Soluzioni - Tutorato di Automi e Linguaggi Formali

Homework 1: DFA, NFA ed ϵ–NFA, conversioni ed operazioni su linguaggi

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

Tutorato 1 - 10-03-2025

1 Progettazione di DFA

Esercizio 1. Progettare un DFA sull'alfabeto $\Sigma = \{0, 1\}$ che riconosca il linguaggio

$$L_1 = \{w \in \{0, 1\}^* \mid w \text{ termina con } 10\}.$$

a) Disegnare il diagramma degli stati.

b) Fornire la tabella di transizione completa.

Soluzione. Per riconoscere le stringhe che terminano con 10, abbiamo bisogno di tracciare gli ultimi due simboli letti. Possiamo usare 3 stati:

- q_0: lo stato iniziale, non abbiamo ancora letto nulla o l'ultimo simbolo non è rilevante
- q_1: abbiamo appena letto un 1
- q_2: abbiamo letto 10 (stato finale)

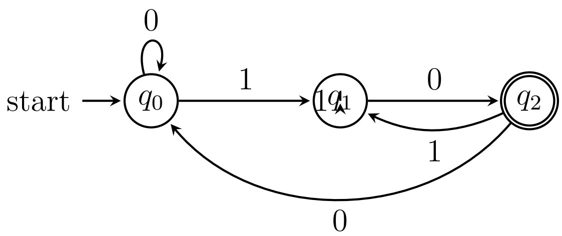

Figure 1: DFA per il linguaggio $L_1 = \{w \in \{0, 1\}^* \mid w \text{ termina con } 10\}$

Tabella di transizione:

	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_2	q_1
$*q_2$	q_0	q_1

Esercizio 2. Progettare un DFA sull'alfabeto $\Sigma = \{0, 1\}$ che riconosca il linguaggio

$$L_2 = \{w \in \{0,1\}^* \mid w \text{ contiene un numero pari di 1 e un numero dispari di 0}\}.$$

a) Disegnare il diagramma degli stati.

b) Fornire la tabella di transizione completa.

Soluzione. Dobbiamo tenere traccia contemporaneamente sia della parità del numero di 1 che di 0. Questo porta a 4 stati:

- q_0: numero pari di 1 e numero pari di 0 (stato iniziale)
- q_1: numero pari di 1 e numero dispari di 0 (stato finale)
- q_2: numero dispari di 1 e numero pari di 0
- q_3: numero dispari di 1 e numero dispari di 0

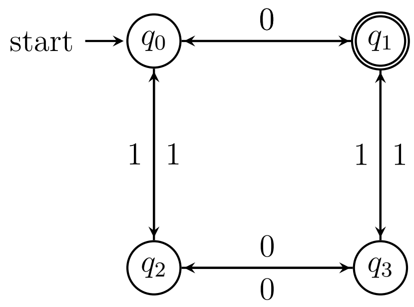

Figure 2: DFA per il linguaggio $L_2 = \{w \in \{0,1\}^* \mid w \text{ contiene un numero pari di 1 e un numero dispari di 0}\}$

Tabella di transizione:

	0	1
$\rightarrow q_0$	q_1	q_2
$*q_1$	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1

Esercizio 3. Progettare un DFA sull'alfabeto $\Sigma = \{0, 1\}$ che riconosca il linguaggio

$$L_3 = \{w \in \{0,1\}^* \mid \text{ogni occorrenza di 0 è immediatamente seguita da almeno due 1}\}.$$

a) Disegnare il diagramma degli stati.

b) Fornire la tabella di transizione completa.

Soluzione. Per questo linguaggio, dobbiamo tenere traccia di quanti 1 consecutivi abbiamo letto dopo un 0. Possiamo utilizzare 4 stati:

- q_0: stato iniziale e accettante, non abbiamo letto 0 oppure ogni 0 letto è stato seguito da almeno due 1
- q_1: abbiamo letto un 0 e dobbiamo leggere almeno due 1
- q_2: abbiamo letto un 0 seguito da un solo 1, quindi dobbiamo leggere almeno un altro 1
- q_3: stato pozzo, la stringa non può più essere accettata (abbiamo violato la condizione)

