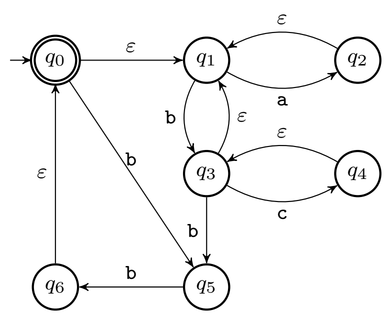
Automi e Linguaggi (M. Cesati)

Facoltà di Ingegneria, Università degli Studi di Roma Tor Vergata

Compito scritto del 13 settembre 2022

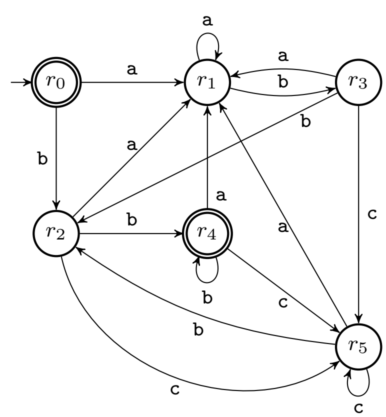
Esercizio 1 [6] Determinare un automa deterministico che riconosca il linguaggio generato dalla espressione regolare $((\mathtt{a}^*\mathtt{bc}^*)^*\mathtt{bb})^*$.

Soluzione: L'esercizio si può risolvere in modo totalmente meccanico derivando innanzi tutto un NFA dalla espressione regolare, e successivamente trasformando lo NFA in un DFA. Applicando qualche semplificazione allo NFA derivato dalla espressione regolare si ottiene:



Un automa deterministico equivalente è il seguente:

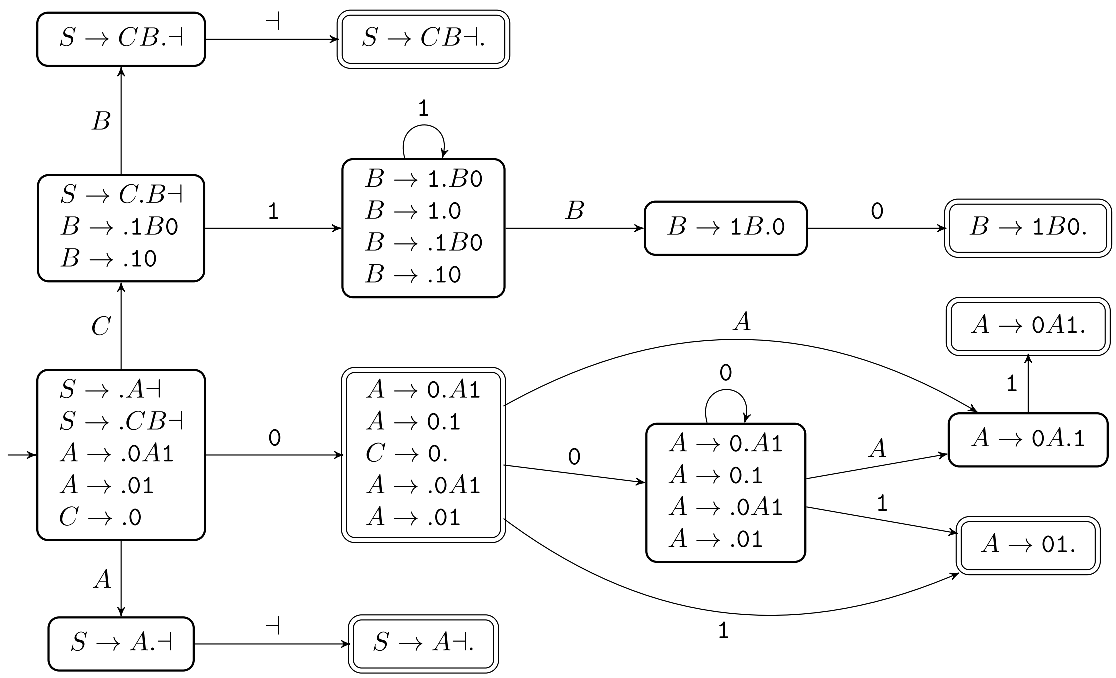
$$
\begin{aligned}
r_0 &= \{q_0, q_1\}\\
r_1 &= \{q_1, q_2\}\\
r_2 &= \{q_1, q_3, q_5\}\\
r_3 &= \{q_1, q_3\}\\
r_4 &= \{q_0, q_1, q_3, q_5, q_6\}\\
r_5 &= \{q_1, q_3, q_4\}
\end{aligned}
$$



Esercizio 2 [6] Determinare se la seguente grammatica CFG con variabile iniziale S è deterministica:

$$
\begin{array}{lcl}
S \to & A\dashv & | \quad CB\dashv \\
A \to & \mathtt{0}A\mathtt{1} & | \quad \mathtt{01} \\
B \to & \mathtt{1}B\mathtt{0} & | \quad \mathtt{10} \\
C \to & \mathtt{0} &
\end{array}
$$

Soluzione: Per determinare se la grammatica è deterministica eseguiamo il DK-test, ottenendo così il seguente diagramma:



Lo stato accettante raggiungibile dallo stato iniziale seguendo il simbolo ‘0’ contiene una regola completata e diverse regole in cui il punto precede un simbolo terminale. Perciò il DK-test è fallito, e di conseguenza la grammatica non è DCFG.

