOUBLEHAMCIRCUIT} = \{\langle G \rangle \mid G \text{ è un grafo non orientato che contiene un ciclo che visita ogni vertice esattamente due volte e attraversa ogni arco esattamente una volta}\}$$

V = “su input $\langle G, C \rangle$, dove G è un grafo ed il certificato C è una sequenza di vertici (v_1, \dots, v_k) :

1. Controlla che ogni elemento di C sia un vertice del grafo;
2. controlla che C sia un ciclo ($v_1 = v_k$);
3. controlla che ogni vertice del grafo compaia 2 volte in C , tranne v_1 che deve comparire 3 volte;
4. controlla che (v_i, v_{i+1}) sia un arco del grafo per ogni $i = 1, \dots, k - 1$;
5. controlla che ogni arco compaia in C esattamente una volta;
6. se tutti i test sono superati accetta, altrimenti rifiuta.”