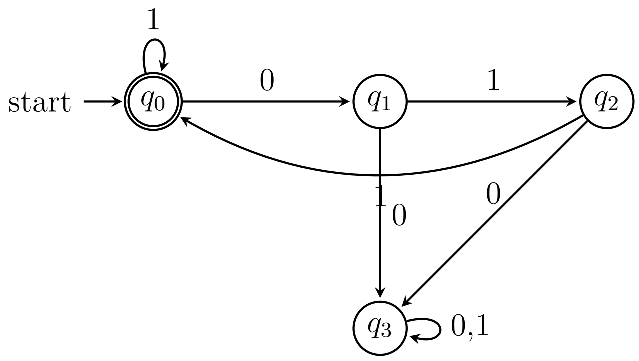

Figure 3: DFA per il linguaggio $L_3 = \{w \in \{0,1\}^* \mid \text{ogni occorrenza di 0 è immediatamente seguita da almeno due 1}\}$

Tabella di transizione:

	0	1
$\rightarrow *q_0$	q_1	q_0
q_1	q_3	q_2
q_2	q_3	q_0
q_3	q_3	q_3

2 Progettazione di NFA

Esercizio 4. Progettare un NFA sull'alfabeto $\Sigma = \{a, b\}$ che riconosca il linguaggio

$$L_4 = \{w \in \{a,b\}^* \mid w \text{ contiene la sottostringa } aba\}.$$

a) Disegnare il diagramma degli stati.

b) Fornire la tabella di transizione completa.

Soluzione. Per riconoscere le stringhe che contengono la sottostringa *aba*, possiamo utilizzare un NFA con 4 stati che traccia il progresso nella lettura di questa sottostringa:

- q_0: lo stato iniziale
- q_1: abbiamo letto una 'a'
- q_2: abbiamo letto 'ab'
- q_3: abbiamo letto 'aba' (stato finale)

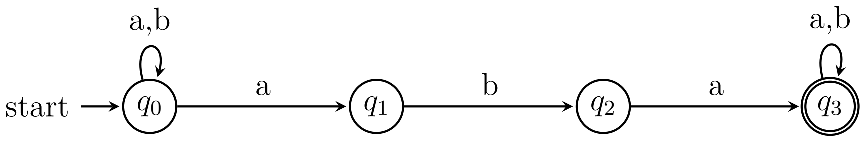

Figure 4: NFA per il linguaggio $L_4 = \{w \in \{a, b\}^* \mid w \text{ contiene la sottostringa } aba\}$

Tabella di transizione:

	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
q_2	$\{q_3\}$	$\emptyset$
$*q_3$	$\{q_3\}$	$\{q_3\}$

Esercizio 5. Progettare un NFA sull'alfabeto $\Sigma = \{0, 1\}$ che riconosca il linguaggio

$$L_5 = \{w \in \{0, 1\}^* \mid w \text{ inizia con 1 oppure termina con 0}\}.$$

a) Disegnare il diagramma degli stati.

b) Fornire la tabella di transizione completa.

Soluzione. Questo è un caso ideale per sfruttare il non determinismo. Possiamo costruire un NFA con due "componenti" separate: una che riconosce le stringhe che iniziano con 1 e un'altra che riconosce le stringhe che terminano con 0.

Tabella di transizione:

	0	1
$\rightarrow q_0$	$\{q_2\}$	$\{q_1, q_2\}$
$*q_1$	$\{q_1\}$	$\{q_1\}$
q_2	$\{q_2, q_3\}$	$\{q_2\}$
$*q_3$	$\{q_2, q_3\}$	$\{q_2\}$

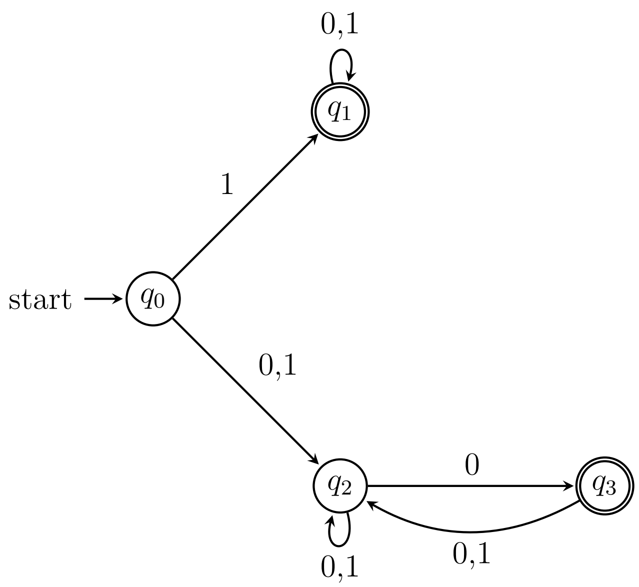

Figure 5: NFA per il linguaggio $L_5 = \{w \in \{0,1\}^* \mid w \text{ inizia con 1 oppure termina con 0}\}$

