Linguaggi Formali e Traduttori

2.2 Automi a stati finiti non deterministici (NFA)

Sommario

Automa deterministico

Automa in cui la transizione di stato è **univocamente determinata** dallo stato corrente e dal prossimo simbolo nella stringa da riconoscere

$$\delta : Q \times \Sigma \to Q$$

Automa non deterministico

Automa che può "scegliere" transizioni diverse a parità di stato corrente e prossimo simbolo nella stringa da riconoscere

$$\delta : Q \times \Sigma \to \wp(Q)$$

In questa lezione

1. Introduciamo la classe degli automi a stati finiti **non deterministici**
2. Mostriamo che ogni linguaggio riconosciuto da un automa non deterministico può essere riconosciuto anche da un automa deterministico il quale, però, può avere più stati e/o transizioni di quello non deterministico

Esempio di riconoscimento non deterministico

Problema

Sono in una stanza con un recipiente contenente un numero imprecisato (ma all'apparenza molto grande) di biglie. Ho il compito di svuotare il recipiente e dire "sì" se il numero di biglie è dispari, "no" altrimenti. **Non c'è la lampada!**

Osservazioni

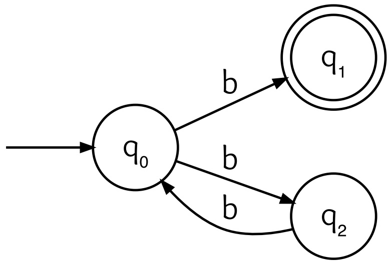
- per ogni $n \geq 2$, il numero $n - 2$ è pari se e solo se n è pari
- per ogni $n \geq 2$, il numero $n - 2$ è dispari se e solo se n è dispari

Soluzione

Mi comporto diversamente in base a quante biglie vedo nel recipiente:

- Se il recipiente è vuoto, dico "no"
- Se il recipiente contiene **una sola biglia**, la rimuovo e dico "sì"
- Se il recipiente contiene **due o più biglie**, ne rimuovo due e ripeto

La soluzione come automa non deterministico



- q_0 = guardo il recipiente e decido cosa fare, se non ci sono più biglie dico “no”
- q_1 = il recipiente conteneva una sola biglia, l’ho rimossa e dico “sì”
- q_2 = il recipiente conteneva due o più biglie, ne ho rimossa una e ora rimuovo l’altra

Automi a stati finiti **non deterministici**

Definizione

Un **automa a stati finiti non deterministico** (detto anche **NFA**, da **N**on-deterministic **F**inite-state **A**utomaton) è una quintupla $A = (Q, \Sigma, \delta, q_0, F)$ dove:

- Q è un insieme **finito** di **stati**
- Σ è l'**alfabeto** riconosciuto dall'automa
- $\delta : Q \times \Sigma \to \wp(Q)$ è la **funzione di transizione** (notare il codominio)
- $q_0 \in Q$ è lo **stato iniziale**
- $F \subseteq Q$ è l'insieme di **stati finali**

Note

- $\delta(q, a)$ è l'insime degli stati in cui l'NFA può scegliere di transire quando si trova nello stato q e legge il simbolo a
- se $\delta(q, a)$ è un **singoletto**, c'è una sola scelta (è il caso deterministico)
- se $\delta(q, a)$ è **vuoto** l'automa **rifiuta** la stringa

Linguaggio riconosciuto da un NFA

Definizione

La **funzione di transizione estesa** dell'NFA $A = (Q, \Sigma, \delta, q_0, F)$ è la funzione $\hat{\delta} : Q \times \Sigma^* \to \wp(Q)$ definita per induzione sul suo secondo argomento come segue:

$$\hat{\delta}(q, \varepsilon) = \{q\} \qquad \hat{\delta}(q, wa) = \{r \in \delta(p, a) \mid p \in \hat{\delta}(q, w)\}$$

Definizione

Il **linguaggio riconosciuto** (o **accettato**) dall'NFA $A = (Q, \Sigma, \delta, q_0, F)$ è denotato da $L(A)$ e definito come segue:

$$L(A) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

Nota

- L'NFA riconosce una stringa w se **esiste** un percorso etichettato con w che lo porta dallo stato iniziale q_0 a uno dei suoi stati finali in F.

