NFA - Slide

Gabriel Rovesti

1 Esercizio 1

Testo: *Determinare il DFA equivalente all'NFA con la seguente tabella di transizione, e stabilire quale linguaggio viene accettato.*

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

Soluzione

L'NFA ha tre stati: q_0 (iniziale), q_1 e q_2 (finale). Dalle funzioni di transizione:

- Da q_0, leggendo 0, rimaniamo in $\{q_0\}$; leggendo 1, andiamo in $\{q_0, q_1\}$.
- Da q_1, leggendo 0 andiamo in $\{q_1\}$, leggendo 1 in $\{q_0, q_2\}$.
- Da q_2 (finale), leggendo 0 andiamo in $\{q_1, q_2\}$, leggendo 1 in $\{q_0, q_1, q_2\}$.

Costruzione del DFA (Subset Construction). Identifichiamo come stato iniziale del DFA $S_0 = \{q_0\}$. Dopodiché:

$$\delta'(S_0, 0) = \{q_0\}, \quad \delta'(S_0, 1) = \{q_0, q_1\}.$$

Chiamiamo $S_1 = \{q_0, q_1\}$. Ora da S_1:

$$\begin{aligned} \delta'(S_1, 0) &= \{\,\delta(q_0, 0) \cup \delta(q_1, 0)\} = \{q_0\} \cup \{q_1\} = \{q_0, q_1\} = S_1, \\ \delta'(S_1, 1) &= \{\,\delta(q_0, 1) \cup \delta(q_1, 1)\} = \{q_0, q_1\} \cup \{q_0, q_2\} = \{q_0, q_1, q_2\}. \end{aligned}$$

Chiamiamo $S_2 = \{q_0, q_1, q_2\}$. Infine da S_2:

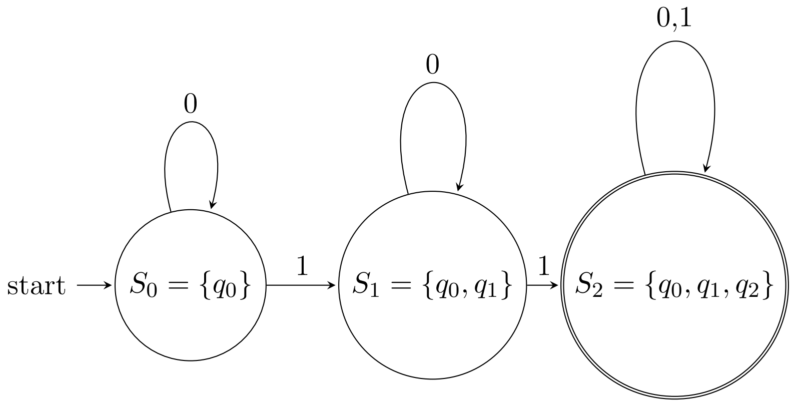
$$\delta'(S_2, 0) = \{q_1, q_2\} \cup \{q_1\} \cup \ldots = \{q_0, q_1, q_2\} \quad \text{e} \quad \delta'(S_2, 1) = \{q_0, q_1, q_2\}.$$

Verificando i dettagli, in pratica da S_2 si resta in S_2 su 0 e 1.

Le *uniche* partizioni di stati effettivamente raggiunte sono:

$$S_0 = \{q_0\}, \quad S_1 = \{q_0, q_1\}, \quad S_2 = \{q_0, q_1, q_2\}.$$

Lo *stato finale* nel DFA è ogni insieme che contenga q_2 (perché q_2 è finale nell'NFA). Dunque S_2 è stato finale; S_0 e S_1 non lo sono.



Linguaggio accettato. Per arrivare in S_2, occorre leggere *almeno due* simboli 1 nella stringa (in qualunque ordine di 0 e 1), perché:

- Dallo stato iniziale $S_0 = \{q_0\}$, la **prima** volta che leggo 1 passo in S_1.
- Da S_1, la **seconda** volta che leggo 1 passo in S_2 (finale).

Dopo di ciò, in S_2 rimango su qualunque ulteriore ingresso. *Conclusione:* $L =$ insieme di stringhe binarie contenenti almeno due simboli 1 (in qualsiasi posizione).