Esercizio 3 [7] Sia $A = \{u^{\mathcal{R}}\texttt{\#}v \mid u, v \in \{\texttt{0},\texttt{1}\}^*$, u è la codifica binaria di un numero $n \geq 1$, e v è la codifica binaria del numero $n+1\}$. Le codifiche binarie non hanno zeri non significativi a sinistra. Si noti che il linguaggio codifica u invertendo l'ordine dei suoi bit. Ad esempio fanno parte di A le stringhe `01#11` e `11#100`, mentre non fanno parte di A le stringhe `10#10` (perché $\texttt{10}^{\mathcal{R}} = \texttt{01}$ ha uno zero non significativo) e `1#11`. Dimostrare che A è un linguaggio libero dal contesto (CFL).

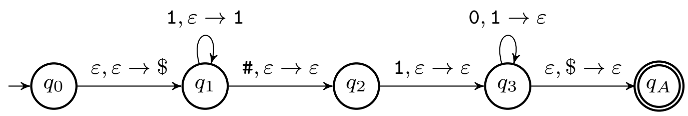
Soluzione:

Una semplice dimostrazione che A è CFL può essere ottenuta osservando che A è in effetti l'unione di due linguaggi:

$$A = B \cup C, B = \{\texttt{1}^n\texttt{\#10}^n \mid n \geq 1\}, C = \left\{ u^{\mathcal{R}}\texttt{\#}v \mid u, v \in \{\texttt{0},\texttt{1}\}^*, |u| = |v|, (u)_2 + 1 = (v)_2 \right\},$$

ove $(x)_2$ rappresenta il numero codificato in binario dalla stringa x. In altri termini, B rappresenta le istanze in cui è presente un riporto nella addizione $+1$, mentre C rappresenta le istanze di A in cui non è presente un riporto, e dunque tali che le stringhe u e v hanno la stessa lunghezza.

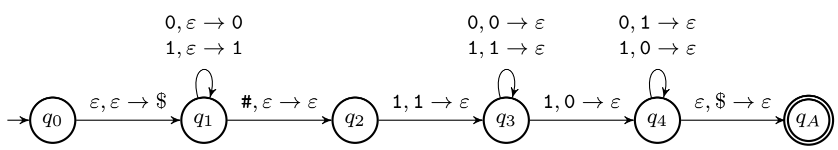
Per dimostrare che A è CFL è sufficiente dimostrare che sia B che C sono CFL (i linguaggi liberi dal contesto sono infatti chiusi rispetto all'operazione di unione). Il linguaggio B è una semplice variante del linguaggio $\texttt{a}^n\texttt{b}^n$, ed un PDA che lo riconosce è ad esempio:



Poiché le codifiche dei numeri nelle istanze-sì di A non devono contenenere zeri non significativi, il linguaggio C può essere descritto come:

$$C = \left\{ x\,\mathtt{0}\,w\,\mathtt{1}\,\mathtt{\#}\,\mathtt{1}\,y\,\mathtt{1}\,z \,|\, w, x, y, z \in \{\mathtt{0}, \mathtt{1}\}^*, w^{\mathcal{R}} = y, x^{\mathcal{R}} = \overline{z} \right\}$$

Un esempio di PDA che riconosce C è:



(Si osservi che la definizione di A richiede che il numero n codificato da u sia maggiore di zero. Se fosse ammesso anche il caso $n = 0$ allora bisognerebbe aggiungere a C la stringa $\mathtt{0\#1}$. Poiché un linguaggio costituito da una sola stringa è regolare, C continuerebbe comunque ad essere CFL.)

Esercizio 4 [6] Si consideri il linguaggio $A = \{x \in \{0, 1\}^* \mid x$ ha un numero di bit 1 uguale al numero di bit $0\}$. Sia $B = \{\langle M \rangle \mid M$ è una TM e $L(M) \subseteq A\}$. Il linguaggio B è decidibile oppure no? Giustificare la risposta con una dimostrazione.

Soluzione: Il linguaggio B non è decidibile, e per dimostrarlo è sufficiente verificare che le ipotesi del Teorema di Rice sono verificate. Si consideri come proprietà P del linguaggio della TM l'essere un sottinsieme del linguaggio A. Tale proprietà è non banale: infatti la TM che accetta tutte le stringhe non soddisfa la proprietà ($\Sigma^* \not\subseteq A$), mentre la TM che rifiuta tutte le stringhe soddisfa la proprietà ($\emptyset \subseteq A$). Inoltre, P è una proprietà del linguaggio riconosciuto dalla TM: infatti se due diverse TM riconoscono lo stesso linguaggio, per entrambe vale che il linguaggio è un sottinsieme di A, e dunque entrambe le TM soddisfano la priorità P. Poiché tutte le ipotesi del Teorema di Rice sono soddisfatte possiamo concludere immediatamente che il linguaggio B contenente codifiche di TM che soddisfano la proprietà P è indecidibile.