3 Conversione da NFA a DFA

Esercizio 6. Si consideri il seguente NFA N sull'alfabeto $\Sigma = \{a, b\}$:

	a	b
$\to q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
$*q_2$	$\{q_2\}$	$\{q_1\}$

a) Applicare la costruzione per sottoinsiemi per ottenere il DFA equivalente D.

b) Disegnare il diagramma degli stati del DFA ottenuto.

c) Determinare il linguaggio riconosciuto dall'automa.

Soluzione. Applichiamo la costruzione per sottoinsiemi:

- Stato iniziale del DFA: $\{q_0\}$
- Per $\{q_0\}$:
  - $\delta_D(\{q_0\}, a) = \delta_N(q_0, a) = \{q_0, q_1\}$
  - $\delta_D(\{q_0\}, b) = \delta_N(q_0, b) = \{q_0\}$
- Per $\{q_0, q_1\}$:
  - $\delta_D(\{q_0, q_1\}, a) = \delta_N(q_0, a) \cup \delta_N(q_1, a) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
  - $\delta_D(\{q_0, q_1\}, b) = \delta_N(q_0, b) \cup \delta_N(q_1, b) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$
- Per $\{q_0, q_2\}$:

- $\delta_D(\{q_0, q_2\}, a) = \delta_N(q_0, a) \cup \delta_N(q_2, a) = \{q_0, q_1\} \cup \{q_2\} = \{q_0, q_1, q_2\}$
- $\delta_D(\{q_0, q_2\}, b) = \delta_N(q_0, b) \cup \delta_N(q_2, b) = \{q_0\} \cup \{q_1\} = \{q_0, q_1\}$

- Per $\{q_0, q_1, q_2\}$:
  - $\delta_D(\{q_0, q_1, q_2\}, a) = \delta_N(q_0, a) \cup \delta_N(q_1, a) \cup \delta_N(q_2, a) = \{q_0, q_1\} \cup \emptyset \cup \{q_2\} = \{q_0, q_1, q_2\}$
  - $\delta_D(\{q_0, q_1, q_2\}, b) = \delta_N(q_0, b) \cup \delta_N(q_1, b) \cup \delta_N(q_2, b) = \{q_0\} \cup \{q_2\} \cup \{q_1\} = \{q_0, q_1, q_2\}$

Il DFA risultante ha i seguenti stati:

- $\{q_0\}$ - stato iniziale
- $\{q_0, q_1\}$
- $\{q_0, q_2\}$ - stato accettante (contiene q_2)
- $\{q_0, q_1, q_2\}$ - stato accettante (contiene q_2)

