Tutorato di Automi e Linguaggi Formali

Homework 2: Espressioni regolari, Equivalenze con automi, Conversioni

Gabriel Rovesti
Corso di Laurea in Informatica - Università degli Studi di Padova

Tutorato 2 - 17-03-2025

1 Espressioni Regolari e Operazioni

Esercizio 1. Per ciascuno dei seguenti linguaggi sull'alfabeto $\Sigma = \{a, b\}$, costruire un'espressione regolare che lo rappresenti:

a) $L_1 = \{w \in \{a, b\}^* \mid w \text{ contiene un numero pari di } a \text{ e un numero di } b \text{ multiplo di } 3\}$

b) $L_2 = \{w \in \{a, b\}^* \mid w \text{ contiene almeno una sottostringa } aba\}$

c) $L_3 = \{w \in \{a, b\}^* \mid w \text{ ha lunghezza almeno 2 e i primi due simboli sono uguali}\}$

Soluzione.

a) Per costruire questa espressione regolare, dobbiamo considerare che le stringhe devono soddisfare simultaneamente due condizioni: un numero pari di a e un numero di b multiplo di 3.

   Osserviamo che questo linguaggio può essere riconosciuto da un automa a stati finiti con 6 stati (2 stati per contare la parità di a × 3 stati per contare i b modulo 3). Per semplificare l'espressione, notiamo che:

   - Una stringa con numero pari di a può essere descritta come $(b^*ab^*a)^*$
   - Una stringa con numero di b multiplo di 3 può essere descritta come $(a^*ba^*ba^*b)^*a^*$

   Considerando l'intersezione di questi due linguaggi, otteniamo:

   $L_1 = ((b^3)^* + b(b^3)^*b^2 + b^2(b^3)^*b)(aa)^*$

   Una forma più compatta equivalente è:

   $L_1 = (b^3)^*(aa)^* + (b(b^3)^*b^2)(aa)^* + (b^2(b^3)^*b)(aa)^*$

b) L'espressione regolare per questo linguaggio è semplice, poiché richiede solo la presenza della sottostringa aba:

$L_2 = (a + b)^*aba(a + b)^*$

c) Per questo linguaggio, dobbiamo considerare stringhe di lunghezza almeno 2 che iniziano con due simboli uguali:

$L_3 = aa(a + b)^* + bb(a + b)^*$

Esercizio 2. Scrivere un'espressione regolare per ciascuno dei seguenti linguaggi sull'alfabeto $\Sigma = \{0, 1\}$:

a) $L_1 = \{w \in \{0, 1\}^* \mid w$ termina con 01 e ha lunghezza almeno 3$\}$

b) $L_2 = \{w \in \{0, 1\}^* \mid w$ contiene esattamente tre occorrenze del simbolo 1$\}$

c) $L_3 = \{w \in \{0, 1\}^* \mid w$ non contiene due 1 consecutivi$\}$

Soluzione.

a) Per rappresentare stringhe che terminano con 01 e hanno lunghezza almeno 3, dobbiamo garantire che ci sia almeno un simbolo prima del suffisso 01:

$L_1 = (0 + 1)(0 + 1)^*01$

b) Per stringhe con esattamente tre occorrenze del simbolo 1, possiamo usare:

$L_2 = 0^*10^*10^*10^*$

Quest'espressione descrive stringhe con zero o più 0, seguiti da un 1, seguiti da zero o più 0, seguiti da un 1, seguiti da zero o più 0, seguiti da un 1, seguiti da zero o più 0.

c) Per stringhe che non contengono due 1 consecutivi:

$L_3 = 0^*(10^*)*$

In modo equivalente, possiamo scrivere:

$L_3 = (0 + 10)^*(\varepsilon + 1)$

Quest'espressione descrive stringhe che consistono in sequenze di 0 o 10, possibilmente seguite da un 1 singolo alla fine.

Esercizio 3. Date le seguenti espressioni regolari sull'alfabeto $\Sigma = \{a, b, c\}$, descrivere in linguaggio naturale il linguaggio che rappresentano:

a) $(a + b)^*c(a + b + c)^*$

b) $a^*b(a + b)^*b^*$

c) $(ab + bc + ac)^*(a + b + c + \varepsilon)$

Soluzione.

a) $(a+b)^*c(a+b+c)^*$ rappresenta il linguaggio di tutte le stringhe sull'alfabeto $\{a, b, c\}$ che contengono almeno una occorrenza del simbolo c.