Esercizio 5 [7] Siano A e B linguaggi Turing-riconoscibili (ossia ricorsivamente enumerabili). La differenza simmetrica $A \triangle B$ di A e B (gli elementi che stanno in A o in B ma non in entrambi) è necessariamente Turing-riconoscibile? Giustificare la risposta con una dimostrazione.

Soluzione: $A \triangle B$ non è necessariamente Turing-riconoscibile; per dimostrarlo è sufficiente esibire un contro-esempio. Sia dunque $A = \mathcal{A}_{\mathrm{TM}}$, il linguaggio contenente le codifiche delle macchine di Turing e delle stringhe da esse accettate. Sia inoltre $B = \Sigma^*$, ove Σ è l'alfabeto sul quale sono costruite le istanze in $\mathcal{A}_{\mathrm{TM}}$. Si dimostra facilmente che $A \triangle B = (A \setminus B) \cup (B \setminus A)$; inoltre nel nostro caso $\mathcal{A}_{\mathrm{TM}} \setminus \Sigma^* = \emptyset$ e $\Sigma^* \setminus \mathcal{A}_{\mathrm{TM}} = \mathcal{A}_{\mathrm{TM}}^c$. Perciò $\mathcal{A}_{\mathrm{TM}} \triangle \Sigma^* = \emptyset \cup \mathcal{A}_{\mathrm{TM}}^c = \mathcal{A}_{\mathrm{TM}}^c$.

Sappiamo che Σ^* è regolare e quindi Turing-riconoscibile; anche $\mathcal{A}_{\mathrm{TM}}$ è Turing-riconoscibile, ma non decidibile. Perciò $\Sigma^* \triangle \mathcal{A}_{\mathrm{TM}} = \mathcal{A}^c_{\mathrm{TM}}$ non può essere Turing-riconoscibile; se infatti lo fosse, poiché sia $\mathcal{A}_{\mathrm{TM}}$ che $\mathcal{A}^c_{\mathrm{TM}}$ sarebbero Turing-riconoscibili, allora sarebbero anche entrambi decidibili, il che è manifestamente falso.

Esercizio 6 [8] Si consideri il problema FEEDBACK VERTEX SET: dato un grafo diretto $G = (V, A)$ ed un numero $k \in \mathbb{N}$, esiste un sottoinsieme $V' \subseteq V$ con $|V'| \leq k$ tale che ogni *ciclo* diretto entro G include almeno un nodo in V'? Dimostrare che il problema è NP-completo.

Soluzione: Il problema FEEDBACK VERTEX SET (o FVS) è verificabile polinomialmente: un certificato è banalmente la lista di nodi che costituisce l'insieme V'. Si consideri infatti il seguente algoritmo.

M= "On input $\langle G, k, V' \rangle$:
1. Verify that V' is a subset of k or less nodes of G
2. Build the graph $G' = G \setminus V'$
3. For every pairs of nodes s, t in G':
    4. Run PATH(G', s, t) to determine if there is a path from s to t
    5. If the path exists:
        6. Let I be the nodes of the path except s and t
        6. Build the graph $G'' = G \setminus I$
        7. Run PATH(G'', s, t)
            8. If the path exists, reject (cycle found)
9. There is no cycle in G', hence accept"

La notazione $G \setminus V$ indica il grafo ottenuto da G rimuovendo tutti i nodi dell'insieme V e tutti gli archi incidenti su questi nodi. Poiché l'algoritmo PATH è polinomiale, il verificatore esegue in tempo polinomiale nel numero di nodi del grafo G in istanza.

Per dimostrare che FVS è NP-hard possiamo considerare una semplicissima riduzione dal problema NP-completo VERTEX COVER. Sia infatti (G, k) una istanza del problema VC, ove G è un grafo non diretto e $k \in \mathbb{N}$. Consideriamo il grafo diretto G' ottenuto sostituendo ad ogni arco non diretto di G una coppia di archi in direzione opposta tra gli stessi nodi in G'. Sia dunque (G', k) l'istanza ridotta di FVS, che ha ovviamente dimensione polinomiale ed è meccanicamente costruibile.

Supponiamo che (G, k) sia una istanza-sì di VC. Dunque esiste un sottoinsieme di al più k nodi che copre ogni arco di VC. Lo stesso sottoinsieme di nodi in G' copre ogni arco diretto, quindi la rimozione dei nodi di V' rende G' un grafo senza archi, e quindi senza cicli. Perciò (G', k) è una istanza-sì di FVS. Viceversa, supponiamo che (G', k) sia una istanza-sì di FVS, pertanto esiste un sottoinsieme di al più k nodi che copre ogni ciclo del grafo G'. Si consideri ora che ciascun arco del grafo G ha dato origine ad un ciclo di dimensione 2 in G', e anche tali cicli devono essere coperti da V'. Pertanto l'insieme di nodi V' copre ogni arco del grafo G, e dunque (G, k) è una istanza-sì di VC.

Concludendo, VC $\leq_p$ FVS, pertanto FVS è NP-hard, e dunque NP-completo.