2 Esercizio 2 (variante)

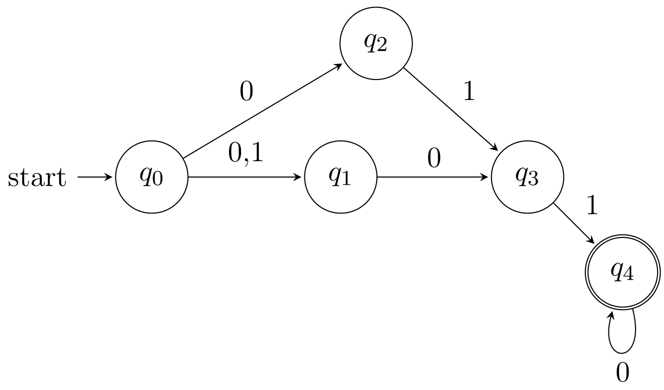
Testo: *La tabella di transizione è la stessa di Esercizio 1. Si richiede nuovamente di determinare il DFA equivalente e descrivere il linguaggio.*

Essendo la stessa specifica di Esercizio 1 (stessi stati e stesse transizioni, con q_2 finale), la soluzione è ovviamente identica:

- Il DFA risultante ha 3 stati: $\{q_0\}$, $\{q_0, q_1\}$, $\{q_0, q_1, q_2\}$, col terzo stato *unico* finale.
- Il linguaggio è "tutte le stringhe su $\{0, 1\}$ con almeno due `1`".

3 Esercizio 3

Testo: *Trasformare il seguente NFA in un DFA equivalente:*



Soluzione

Applicheremo la subset construction, rinominando gli insiemi di stati come S_i. (Si omette qualche passaggio esplicito per brevità.)

1. Stato iniziale $S_0 = \{q_0\}$.

2. Transizioni da S_0

- Con 0: $\delta(q_0, 0) = \{q_1, q_2\}$, dunque $S_1 = \{q_1, q_2\}$.
- Con 1: $\delta(q_0, 1) = \{q_1\}$, dunque $S_2 = \{q_1\}$.

3. Transizioni da $S_1 = \{q_1, q_2\}$

- Con 0: $\delta(q_1, 0) = \{q_3\}$, $\delta(q_2, 0) = \emptyset$, unione $= \{q_3\}$, chiamiamola S_3.
- Con 1: $\delta(q_1, 1) = \emptyset$, $\delta(q_2, 1) = \{q_3\}$, unione $= \{q_3\} = S_3$.

4. Transizioni da $S_2 = \{q_1\}$

- Con 0: $\{q_3\}$ (come sopra).
- Con 1: $\emptyset$ (se q_1 su 1 è vuoto, stando al diagramma).

5. Transizioni da $S_3 = \{q_3\}$

- Con 0: $\emptyset$ (nel disegno, q_3 su 0 non porta da nessuna parte).
- Con 1: $\{q_4\}$, chiamiamola S_4.

6. Transizioni da $S_4 = \{q_4\}$

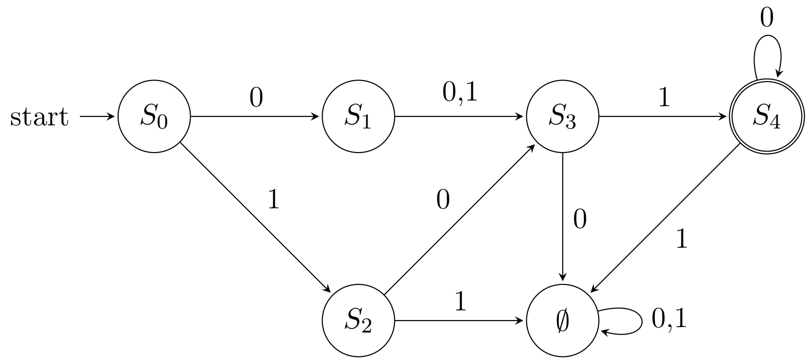
- Con 0: q_4 su 0 rimane in $\{q_4\}$ (loop).
- Con 1: non c'è transizione q_4 su 1 (vuoto), quindi $\emptyset$.

7. Stato $\emptyset$ Rimane come "stato pozzo" per entrambe le lettere $0, 1$.

Raccogliendo:

$S_0 = \{q_0\}$ (*iniziale*)
$S_1 = \{q_1, q_2\}$, $S_2 = \{q_1\}$, $S_3 = \{q_3\}$, $S_4 = \{q_4\}$, $S_5 = \emptyset$ (stato pozzo).

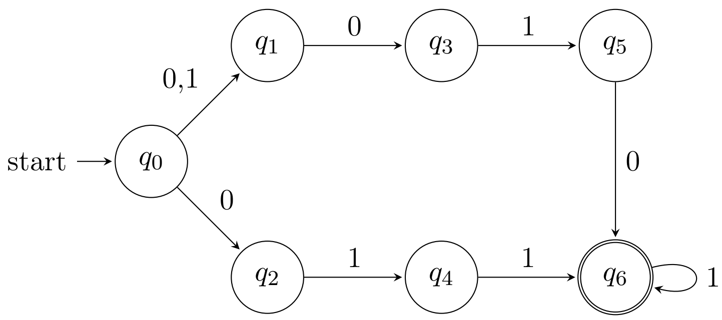
Se q_4 è finale nell'NFA, allora ogni sottoinsieme che contenga q_4 è finale, *i.e.* S_4 lo è. Il DFA risultante è mostrato schematicamente:



4 Esercizio 4

Testo: *Dato il seguente NFA, determinare:*

1. *il linguaggio riconosciuto,*
2. *un DFA equivalente.*



(*Lo schema è puramente indicativo; in caso di discrepanze, bisogna vedere la tabella di transizione effettiva.*)

Soluzione (cenni)

Bisogna capire quali stati siano finali. Se ad esempio q_6 è finale, potremmo individuare tutti i percorsi che portano a q_6 e vedere che cosa implichino sulla struttura delle stringhe lette. In linea di massima, poi, si esegue la costruzione a sottoinsiemi e si ottiene il DFA con gli insiemi di stati. Il *linguaggio* potrebbe includere le stringhe che compiono un certo percorso $q_0 \to q_1 \to q_3 \to q_5 \to q_6$ (magari 0–1–0–1 in quell'ordine), o $q_0 \to q_2 \to q_4 \to q_6$ (magari 0–1–1). Ogni *loop* su q_6 con 1 permette stringhe che finiscono con un certo numero di 1, ecc. Senza la tabella completa e gli stati finali ben definiti, la descrizione resta qualitativa. Il *DFA* si ricava con il processo standard.

5 Esercizio 5

Testo: *Convertire il seguente NFA in DFA:*

	0	1
$\to A$	$\{A, C\}$	$\{B\}$
$*B$	$\{C\}$	$\{B\}$
C	$\{B\}$	$\{D\}$
D	$\emptyset$	$\emptyset$

Soluzione

Per prima cosa individuiamo lo *stato iniziale* $\{A\}$ nel DFA, e andiamo via via costruendo tutti i sottoinsiemi raggiungibili:

$$S_0 = \{A\},$$
$$\delta'(S_0, 0) = \{A, C\}, \quad \delta'(S_0, 1) = \{B\}.$$

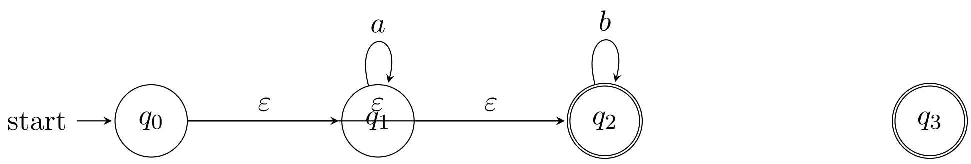
Poi da $\{A, C\}$ si considerano $\delta(A, 0) \cup \delta(C, 0) = \{A, C\} \cup \{B\} = \{A, B, C\}$ ecc. Lo stato B è *finale* nell'NFA, quindi ogni insieme che contenga B risulterà finale nel DFA. Lo stato D è "morto" (non produce transizioni), e se capita in un insieme anch'esso si gestisce come un trap. Lo schema finale del DFA è (in forma di tabella o diagramma) con tutti i sottoinsiemi $\{A\}, \{B\}, \{C\}, \{A, B, C\}, \{B, D\}, \ldots, \emptyset$, segnando come "finali" quelli che includono B.

6 Esercizio 6

Testo: *Costruire un NFA che riconosca le parole costituite da zero o più* `a`*, seguite da zero o più* `b`*, seguite da zero o più* `c`*. Calcolarne la ϵ-chiusura di ogni stato, e infine convertire l'NFA in DFA.*

Soluzione

L'NFA "naturale" è:



Naturalmente, per "tutte le a seguite da tutte le b seguite da tutte le c" si possono usare più stati. Ad esempio:

$$q_0 \xrightarrow{\varepsilon} q_1 \xrightarrow{\varepsilon} q_2, \quad q_1 \text{ con loop } a, \quad q_2 \text{ con loop } b, \quad \ldots$$

E infine un terzo anello per c. Dipende esattamente da come si vogliono distribuire le ε-transizioni. Si calcolano poi le ε-chiusure. Per esempio:

$$\text{ECLOSE}(q_0) = \{\, q_0, q_1, q_2 \}, \quad \text{ECLOSE}(q_1) = \{\, q_1, q_2 \}, \quad \text{ECLOSE}(q_2) = \{\, q_2 \}, \ \ldots$$

e si applica la subset construction standard. Otterremo un DFA i cui stati sono i vari insiemi $\{q_0, q_1, q_2\}, \{q_1, q_2\}, \{q_2\}, \ldots$, e uno stato $\emptyset$ di "pozzo". Tutti gli insiemi che contengano lo stato finale corrispondente (nell'esempio, quello del blocco c) saranno finali.