Analisi: l'espressione inizia con $(a+b)^*$ che genera qualsiasi sequenza (anche vuota) di a e b, seguita da una c obbligatoria, seguita da $(a+b+c)^*$ che genera qualsiasi sequenza (anche vuota) di a, b e c. Quindi la presenza di almeno una c è garantita.

b) $a^*b(a+b)^*b^*$ rappresenta il linguaggio di tutte le stringhe sull'alfabeto $\{a, b\}$ che contengono almeno una occorrenza del simbolo b e, se iniziano con una sequenza di a, la prima occorrenza di un simbolo diverso da a deve essere una b.

Analisi: l'espressione inizia con a^* (zero o più a), seguita da una b obbligatoria, seguita da $(a+b)^*$ (qualsiasi sequenza di a e b), seguita da b^* (zero o più b).

c) $(ab+bc+ac)^*(a+b+c+\varepsilon)$ rappresenta il linguaggio di tutte le stringhe sull'alfabeto $\{a, b, c\}$ che possono essere formate concatenando zero o più delle sottostringhe ab, bc, o ac, possibilmente seguite da un singolo simbolo a, b, c o nessun simbolo aggiuntivo.

Analisi: l'espressione inizia con $(ab+bc+ac)^*$ che genera qualsiasi sequenza (anche vuota) delle sottostringhe ab, bc, o ac, seguita opzionalmente da uno tra a, b, c o nulla (ε).

2 Conversione da Espressioni Regolari a NFA

Esercizio 4. Convertire le seguenti espressioni regolari in NFA utilizzando le costruzioni viste a lezione:

a) $(ab)^* + (ba)^*$

b) $a(a+b)^*b$

c) $(a+\varepsilon)(b+c)^*$

Per ogni NFA ottenuto, fornire:

i) Il diagramma degli stati

ii) La tabella di transizione completa

Soluzione. Utilizzerò la costruzione di Thompson per convertire le espressioni regolari in NFA.

a) $(ab)^* + (ba)^*$

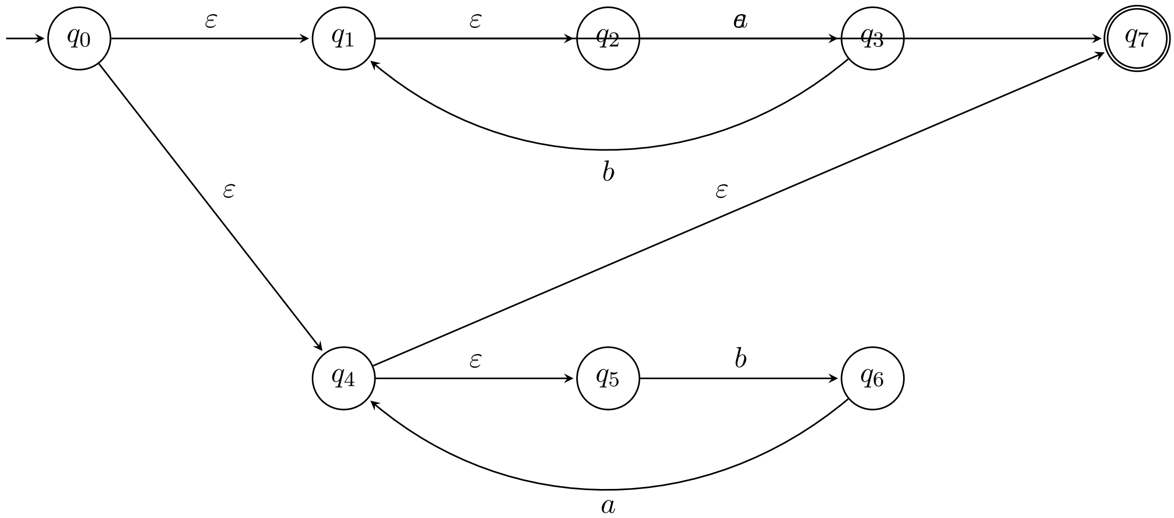


Tabella di transizione:

