Tutorato 02 Soluzioni Esercizi

Giulio Umbrella

03 Aprile 2022

1 Espressioni regolari

1.1 Ex1

Scrivere l'espressione regolare per i seguenti linguaggi.

1. 1(0+1)*1
2. ((0+1)(0+1))*
3. ((0+1)(0+1))*(0+1)
4. (0+1)*00
5. (0+1)(0+1)(0+1)0(0+1)
6. (1 + 01 + 0011)*(0+e)
7. (0+1)*00 + 0
8. (0+1)(0+1)
9. 01 + 10 + 11

1.2 Ex2

Dire se le seguenti affermazioni sono vere o false.

1. Falso
2. Vero
3. Devo controllare l'ordine di applicazione delle parentesi, aggiornero' il prima possibile.

2 Conversione da RE a FA

Convertire le seguenti espressioni regolari in FA
Per la soluzione applicare quanto riportato nelle slide

3 Conversion da FA a RE

3.1 Ex1

Fornire un FA per i seguenti linguaggi e convertire l'automa in una espressione regolare.

3.1.1 Ex1

Stringhe binarie di lunghezza pari.

Per prima cosa, costruiamo l'automa che riconosce le stringhe binarie di lunghezza pari, Figura 1

Prima di procedere, dobbiamo verificare se dobbiamo modifiare l'automa. Le etichette che contengono la virgola $0, 1$ vengono modificate in somme $0 + 1$ come vediamo in 2. L'automa ha un solo stato accettante, quindi possiamo applicare l'algoritmo.

Abbiamo un solo stato da rimuovere q_1, quindi procediamo come indicato dall'algoritmo. Il percorso da eliminare e' $q_0 \to q_0$, quindi facciamo la concatenazione delle espressioni regolari lungo il percorso - senza includere self loop perche' asssenti in q_1. Notare che il punto di partenza e di arrivo conicidono ma la tecnica da applicare e' la stessa. L'espressione regolare ottenuta viene aggiunta come self loop su q_1, Figura 3

In Figura 3 vediamo che lo stato iniziale e finale coincidono, quindi possiamo ricavare l'espressione regolare come $((0 + 1)(0 + 1))^\star$.

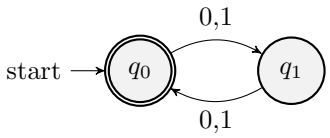

Figure 1: Automa di partenza

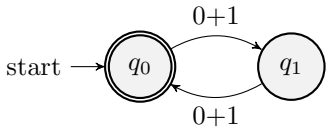

Figure 2: Sostituzione con espressoni regolari

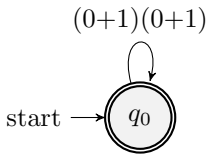

Figure 3: Rimozione stato q1

3.1.2 Ex2

Strighe binarie in cui il simbolo 1 e' ripetuto al piu' una volta sola

Anche in questo esercizio, il primo passaggio e' rappresentare l'automa che riconosce il linguaggio (Figura 4).

Rispetto all'esercizio precedente, l'automa ha bisogno di una modifica in piu'. Oltre a trasformare le etichette in espressioni regolari, dobbiamo ricavare un automa con un solo stato accettante. Per farlo e' sufficiente:

1. Aggiungere un nuovo stato accettante
2. Aggiungere una epsilon transizione da ciascuno stato accettante al nuovo stato accettante
3. Gli stati accettanti nell'automa di partenza diventano stati ordinari.

Il risultato e' in Figura 5

Ora possiamo procedere ed eliminiamo lo stato q_1. Osservando le frecce in ingresso ed in uscita, vediamo che lo stato ha un precedessore e due successori, quindi dobbiamo modificare due percorsi $q_0 \rightarrow q_3$ e $q_0 \rightarrow q_2$.