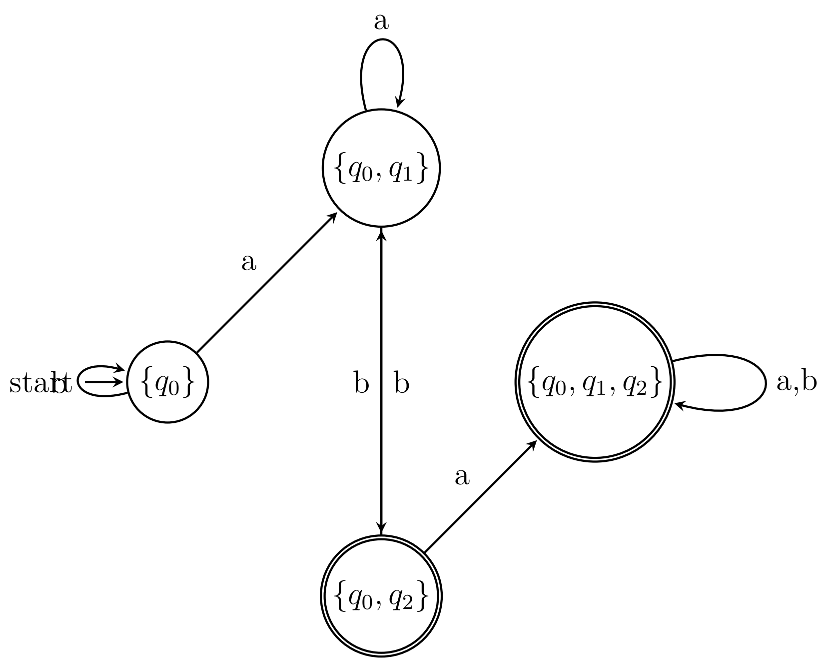

Figure 6: DFA equivalente ottenuto mediante costruzione per sottoinsiemi

Tabella di transizione del DFA:

	a	b
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$*\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

Il linguaggio riconosciuto dall'automa è l'insieme delle stringhe che contengono la sottostringa 'ab'. Infatti, analizzando l'NFA originale, vediamo che:

- Lo stato q_0 ha un self-loop su se stesso con 'a' e 'b'
- Da q_0 possiamo passare a q_1 leggendo 'a'
- Da q_1 possiamo passare a q_2 (finale) leggendo 'b'
- Una volta in q_2, rimaniamo in uno stato accettante

Quindi, $L(N) = \{w \in \{a,b\}^* \mid w \text{ contiene la sottostringa } ab\}$.

4 ε-NFA e ε-chiusure

Esercizio 7. Dato il seguente ϵ-NFA sull'alfabeto $\Sigma = \{a, b\}$:

	a	b	ϵ
$\rightarrow q_0$	$\emptyset$	$\emptyset$	$\{q_1, q_3\}$
q_1	$\{q_2\}$	$\emptyset$	$\emptyset$
q_2	$\emptyset$	$\{q_4\}$	$\{q_3\}$
q_3	$\{q_5\}$	$\emptyset$	$\emptyset$
q_4	$\emptyset$	$\{q_5\}$	$\emptyset$
$*q_5$	$\emptyset$	$\emptyset$	$\emptyset$

a) Calcolare l'ϵ-chiusura di ciascuno stato: ECLOSE(q_0), ECLOSE(q_1), ECLOSE(q_2), ECLOSE(q_3), ECLOSE(q_4), ECLOSE(q_5).

b) Convertire l'ϵ-NFA in un NFA equivalente senza ϵ-transizioni, fornendo la tabella di transizione completa.

Soluzione.

a) Calcoliamo le ϵ-chiusure:

   - ECLOSE(q_0) $= \{q_0, q_1, q_3\}$ (da q_0 possiamo raggiungere q_1 e q_3 tramite ϵ-transizioni)
   - ECLOSE(q_1) $= \{q_1\}$ (non ci sono ϵ-transizioni da q_1)
   - ECLOSE(q_2) $= \{q_2, q_3\}$ (da q_2 possiamo raggiungere q_3 tramite ϵ-transizione)
   - ECLOSE(q_3) $= \{q_3\}$ (non ci sono ϵ-transizioni da q_3)
   - ECLOSE(q_4) $= \{q_4\}$ (non ci sono ϵ-transizioni da q_4)
   - ECLOSE(q_5) $= \{q_5\}$ (non ci sono ϵ-transizioni da q_5)

b) Per convertire l'ϵ-NFA in un NFA senza ϵ-transizioni, utilizziamo la seguente regola: $\delta'(q, a) = \bigcup_{p \in \text{ECLOSE}(q)} \text{ECLOSE}(\delta(p, a))$, dove δ' è la funzione di transizione del nuovo NFA e δ è quella dell'ϵ-NFA originale.

   Tabella di transizione del nuovo NFA:

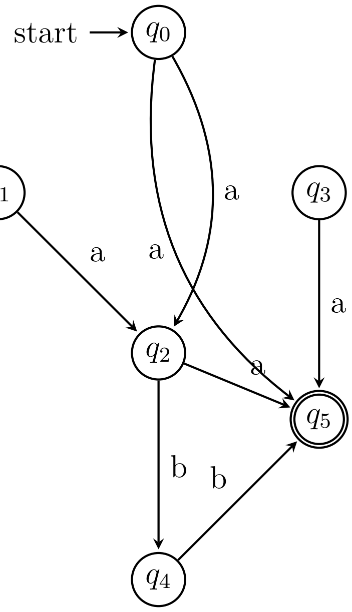

Figure 7: NFA senza ϵ-transizioni equivalente all'ϵ-NFA originale

	a	b
$\rightarrow q_0$	$\{q_2, q_3, q_5\}$	$\emptyset$
q_1	$\{q_2, q_3\}$	$\emptyset$
q_2	$\{q_5\}$	$\{q_4\}$
q_3	$\{q_5\}$	$\emptyset$
q_4	$\emptyset$	$\{q_5\}$
$*q_5$	$\emptyset$	$\emptyset$

