UNIVERSITÀ
DEGLI STUDI
DI TORINO

Automi a stati finiti

a.a. 2018-2019

Riferimenti bibliografici

Automi, Linguaggi e Calcolabilità, J. E. Hopcroft, R. Motwani, J. D. Ullman

Automi a stati finiti [cap. 2]

2.1 Una descrizione informale degli automi a stati finiti. (leggere).

2.2 Automi a stati finiti deterministici.

2.3 Automi a stati finiti non deterministici.

(Tranne *Esempio 2.9 e paragrafi* 2.3.5 e 2.3.6)

2.4 Un'applicazione: ricerche testuali. (leggere).

2.5 Automi a stati finiti con epsilon-transizioni.

(*Teorema 2.22 solo enunciato, senza dimostrazione*.)

Lessemi: descrizione e riconoscimento

Come descrivere i lessemi e come riconoscerli?

a) Ogni categoria sintattica è formata da lessemi, che soddisfano un pattern su un alfabeto

Insiemi di stringhe

Linguaggio

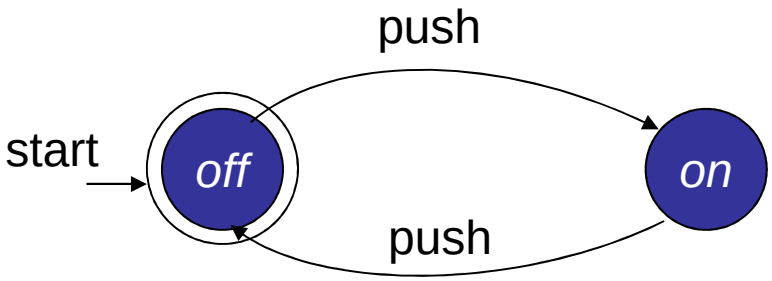
b) La descrizione dei lessemi (il pattern) è fornita da espressioni regolari o altre forme di descrizione come le grammatiche

c) Il riconoscimento è fatto per mezzo di una macchina (un algoritmo che la implementa)

Verso gli “automi a stati finiti”

Esempio: un semplice interruttore ha due *stati*, *on* e *off*.

Possiamo rappresentarlo con un diagramma di questo tipo:



- C’è uno stato iniziale “*off* ”
- Un’azione sull’interruttore (push) fa cambiare lo stato (*transizione*).
- C’è uno stato “finale”, ancora “*off* ”, quando le operazioni compiute sono finite.
- La macchina non è in grado di ricordare se non per mezzo degli stati.

Gli “automi a stati finiti” come riconoscitori di linguaggi

Nel caso dei linguaggi

- le azioni corrispondono alla lettura dei simboli della stringa su un ‘’nastro’’ di input,
- quando l’automa inizia a leggere il controllo si trova nello stato iniziale e,
- se si trova in uno stato finale quando la lettura termina, la risposta è “*si”*, in caso contrario la risposta è “*no”*.

Ad esempio, cosa possono ricordare gli stati se l’automa deve rispondere “si” quando esamina:

- una stringa sull’alfabeto {0,1} delle *che presenta almeno un’occorrenza della sottostringa 011*, oppure
- una stringa sull’alfabeto {0,1,**.**} che rappresenta la notazione in *binari*o di un numero razionale,

Verso gli “automi a stati finiti”

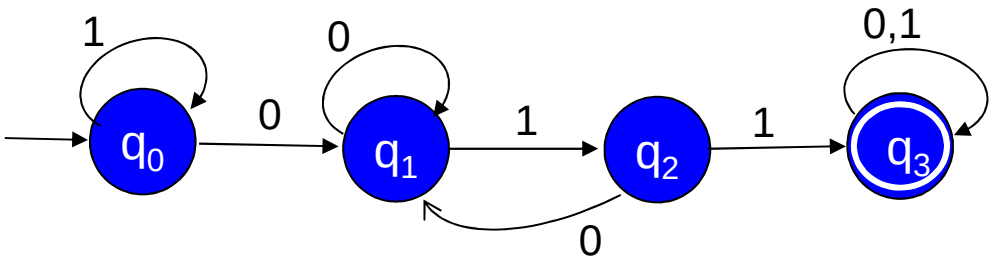
Linguaggio delle stringhe su {0,1} che hanno almeno un’occorrenza della sottostringa 011.