Per il percorso $q_0 \rightarrow q_2$ concateniamo le espressioni regolari lungo il percorso e inseriamo in mezzo il self-loop su q_1 e otteniamo l'espressione regolare $10^\star 1$. Notiamo che non ci sono percorsi diretti fra $_0$ e q_2, quindi non dobbiamo aggiunggere altri percorsi. Il risultato e' in 6.

Per il percorso $q_0 \rightarrow q_3$ dobbiamo fare piu' attenzione. Abbiamo un percorso rappresentato da $10^\star\epsilon$ e un percorso diretto rappresentato da ϵ, quindi sappiamo che dobbiamo unire le due espressioni regolare tramite il $+$. Qui e' bene fermarsi un attimo e ragionare su cosa stiamo facendo; mentre nella concatenazione il simbolo ϵ puo' essere rimosso, nell'unione dobbiamo convervarlo, come si vede in Figura 6.

Il motivo e' che nella concatenazione stiamo aggiungendo il simbolo vuoto all'espressione regolare $10^\star\epsilon$ e quindi lasciamo il risultato immutanto. Nell'unione

$10^\star\epsilon+\epsilon$ invece stiamo aggiungendo il simbolo vuoto all'espressione regolare, dandoli la possibilita' di accettare la parola vuota. In altri termini, scrivere $10^\star$ e' errato perche' stiamo lasciando fuori una parola, quella vuota.

La rimozione di q_1 da come risultato l'automa di Figura 6. Per prima cosa osserviamo che lo stato q_2 non ha alcuna transione verso q_3. Infatti, il percorso da q_0 a q_2 contiene tutte le stringhe non accettate dal linguaggio, ossia quelle in cui il simbolo 1 e' ripetuto piu' di una volta. Lo stato q_2 rappresenta una sorta di "cestino" o vicolo cieco; una volta entrati non e' piu' possibili arrivare in uno stato accettante. Per proseguire possiamo semplicemente rimuoverlo, ottenendo l'automa in Figura 7.

Abbiamo due stati, quindi terminiamo come suggerito dall'algoritmo e otteniamo l'espressione regolare $0^\star10^\star + \epsilon$.

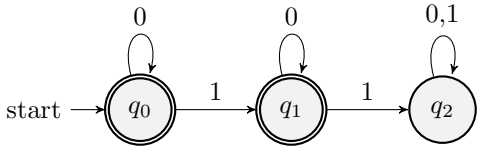

Figure 4: Automa di partenza Step 1

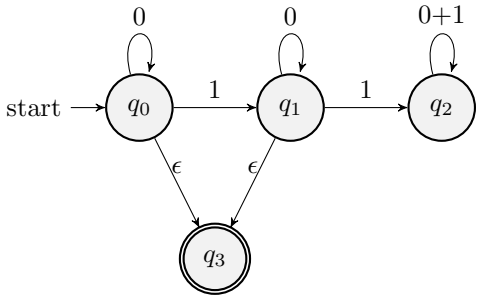

Figure 5: Aggiunta nuovo stato finale q3

3.1.3 Ex3

Stringhe binarie che non comprendono la stringa 101.

Questo linguaggio presenta una complicazione, infatti richiede di ottenere tutte le stringhe che non rispettano una condizione. Anziche' ragionare su questo automa, possiamo ragionare in due passaggi:

1. Costruiamo il DFA che accetta il linguaggio complementare.

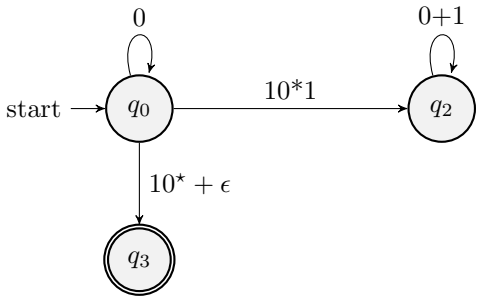

Figure 6: Rimozione stato q1

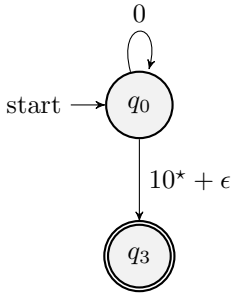

Figure 7: Rimozione stato q2

2. Invertiamo il DFA per ottenere il linguaggio di interesse.

L'operazione complemento infatti e' chiusa per la classe dei linguaggi regolari; se applichiamo il complemento ad un linguaggio regolare, otteniamo un linguaggio regolare.