5 Operazioni su Linguaggi e Automi

Esercizio 8. Siano $L_1 = \{w \in \{0,1\}^* \mid w \text{ inizia con } 0\}$ e $L_2 = \{w \in \{0,1\}^* \mid w \text{ termina con } 1\}$.

a) Progettare un DFA che riconosca L_1.

b) Progettare un DFA che riconosca L_2.

c) Utilizzando le operazioni sui linguaggi regolari, costruire un NFA che riconosca il linguaggio $L_1 \cup L_2$.

d) Convertire il NFA ottenuto in un DFA equivalente mediante la costruzione per sottoinsiemi.

Soluzione.

a) DFA per $L_1 = \{w \in \{0,1\}^* \mid w \text{ inizia con } 0\}$:

Tabella di transizione:

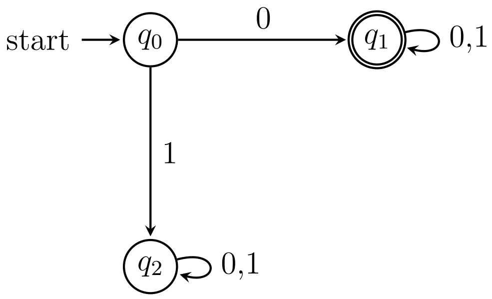

Figure 8: DFA per il linguaggio $L_1 = \{w \in \{0,1\}^* \mid w \text{ inizia con } 0\}$

	0	1
$\rightarrow q_0$	q_1	q_2
$*q_1$	q_1	q_1
q_2	q_2	q_2

b) DFA per $L_2 = \{w \in \{0,1\}^* \mid w \text{ termina con } 1\}$:

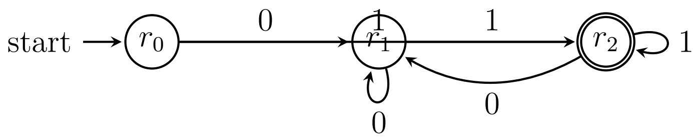

Figure 9: DFA per il linguaggio $L_2 = \{w \in \{0,1\}^* \mid w \text{ termina con } 1\}$

Tabella di transizione:

	0	1
$\rightarrow r_0$	r_1	r_2
r_1	r_1	r_2
$*r_2$	r_1	r_2

c) NFA per $L_1 \cup L_2$:

Per costruire un NFA che riconosca l'unione di due linguaggi regolari, creiamo un nuovo stato iniziale che ha ϵ-transizioni verso gli stati iniziali dei due automi originali.

Tabella di transizione del NFA:

	0	1	ϵ
$\rightarrow s_0$	$\emptyset$	$\emptyset$	$\{q_0, r_0\}$
q_0	$\{q_1\}$	$\{q_2\}$	$\emptyset$
$*q_1$	$\{q_1\}$	$\{q_1\}$	$\emptyset$
q_2	$\{q_2\}$	$\{q_2\}$	$\emptyset$
r_0	$\{r_1\}$	$\{r_2\}$	$\emptyset$
r_1	$\{r_1\}$	$\{r_2\}$	$\emptyset$
$*r_2$	$\{r_1\}$	$\{r_2\}$	$\emptyset$