$\Sigma = \{0,1\}$ $\quad\quad$ $L = \{0,1\}^* \{011\} \{0,1\}^*$

Gli stati servono a ricordare e distinguere i diversi casi che si possono presentare esaminando una stringa sull’alfabeto {0,1}:

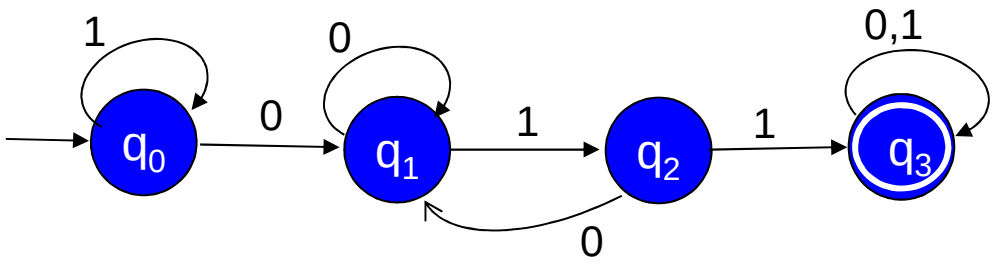
1. La stringa esaminata non contiene 0 (q_0)
2. L’ultimo carattere è 0 (q_1)
3. Gli ultimi due caratteri sono 01 (q_2)
4. È stata incontrata la sottostringa 011 (q_3)

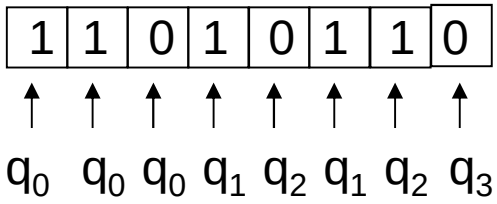
Diagramma di transizione



Verso gli “automi a stati finiti”

Diagramma di transizione





Nastro di input

Stati del controllo

Verso gli “automi a stati finiti”

Linguaggio dei numeri binari

$$\Sigma = \{0,1,.\}$$

$$L = (\{0\} \cup \{1\}\ \{0,1\}^*)\ \{.\}\ \{0,1\}^+$$

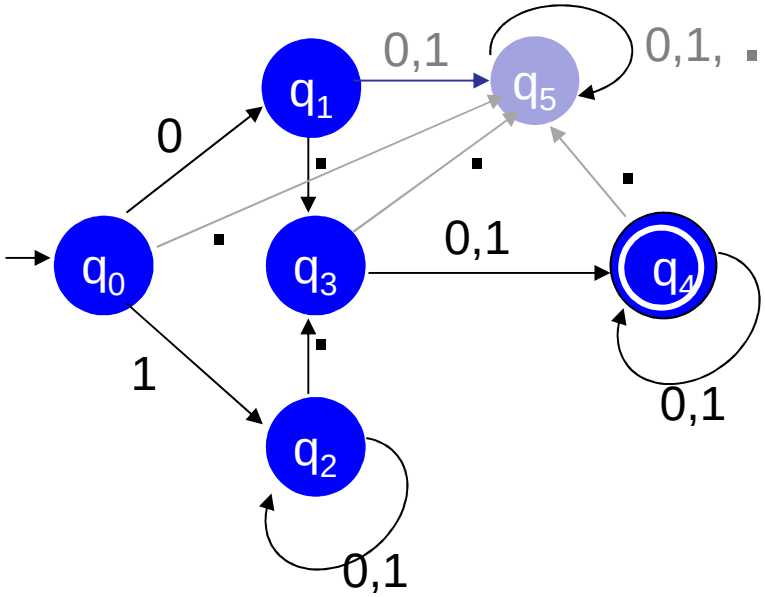
Quali casi si possono presentare esaminando una stringa sull’alfabeto {0,1,.}?. Ad esempio 101.00 e 0.1 appartengono a L, mentre 101. e 0 non vi appartengono.