Stato	a	b	ε
q_0	-	-	$\{q_1, q_4\}$
q_1	-	-	$\{q_2, q_7\}$
q_2	$\{q_3\}$	-	-
q_3	-	$\{q_1\}$	-
q_4	-	-	$\{q_5, q_7\}$
q_5	-	$\{q_6\}$	-
q_6	$\{q_4\}$	-	-
q_7	-	-	-

b) $a(a+b)^*b$

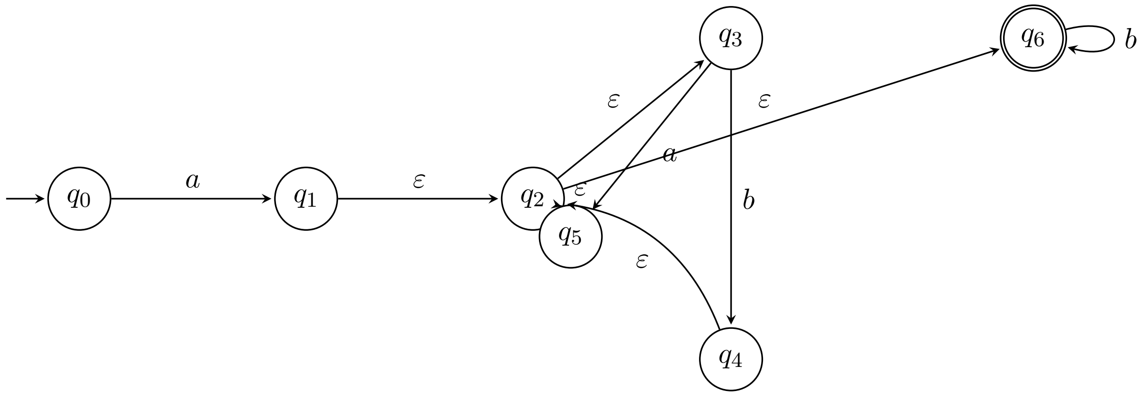


Tabella di transizione:

Stato	a	b	ε
q_0	$\{q_1\}$	-	-
q_1	-	-	$\{q_2\}$
q_2	-	-	$\{q_3, q_6\}$
q_3	$\{q_5\}$	$\{q_4\}$	-
q_4	-	-	$\{q_2\}$
q_5	-	-	$\{q_2\}$
q_6	-	$\{q_6\}$	-

c) $(a + \varepsilon)(b + c)^*$

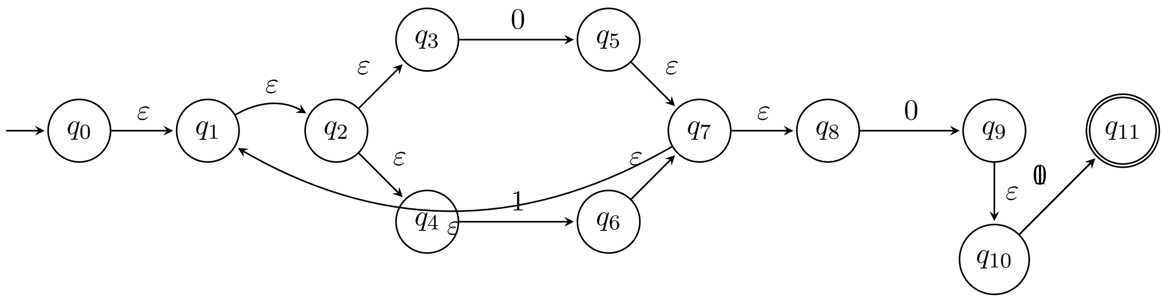


Tabella di transizione:

Stato	a	b	c	ε
q_0	-	-	-	$\{q_1\}$
q_1	-	-	-	$\{q_2, q_3\}$
q_2	$\{q_4\}$	-	-	-
q_3	-	-	-	$\{q_4\}$
q_4	-	-	-	$\{q_5\}$
q_5	-	-	-	$\{q_6\}$
q_6	-	$\{q_7\}$	$\{q_7\}$	-
q_7	-	-	-	$\{q_5\}$

Esercizio 5. Considerare l'espressione regolare $(0 + 1)^*0(0 + 1)$ sull'alfabeto $\Sigma = \{0, 1\}$.