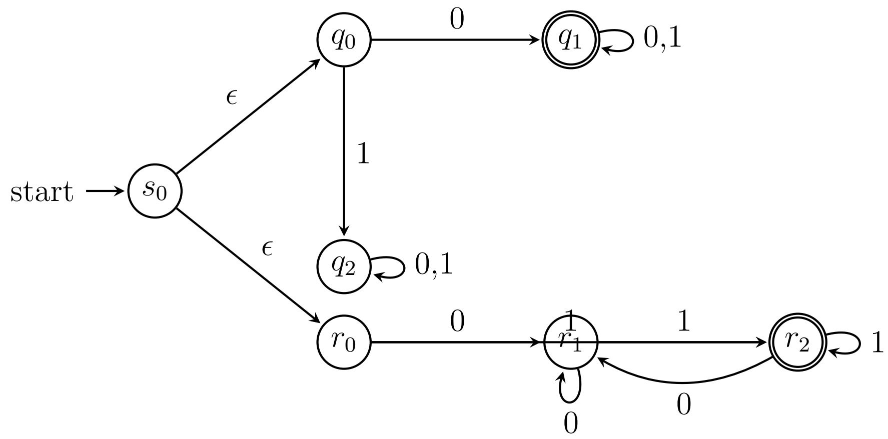

Figure 10: NFA per il linguaggio $L_1 \cup L_2$

d) Conversione del NFA ottenuto in un DFA mediante la costruzione per sottoinsiemi:

Prima calcoliamo le ϵ-chiusure:

- ECLOSE(s_0) = $\{s_0, q_0, r_0\}$
- ECLOSE(q_0) = $\{q_0\}$
- ECLOSE(q_1) = $\{q_1\}$
- ECLOSE(q_2) = $\{q_2\}$
- ECLOSE(r_0) = $\{r_0\}$
- ECLOSE(r_1) = $\{r_1\}$
- ECLOSE(r_2) = $\{r_2\}$

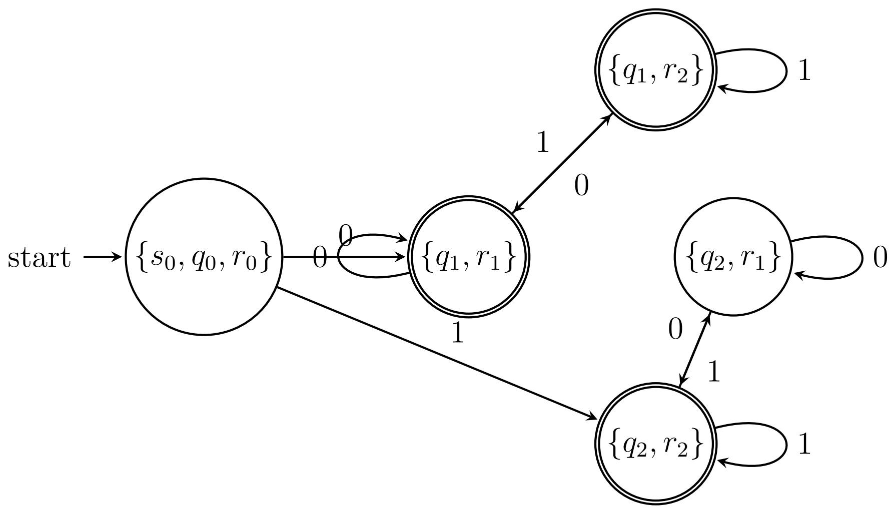

Figure 11: DFA ottenuto dal NFA per $L_1 \cup L_2$ mediante la costruzione per sottoinsiemi

Tabella di transizione del DFA:

	0	1
$\to \{s_0, q_0, r_0\}$	$\{q_1, r_1\}$	$\{q_2, r_2\}$
$*\{q_1, r_1\}$	$\{q_1, r_1\}$	$\{q_1, r_2\}$
$*\{q_1, r_2\}$	$\{q_1, r_1\}$	$\{q_1, r_2\}$
$*\{q_2, r_2\}$	$\{q_2, r_1\}$	$\{q_2, r_2\}$
$\{q_2, r_1\}$	$\{q_2, r_1\}$	$\{q_2, r_2\}$

Esercizio 9. Siano $L_1 = \{w \in \{a,b\}^* \mid \text{ogni } a \text{ è seguita da almeno una } b\}$ e $L_2 = \{w \in \{a,b\}^* \mid w \text{ contiene la sottostringa } ab\}$.

a) Progettare un DFA per L_1 e un NFA per L_2.

b) Costruire un NFA che riconosca $L_1 \cap L_2$.

c) Qual è l'interpretazione di questo linguaggio in linguaggio naturale?