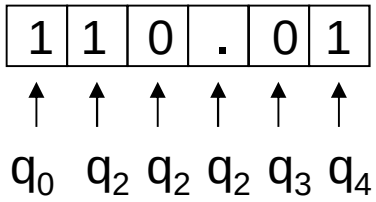
1. nessun simbolo è ancora stato esaminato
2. il primo carattere è 0
3. il primo carattere è 1
4. è stato incontrato il punto
5. si sta esaminando la parte dopo il punto
6. la stringa in esame non può appartenere al linguaggio

Verso gli “automi a stati finiti”

1. nessun simbolo è ancora stato esaminato (q_0 stato iniziale)
2. il primo carattere è 0 (q_1)
3. il primo carattere è 1 e succesivi caratteri sono 0 o1 (q_2)
4. è stato incontrato il punto (q_3)
5. si sta esaminando la parte dopo il punto (q_4)
6. la stringa in esame non può appartenere al linguaggio (q_5)

Diagramma di transizione



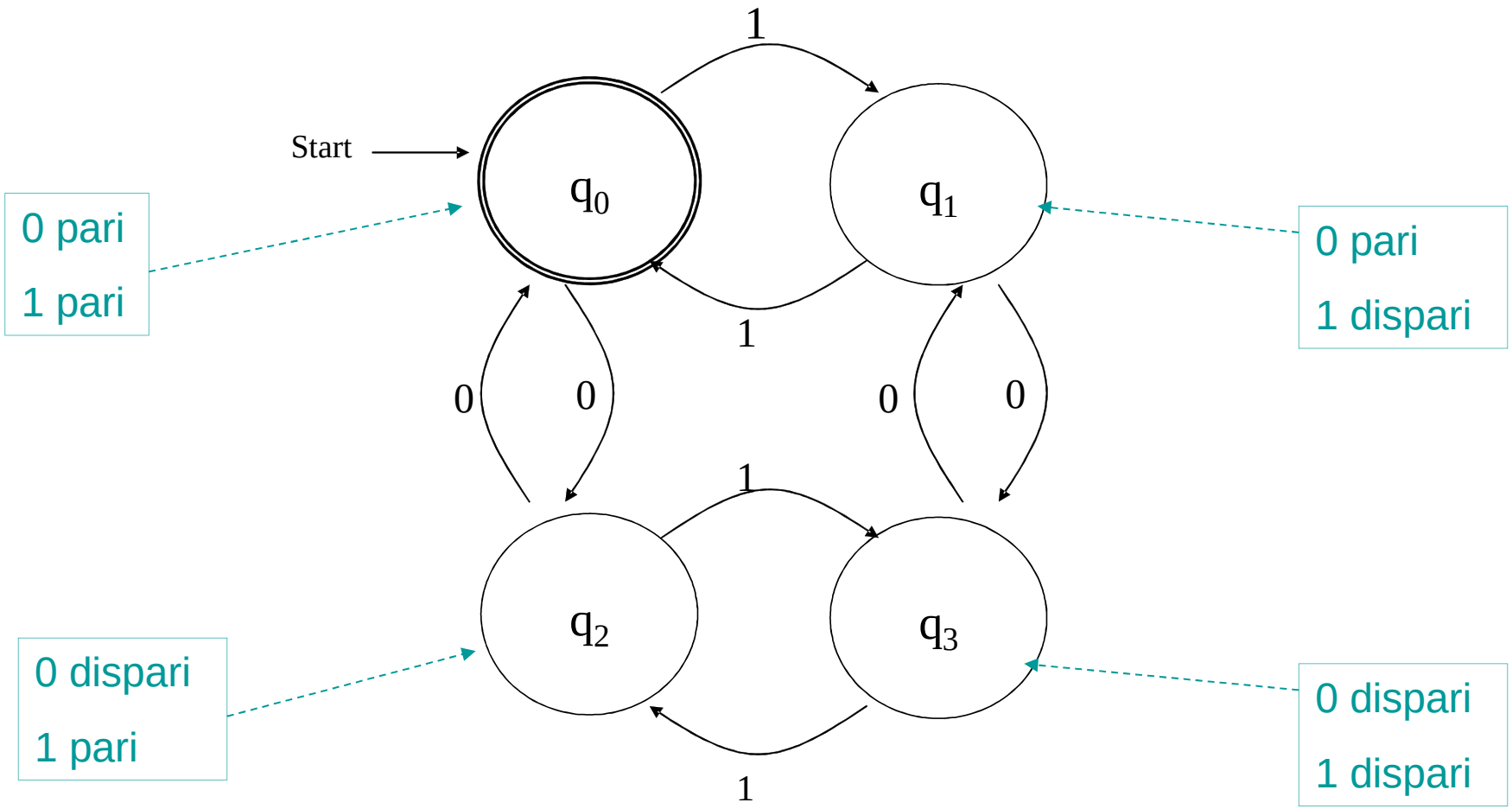


Nastro di input

Stati del controllo

Un altro esempio di automa

Esempio: un automa che riconosce le stringhe di 0 e 1 in cui il numero di occorrenze di 0 e 1 sono *entrambe* pari.