Rappresentazione tabellare di NFA

Automa in slide 4

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

Osservazioni

- gli insiemi **singoletto** indicano transizioni **deterministiche**
- l'insieme **vuoto** indica che l'NFA "non sa cosa fare" e **rifiuta** la stringa
- gli altri insiemi indicano transizioni **non** deterministiche (scelte)

DFA $\longrightarrow$ NFA

Teorema

Dato un DFA D, esiste un NFA N tale che $L(N) = L(D)$

Dimostrazione

Dato un DFA $D = (Q, \Sigma, \delta_D, q_0, F)$ definiamo $N = (Q, \Sigma, \delta_N, q_0, F)$ dove

$$\delta_N(q, a) = \{\delta_D(q, a)\}$$

Si può dimostrare, per induzione su $|w|$, che

$$\hat{\delta}_D(q_0, w) = p \iff \hat{\delta}_N(q_0, w) = \{p\}$$

da cui si conclude che

$$\hat{\delta}_D(q_0, w) \in F \iff \hat{\delta}_N(q_0, w) \cap F \neq \emptyset$$

Conseguenze

- ogni linguaggio regolare (cioè riconosciuto da un DFA) è riconosciuto da un NFA
- il potere riconoscitivo degli NFA è almeno pari a quello dei DFA

NFA $\longrightarrow$ DFA

Teorema

Dato un NFA N, esiste un DFA D tale che $L(D) = L(N)$

Intuizione

- creiamo un DFA i cui stati sono **insiemi di stati** dell'NFA
- il DFA traccia **tutti gli stati** in cui l'NFA si può trovare durante il riconoscimento di una stringa, ovvero il DFA traccia **tutte le scelte** possibili che l'NFA può fare
- siccome l'NFA ha un numero **finito** di stati (diciamo n), anche gli stati del DFA lo sono (al massimo 2^n)

Conseguenze

- ogni linguaggio riconosciuto da un NFA è **regolare**
- combinando questo risultato e quello della slide 8, concludiamo che NFA e DFA hanno lo **stesso potere riconoscitivo**

NFA $\longrightarrow$ DFA: costruzione per sottoinsiemi

Dato un NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ definiamo $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ dove

- $Q_D = \wp(Q_N)$, ovvero Q_D è l'insieme dei sottoinsiemi di Q_N
- per ogni $S \subseteq Q_N$ e ogni $a \in \Sigma$ definiamo $\delta_D(S, a) = \bigcup_{q \in S} \delta_N(q, a)$
- $F_D = \{S \subseteq Q_N \mid S \cap F_N \neq \emptyset\}$

Se si dimostra l'equazione

$$\hat{\delta}_N(q_0, w) = \hat{\delta}_D(\{q_0\}, w)$$

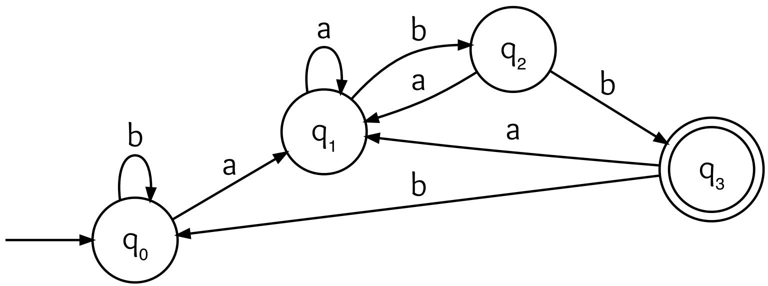
si può concludere che

$$\begin{aligned}
w \in L(N) \quad &\iff \quad \hat{\delta}_N(q_0, w) \cap F_N \neq \emptyset && \text{def. di } L(N) \\
&\iff \quad \hat{\delta}_D(\{q_0\}, w) \cap F_N \neq \emptyset && \text{equazione qui sopra} \\
&\iff \quad \hat{\delta}_D(\{q_0\}, w) \in F_D && \text{def. di } F_D \\
&\iff \quad w \in L(D) && \text{def. di } L(D)
\end{aligned}$$

La dimostrazione è una semplice induzione su $|w|$ (dettagli nel libro di testo)