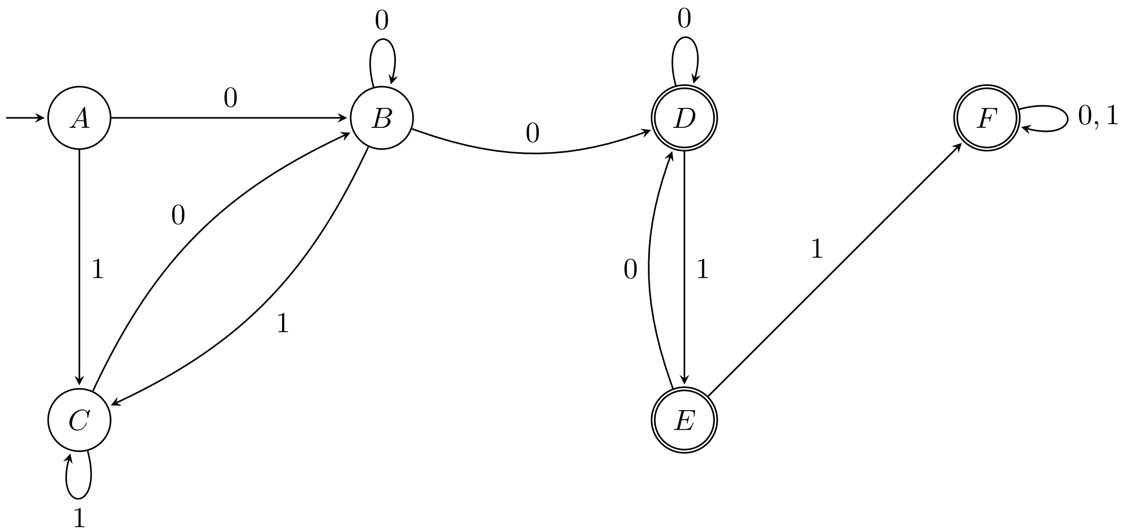
a) Costruire un ε-NFA che riconosce il linguaggio generato da questa espressione.

b) Convertire l'ε-NFA ottenuto in un NFA senza ε-transizioni.

c) Convertire il NFA in un DFA utilizzando la costruzione per sottoinsiemi.

Soluzione.

a) Costruiamo un ε-NFA per l'espressione regolare $(0 + 1)^*0(0 + 1)$:

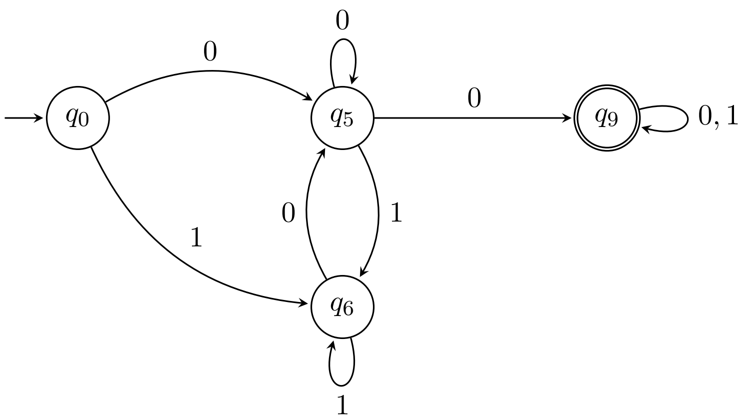


b) Convertire l'ε-NFA in un NFA senza ε-transizioni:

Prima calcoliamo la chiusura-ε per ogni stato:

ECLOSE(q_0) = $\{q_0, q_1, q_2, q_3, q_4\}$
ECLOSE(q_1) = $\{q_1, q_2, q_3, q_4\}$
ECLOSE(q_2) = $\{q_2, q_3, q_4\}$
ECLOSE(q_3) = $\{q_3\}$
ECLOSE(q_4) = $\{q_4\}$
ECLOSE(q_5) = $\{q_5, q_7, q_1, q_2, q_3, q_4, q_8\}$
ECLOSE(q_6) = $\{q_6, q_7, q_1, q_2, q_3, q_4, q_8\}$
ECLOSE(q_7) = $\{q_7, q_1, q_2, q_3, q_4, q_8\}$
ECLOSE(q_8) = $\{q_8\}$
ECLOSE(q_9) = $\{q_9, q_{10}, q_{11}\}$
ECLOSE(q_{10}) = $\{q_{10}, q_{11}\}$
ECLOSE(q_{11}) = $\{q_{11}\}$