Esercizio: trovare un e.r. che denoti lo stesso linguaggio

Automa finito deterministico: DFA. Definizione formale

- Un DFA è una quintupla

$$A = (Q, \Sigma, \delta, q_0, F)$$

  - Q è un insieme finito di *stati*
  - Σ è un alfabeto finito (*simboli in input*)
  - δ è una *funzione di transizione* $Q \times \Sigma \rightarrow Q$
  - $q_0 \in Q$ è lo stato *iniziale*
  - $F \subseteq Q$ è l'insieme degli stati *finali*

La funzione δ può essere rappresentata in tanti modi:
- dando la definizione per punti
- In forma di tabella
- mediante il *diagramma di transizione*

Il diagramma di transizione o la tabella contengono tutti gli elementi che definiscono l'automa. Possono essere considerate una *definizione completa* dell'automa.

Automa finito deterministico: DFA

Se rappresentiamo l'azione di push con il simbolo "*p*" possiamo rappresentare l'interruttore con il seguente automa a stati finiti:

$I = (\{on, off\}, \{p\}, \delta, off, \{off\})$

$$\delta: \begin{cases} \delta(on, p) = off \\ \delta(off, p) = on \end{cases}$$

Diagramma di transizione

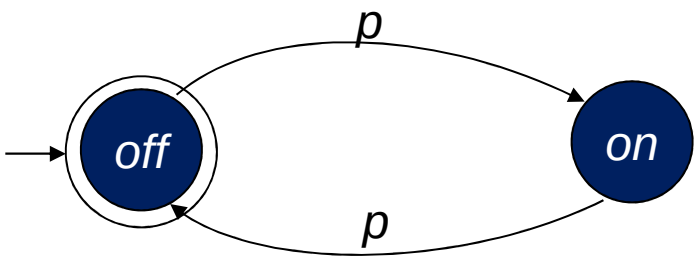


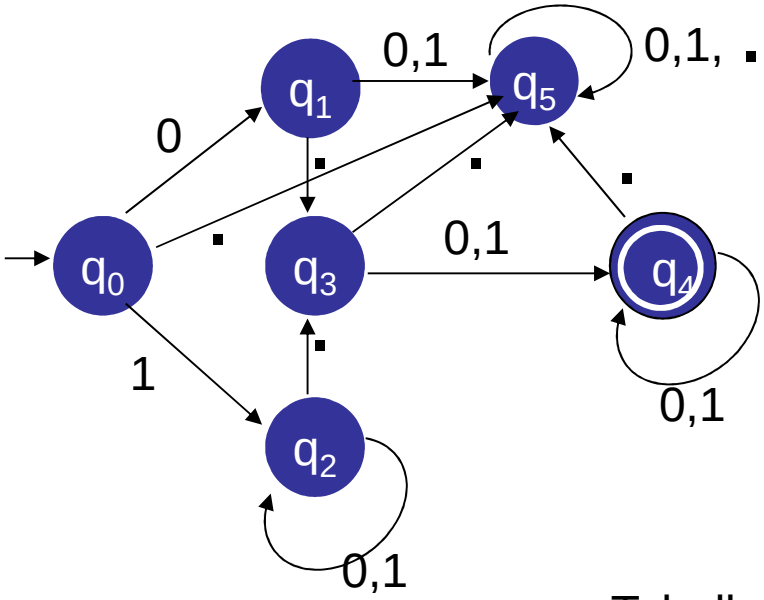
Tabella di transizione

	p
→**off*	*on*
on	*off*

Il linguaggio riconosciuto sarà allora: $\{pp\}^*$.