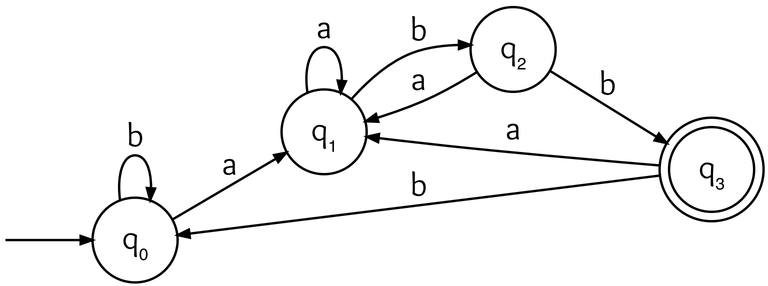
Esempio: stringhe che terminano con abb

Soluzione deterministica

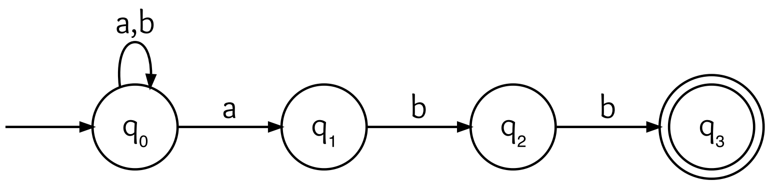


Esempio: stringhe che terminano con abb

Soluzione deterministica



Soluzione non deterministica

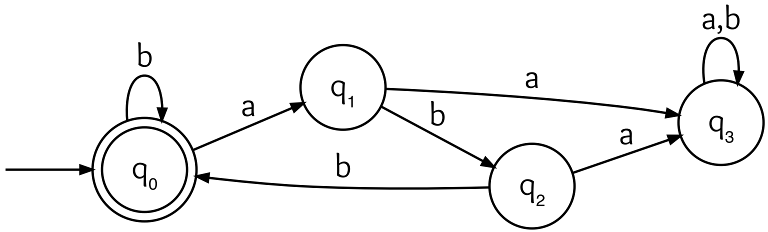


- quando l'automa è nello stato q_0 e legge una a, può **scegliere** se restare in q_0 oppure spostarsi in q_1 e avvicinarsi allo stato finale
- è come se l'automa sapesse qual è la a che annuncia il suffisso abb (quando c'è)
- l'automa non deterministico ha **meno transizioni** di quello deterministico

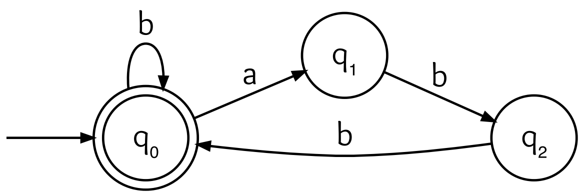
Esempio: ogni a è seguita da bb

Definire un automa che riconosce le stringhe in cui **ogni** a è seguita da bb

Soluzione deterministica

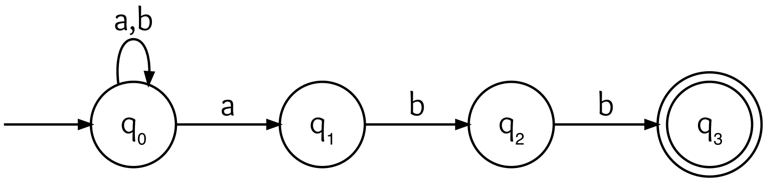


Soluzione non deterministica

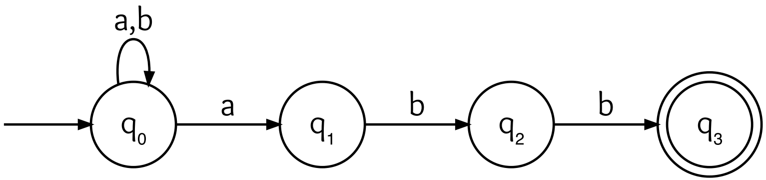


- l'automa non deterministico ha **meno stati** e **meno transizioni** di quello deterministico