Il nuovo NFA avrà le seguenti transizioni:



c) Convertire il NFA in un DFA utilizzando la costruzione per sottoinsiemi:

Stati del DFA: $A = \{q_0\}$ $B = \{q_5\}$ $C = \{q_6\}$ $D = \{q_5, q_9\}$ $E = \{q_6, q_9\}$ $F = \{q_9\}$

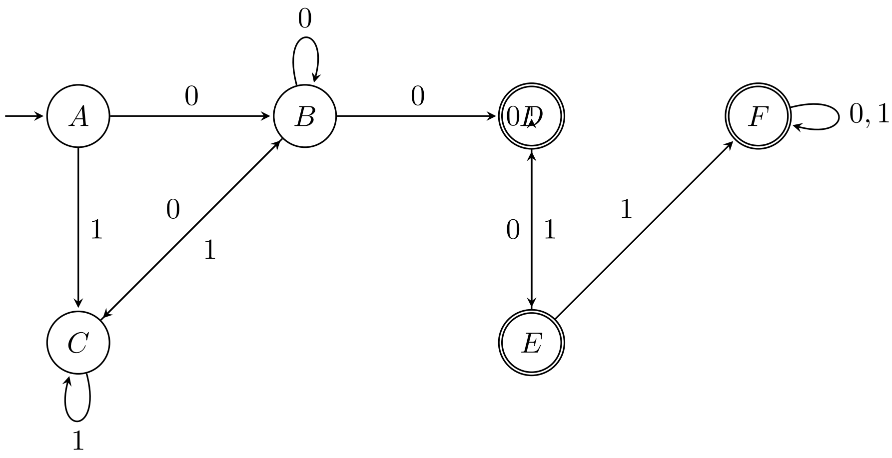
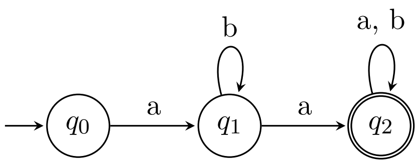


Tabella di transizione del DFA:

Stato	0	1
A	B	C
B	D	C
C	B	C
D	D	E
E	D	F
F	F	F

3 Conversione da NFA/DFA a Espressioni Regolari

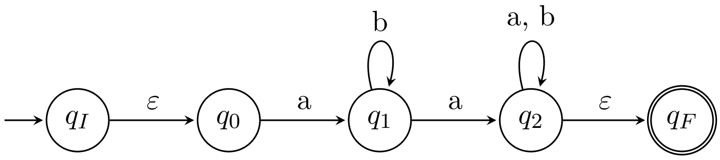
Esercizio 6. Convertire il seguente NFA in un'espressione regolare equivalente utilizzando l'algoritmo di eliminazione degli stati:



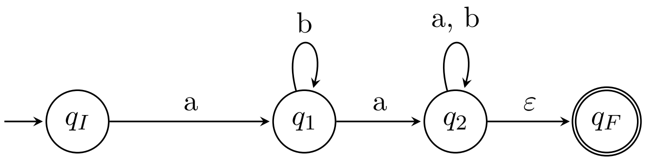
Mostrare tutti i passaggi dell'algoritmo di eliminazione degli stati e l'espressione regolare finale.

Soluzione. Applichiamo l'algoritmo di eliminazione degli stati per convertire l'NFA dato in un'espressione regolare equivalente.

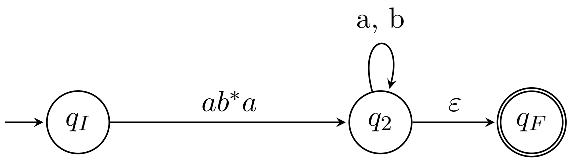
Passo 1: Aggiungere un nuovo stato iniziale q_I e un nuovo stato finale q_F:



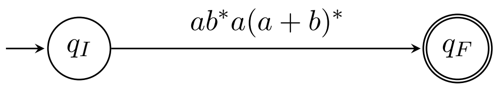
Passo 2: Eliminare lo stato q_0:



Passo 3: Eliminare lo stato q_1:



Passo 4: Eliminare lo stato q_2:



L'espressione regolare finale è quindi: $ab^*a(a+b)^*$