Rappresentazione di automi

Diagramma di transizione



$M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{0, 1, .\}, \delta, q_0, \{q_4\})$

$\delta(q_0,0) = q_1$	$\delta(q_2,0) = q_2$	$\delta(q_4,0) = q_4$
$\delta(q_0,1) = q_2$	$\delta(q_2,1) = q_2$	$\delta(q_4,1) = q_4$
$\delta(q_0,.) = q_5$	$\delta(q_2,.) = q_3$	$\delta(q_4,.) = q_5$
$\delta(q_1,0) = q_5$	$\delta(q_3,0) = q_4$	$\delta(q_5,0) = q_5$
$\delta(q_1,1) = q_5$	$\delta(q_3,1) = q_4$	$\delta(q_5,1) = q_5$
$\delta(q_1,.) = q_3$	$\delta(q_3,.) = q_5$	$\delta(q_5,.) = q_5$

Tabella di transizione

	0	1	.
→ q_0	q_1	q_2	q_5
q_1	q_5	q_5	q_3
q_2	q_2	q_2	q_3
q_3	q_4	q_4	q_5
*q_4	q_4	q_4	q_5
q_5	q_5	q_5	q_5

Automa finito deterministico: DFA

- La funzione di transizione si può estendere alle stringhe: $\hat{\delta}$: Q x Σ^* $\rightarrow$ Q

$$\hat{\delta}(q, \varepsilon) = q$$
$$\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$$

- Il linguaggio riconosciuto (o accettato) da A è

$$L(A) = \{w \mid \hat{\delta}(q_0, w) \in F\}$$

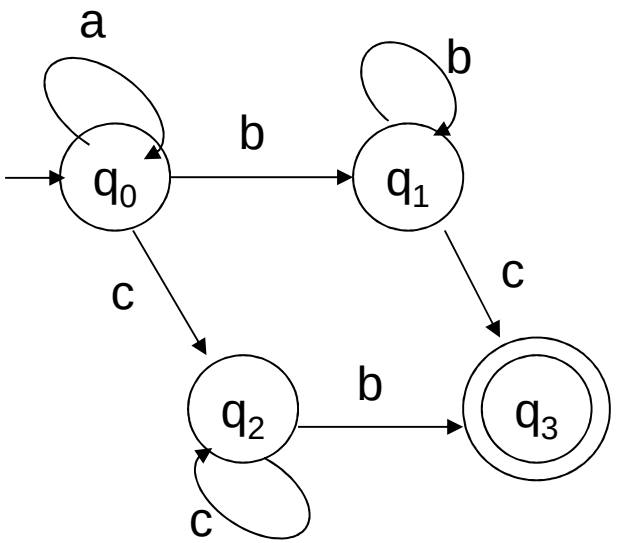
- I linguaggi riconosciuti (accettati) da automi a stati finiti sono chiamati *linguaggi regolari.*

- Due automi sono *equivalenti* se accettano lo stesso linguaggio.

Esempio

$< \{q_0,q_1,q_2,q_3\}, \{a, b, c\}, \delta, q_0, \{q_3\} >$

$\delta(q_0,a) = q_0$
$\delta(q_0,b) = q_1$
$\delta(q_0,c) = q_2$
$\delta(q_1,b) = q_1$
$\delta(q_1,c) = q_3$
$\delta(q_2,c) = q_2$
$\delta(q_2,b) = q_3$
. . . . . .



La definizione di $\delta(q1,a)$, $\delta(q2,a)$, $\delta(q3,a)$, $\delta(q3,b)$, $\delta(q3,c)$, è stata omessa, anche dal diagramma di transizione

Automa finito deterministico: DFA

Ricordiamo

$\hat{\delta}(q, \varepsilon) = q$

$\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$

Esempio:

$\hat{\delta}(q_0, aabbc) = \delta(\hat{\delta}(q_0, aabb), c) = \delta(\delta(\hat{\delta}(q_0, aab), b), c) =$

$= \delta(\delta(\delta(\hat{\delta}(q_0, aa), b), b), c) = \delta(\delta(\delta(\delta(\hat{\delta}(q_0,a), a), b), b), c) =$

$= \delta(\delta(\delta(\delta(\delta(\hat{\delta}(q_0, \varepsilon), a), a), b), b), c) = \delta(\delta(\delta(\delta(q_0,a), a), b), b), c) =$

$= \delta(\delta(\delta(q_0,a), b), b), c) = \delta(\delta(q_0,b), b), c) = \delta(q_1,b), c) =$