Esempio: stringhe che terminano con abb

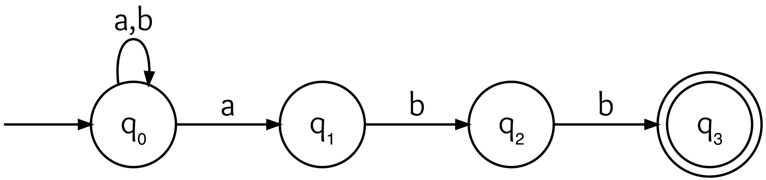


Esempio: stringhe che terminano con abb

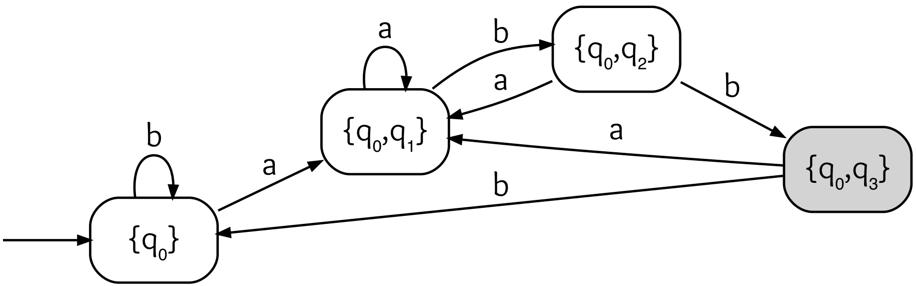


Stato	a	b
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
$*\{q_0, q_3\}$	$\{q_0, q_1\}$	$\{q_0\}$

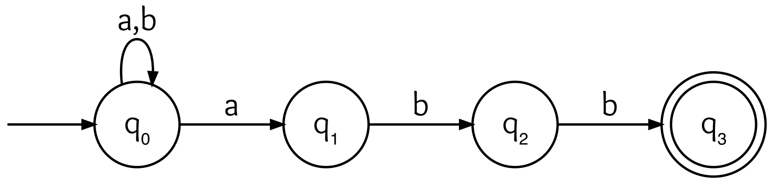
Esempio: stringhe che terminano con abb



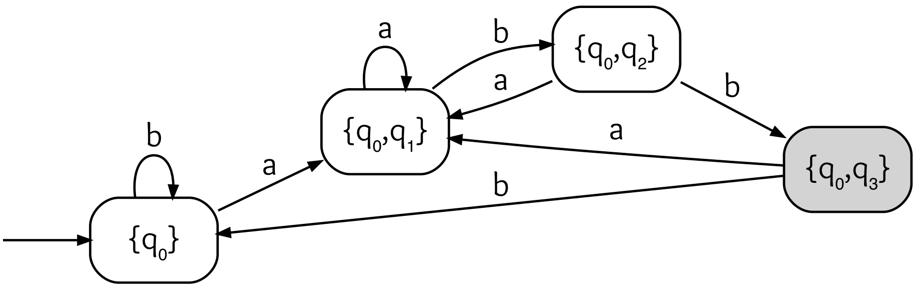
Stato	a	b
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
$*\{q_0, q_3\}$	$\{q_0, q_1\}$	$\{q_0\}$



Esempio: stringhe che terminano con abb



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



Anziché considerare **tutti** i sottoinsiemi di stati dell'NFA, scopriamo man mano quelli che sono raggiungibili dallo stato iniziale $\{q_0\}$

Il DFA ottenuto è isomorfo (anche se non identico) a quello della slide 11 (l'unico stato finale ha lo sfondo grigio)

Il **nome** che diamo agli stati **non influenza** il linguaggio riconosciuto

Esercizi

1. Convertire in DFA l'NFA della slide 12
2. Definire un NFA che riconosce le stringhe di 0 e 1 in cui il terzultimo simbolo è un 1
3. Convertire in DFA l'NFA dell'esercizio precedente
4. Disegnare i diagrammi di transizione dei seguenti NFA e convertirli in DFA

	0	1
$\rightarrow p$	$\{p, q\}$	$\{p\}$
q	$\{r\}$	$\{r\}$
r	$\{s\}$	$\emptyset$
$*s$	$\{s\}$	$\{s\}$

	0	1
$\rightarrow p$	$\{q, s\}$	$\{q\}$
$*q$	$\{r\}$	$\{q, r\}$
r	$\{s\}$	$\{p\}$
$*s$	$\emptyset$	$\{p\}$

5. Definire un NFA sull'alfabeto { a, c, e, n, s } che riconosca le parole cane, casa e cena, poi convertirlo in DFA