Soluzione.

a) DFA per $L_1 = \{w \in \{a,b\}^* \mid \text{ogni } a \text{ è seguita da almeno una } b\}$:

Il linguaggio L_1 contiene stringhe in cui ogni 'a' è seguita da almeno una 'b'. Possiamo usare 2 stati:

- q_0: stato iniziale e accettante, tutte le 'a' viste finora sono state seguite da almeno una 'b' (anche se non abbiamo visto nessuna 'a')
- q_1: abbiamo visto una 'a' che non è ancora stata seguita da una 'b'

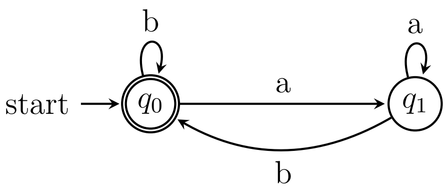

Figure 12: DFA per il linguaggio $L_1 = \{w \in \{a,b\}^* \mid \text{ogni } a \text{ è seguita da almeno una } b\}$

Tabella di transizione:

	a	b
$\to *q_0$	q_1	q_0
q_1	q_1	q_0

NFA per $L_2 = \{w \in \{a,b\}^* \mid w \text{ contiene la sottostringa } ab\}$:

Tabella di transizione:

	a	b
$\to r_0$	$\{r_0, r_1\}$	$\{r_0\}$
r_1	$\emptyset$	$\{r_2\}$
$*r_2$	$\{r_2\}$	$\{r_2\}$

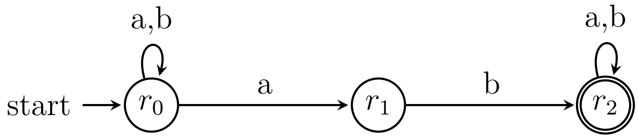

Figure 13: NFA per il linguaggio $L_2 = \{w \in \{a,b\}^* \mid w \text{ contiene la sottostringa } ab\}$

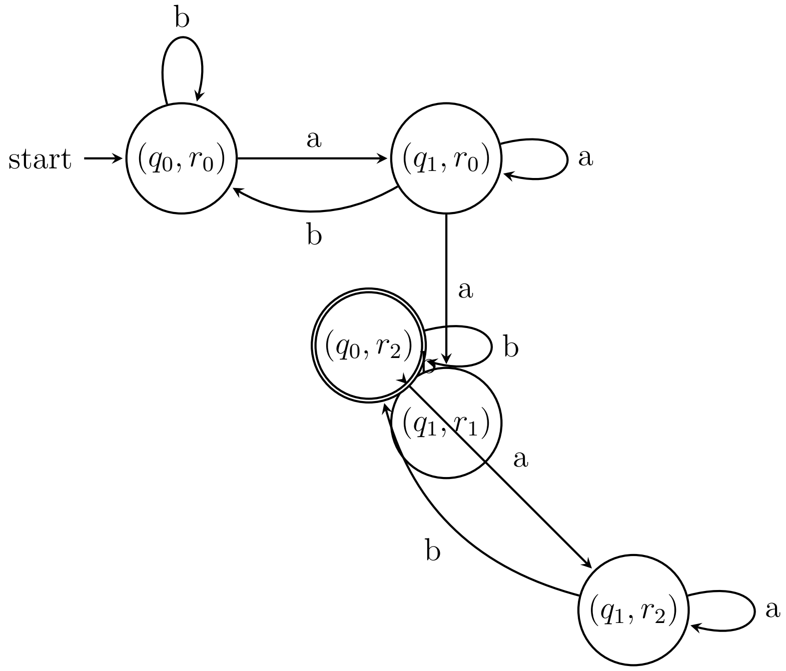

Figure 14: NFA per il linguaggio $L_1 \cap L_2$

b) NFA per $L_1 \cap L_2$:

Per costruire un NFA che riconosca l'intersezione di due linguaggi regolari, possiamo utilizzare il prodotto cartesiano degli stati dei due automi originali.