$= \delta(q_1,c) = q_3$

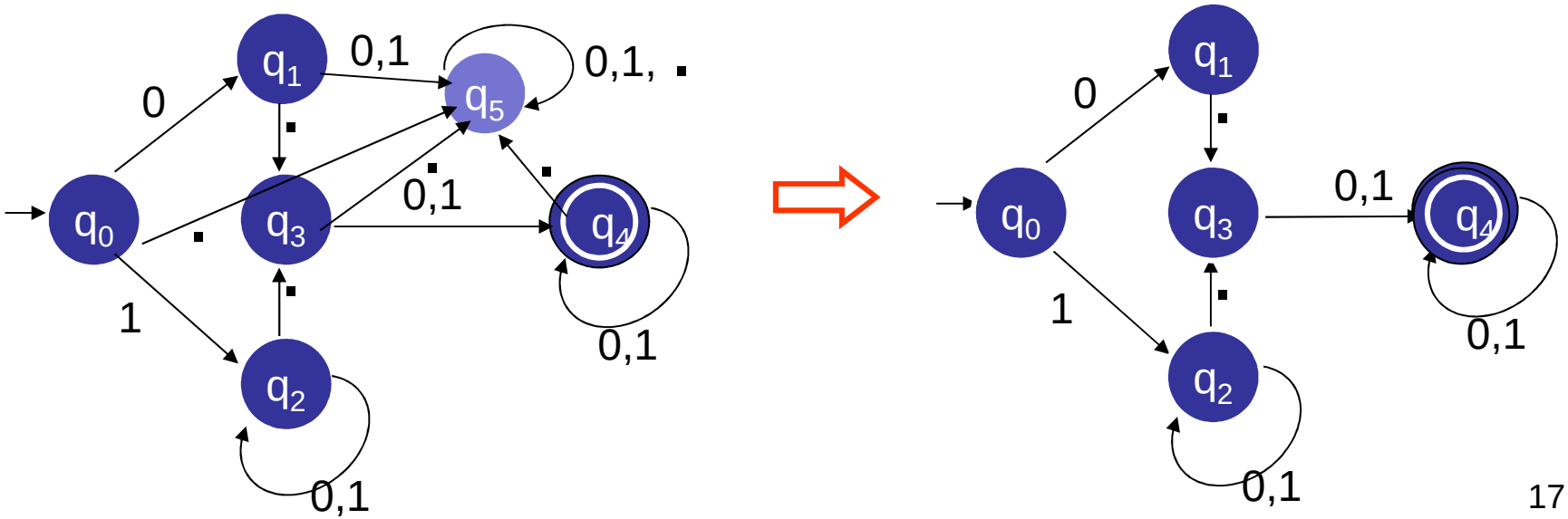
Useremo spesso δ al posto di $\hat{\delta}$ in quanto si comportano nello stesso modo sulle stringhe di lunghezza 1 (caratteri dell'alfabeto).

Automa finito deterministico: stato di errore

Una stringa viene accettata se, quando l'analisi termina, il controllo si trova in uno stato finale.

In alcuni casi, come nell'esempio precedente, l'automa prima di fermarsi può raggiungere uno stato in cui si può continuare la lettura della stringa, ma dal quale il controllo non può più uscire: *stato di errore* (o 'dead state', stato di morte, o 'stato trappola').

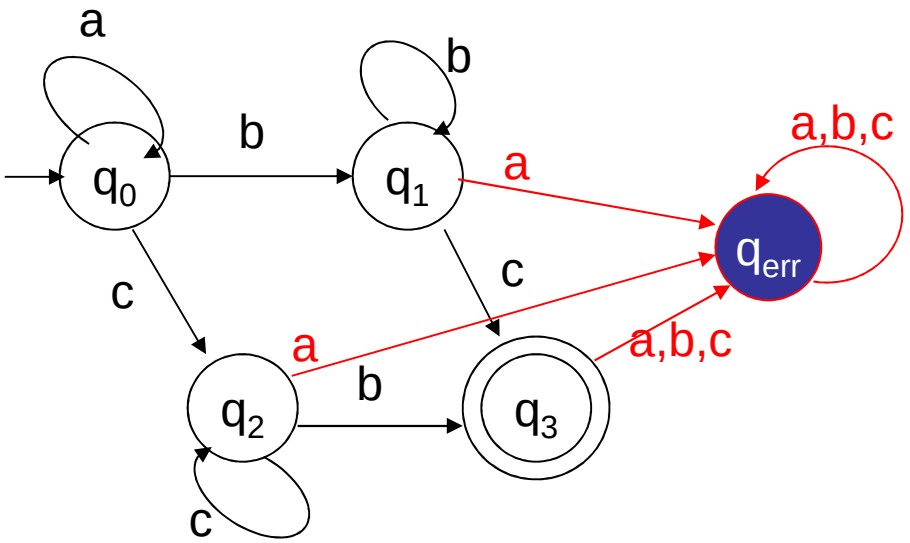
Lo stato di errore di solito viene *sottinteso* per non appesantire il diagramma di transizione o la tabella.