Per poter ottenere l'inverso, scegliamo di costruire un DFA che accetta il linguaggio complementare (Figura 8). Infatti, sappiamo che DFA e NFA sono equivalenti, ma come avete visto in classe, possiamo ottenere il linguaggio complementare da un DFA in modo piu' agevole, invertendo gli stati (Figura 9).

Una volta ottenuto l'automa, come primo step lo modifichiamo per ottenere un automa su cui applicare l'algoritmo di conversione, il risultato e' in Figura 10. Anche per questo esercizio, abbiamo uno stato che porta a parole non accettate dal linguaggio, lo stato q_3. Possiamo quindi rimuovere questo stato e procedere come l'algoritmo.

3.2 Ex2

Convertire in RE il seguente automa (Figura) 11).

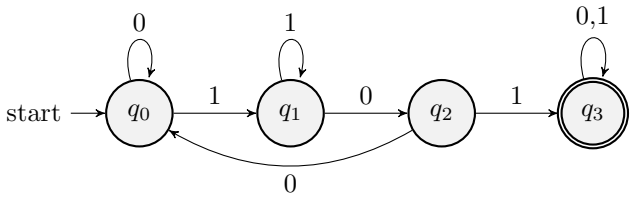

Figure 8: DFA linguaggio complementare

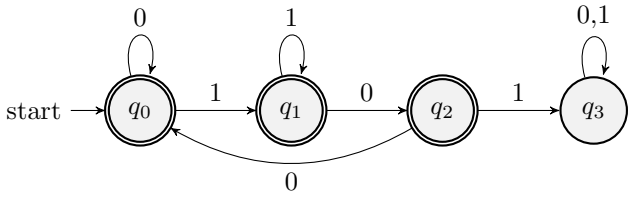

Figure 9: DFA inverso per lingugaggio di partenza

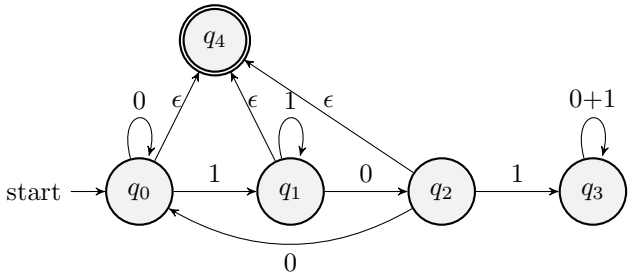

Figure 10: NFA

Per procedere rimuoviamo lo stato q_1 (Figura 12 e ricaviamo l'espressione regolare $1 + 01^{\star}0(0 + 1)^{\star}$

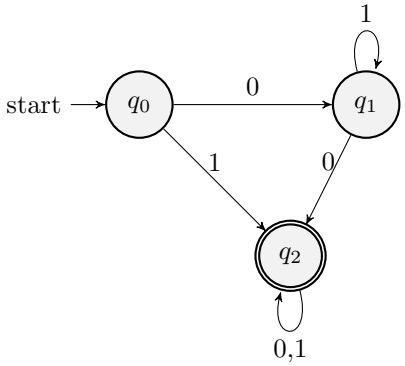

Figure 11: Automa di partenza

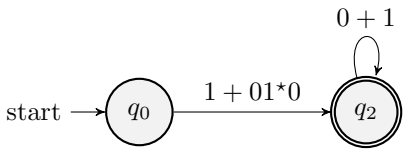

Figure 12: Rimozione stato q_1