Tabella di transizione:

	a	b
$\to (q_0, r_0)$	$\{(q_1, r_0), (q_1, r_1)\}$	$\{(q_0, r_0)\}$
(q_1, r_0)	$\{(q_1, r_0), (q_1, r_1)\}$	$\{(q_0, r_0)\}$
(q_1, r_1)	$\emptyset$	$\{(q_0, r_2)\}$
$*(q_0, r_2)$	$\{(q_1, r_2)\}$	$\{(q_0, r_2)\}$
(q_1, r_2)	$\{(q_1, r_2)\}$	$\{(q_0, r_2)\}$

c) Interpretazione del linguaggio $L_1 \cap L_2$ in linguaggio naturale:

Il linguaggio $L_1 \cap L_2$ rappresenta l'insieme di tutte le stringhe che soddisfano entrambe le seguenti condizioni:

- Ogni 'a' è seguita da almeno una 'b'
- La stringa contiene la sottostringa 'ab'

In linguaggio naturale, possiamo descriverlo come: "L'insieme delle stringhe che contengono almeno una volta la sottostringa 'ab' e dove ogni 'a' nella stringa è seguita (non necessariamente immediatamente) da almeno una 'b'."

Una caratterizzazione alternativa potrebbe essere: "L'insieme delle stringhe che contengono almeno una 'a' seguita da almeno una 'b' e dove ogni 'a' è eventualmente seguita da una 'b'."

Esercizio 10. Sia $L = \{w \in \{0,1\}^* \mid w \text{ contiene un numero pari di } 0\}$.

a) Progettare un DFA A che riconosca L.

b) Costruire un DFA che riconosca il complemento di L, ovvero $\overline{L} = \{w \in \{0,1\}^* \mid w \text{ contiene un numero dispari di } 0\}$.

c) Costruire un NFA che riconosca $L^* = \{w_1 w_2 \dots w_k \mid k \geq 0 \text{ e } w_i \in L \text{ per ogni } i\}$.

Soluzione.

a) DFA per $L = \{w \in \{0,1\}^* \mid w \text{ contiene un numero pari di } 0\}$:

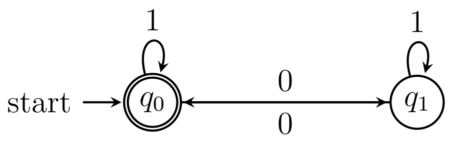

Figure 15: DFA per il linguaggio $L = \{w \in \{0,1\}^* \mid w \text{ contiene un numero pari di } 0\}$

Tabella di transizione:

	0	1
$\rightarrow *q_0$	q_1	q_0
q_1	q_0	q_1

b) DFA per $\overline{L} = \{w \in \{0,1\}^* \mid w \text{ contiene un numero dispari di } 0\}$:

Per ottenere un DFA che riconosca il complemento di un linguaggio, basta invertire gli stati finali e non finali nel DFA originale.

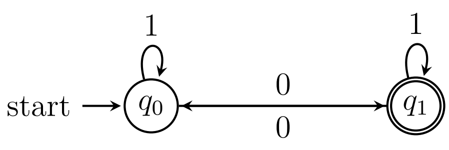

Figure 16: DFA per il linguaggio $\overline{L} = \{w \in \{0,1\}^* \mid w \text{ contiene un numero dispari di } 0\}$

Tabella di transizione:

	0	1
$\rightarrow q_0$	q_1	q_0
$*q_1$	q_0	q_1

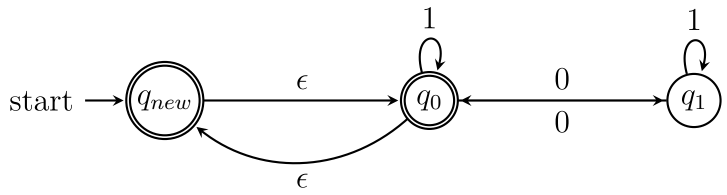

Figure 17: NFA per il linguaggio L^*

c) NFA per $L^* = \{w_1 w_2 \dots w_k \mid k \geq 0 \text{ e } w_i \in L \text{ per ogni } i\}$:

Per costruire un NFA che riconosca L^*, possiamo partire dal DFA che riconosce L e aggiungere un nuovo stato iniziale che è anche finale, con ϵ-transizioni opportune.

Tabella di transizione:

	0	1	ϵ
$\rightarrow *q_{new}$	$\emptyset$	$\emptyset$	$\{q_0\}$
$*q_0$	$\{q_1\}$	$\{q_0\}$	$\{q_{new}\}$
q_1	$\{q_0\}$	$\{q_1\}$	$\emptyset$

Notiamo che $L^* = \Sigma^*$ in questo caso particolare, poiché L contiene la stringa vuota (il numero di 0 nella stringa vuota è 0, che è pari) e quindi L^* contiene tutte le stringhe sull'alfabeto $\{0, 1\}$.