Automa finito deterministico: stato di errore, altro esemp

$< \{q_0,q_1,q_2,q_3\}, \{a, b, c\}, \delta, q_0, \{q_3\} >$

$\delta(q_0,a) = q_0$
$\delta(q_0,b) = q_1$
$\delta(q_0,c) = q_2$
$\delta(q_1,b) = q_1$
$\delta(q_1,c) = q_3$
$\delta(q_2,c) = q_2$
$\delta(q_2,b) = q_3$



Notare che $\delta(q_1,a)$, $\delta(q_2,a)$, $\delta(q_3,a)$, $\delta(q_3,b)$, $\delta(q_3,c)$, non sono esplicitamente definite, la loro definizione è sottintesa, ma la funzione di transizione è comunque una funzione totale.

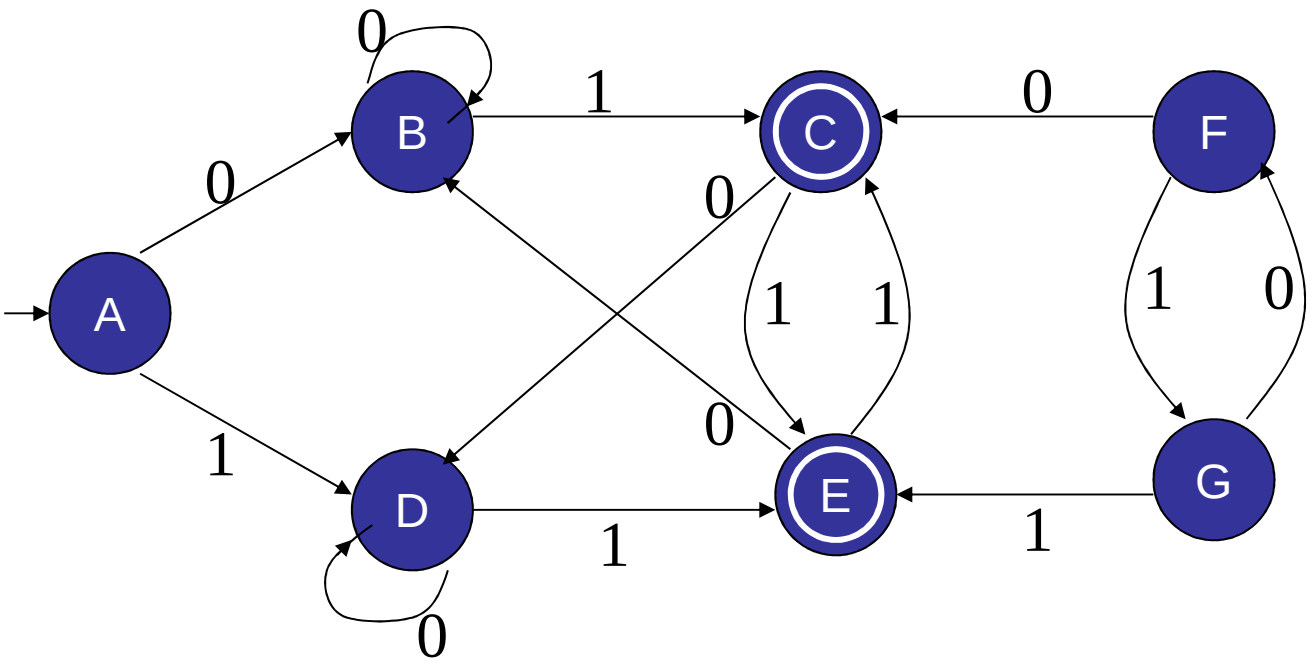
Stati inutili

Come nel caso dei simboli intii delle grammatiche anche per gli automi possiamo definire una nozione di stati *inutili*, che possono quindi essere eliminati dal'automa senza alterare il linguaggio riconosciuto.

Uno stato q e' *utile* se non esiste una stringa x tale che $\delta(q_0, x) = q$, cioe' q e' **accessibile** (raggiungibile dallo stato iniziale).
Uno stato e' **inutile** se non e' accessibile.

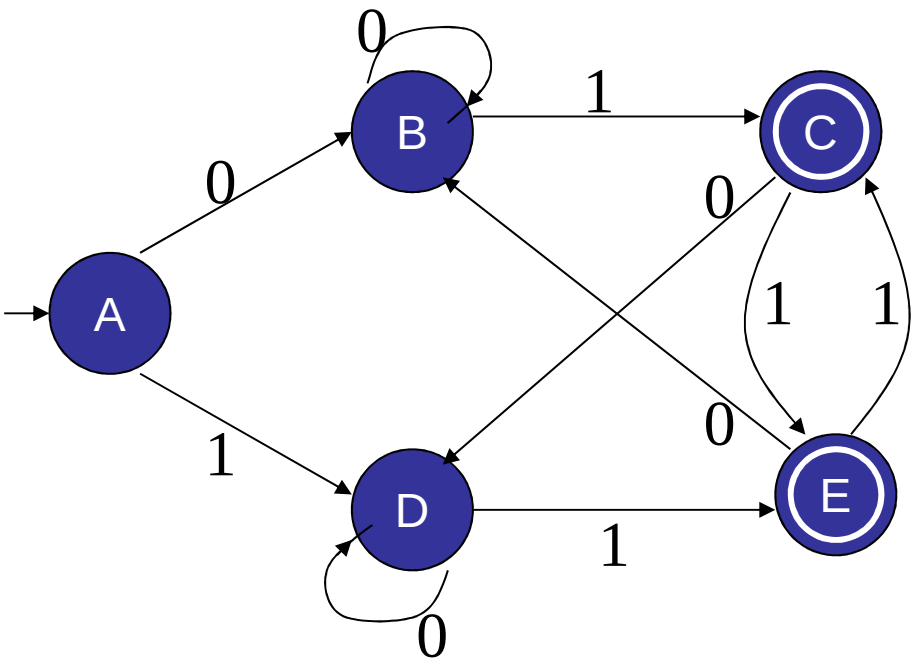
Gli stati inutili possono essere quindi cancellati senza alterare il linguaggio riconosciuto dall'automa.
Tuttavia anche un automa senza stati "inutili" può a volte essere ridotto per ottenere un automa *minimo*.

Esempio



F e G sono stati non raggiungibili dallo stato iniziale e perciò li possiamo eliminare.

Esempio (continua)



Questo automa non ha più stati inutili ma si può ancora trasformare rendendo minimo il numero degli stati.

Esercizi

Si considerino i DFA con le seguenti tabelle di transizione:

δ	0	1
→ A	A	B
*B	B	A

δ	0	1
→ *A	B	A
*B	C	A
C	C	C

Per ognuno dei due DFA descrivere il linguaggio accettato sia informalmente sia per mezzo di un descrittore di insiemi.

Esercizi

Costruire un DFA per ognuno dei seguenti linguaggi sull'alfabeto {0,1}:

- Insieme di tutte le stringhe che finiscono con 00.
- Insieme di tutte le stringhe con tre zero consecutivi.
- Insieme delle stringhe con 011 come sottostringa.
- Insieme delle stringhe che cominciano o finiscono (o entrambe le cose) con 01.
- Insieme di tutte le stringhe che iniziano con almeno due 0 e terminano con almeno due 1.
- Insieme delle stringhe che contengono un numero di 1 divisibile per 5.

Costruire un DFA per ognuno dei seguenti linguaggi sull'alfabeto {a, b}:

- $\{a^n b^n \mid 0 \leq n \leq 3\}$.
- Insieme di tutte le stringhe che non presentano b consecutivi e in cui tutte le sequenze di a siano di lunghezza pari.
- Insieme delle stringhe che contengono un numero di a multiplo di tre.

Esercizi

Disegnare il diagramma di transizione di automi finiti deterministici che riconoscano i linguaggi sull'alfabeto {a, b} che soddisfano una delle seguenti condizioni:

- ogni occorrenza del carattere a sia seguita immediatamente da almeno due occorrenze del carattere b.
- ogni occorrenza del carattere a sia seguita immediatamente da esattamente due occorrenze del carattere b.
- inizino con almeno due a e terminino con almeno due b.