Automi finiti e riconoscimento dei linguaggi regolari

Prof. A. Morzenti

aa 2008-2009

AUTOMI FINITI E RICONOSCIMENTO DEI LINGUAGGI REGOLARI

scaletta:

* automi finiti e linguaggi regolari
* determinismo e non determinismo (o indeterminismo) del riconoscitore
* automa minimo
* composizione di riconoscitori con operazioni per cui famiglia REG e` chiusa
* utilizzo degli automi finiti nella compilazione

ALGORITMI DI RICONOSCIMENTO

DOMINIO: un insieme di stringhe di un certo alfabeto Σ
CODOMINIO: risposte *sì / no*
applicazione algoritmo di riconoscimento α a una stringa data x: $\alpha(x)$
stringa x è (risp. non è) *riconosciuta* o *accettata*
*s*se $\alpha(x)$ = *sì* (risp. $\alpha(x)$ = *no*).

linguaggio riconosciuto da algoritmo α: $L(\alpha) = \{ x \in \Sigma^* \mid \alpha(x)=\text{sì}\}$

Se linguaggio L semidecidibile (o ricorsivamente enumerabile) allora per qualche stringa $\notin$ l'algoritmo può non terminare, cioè $\alpha(x)$ può non essere definita.

In questo corso consideriamo solo linguaggi decidibili

Quasi tutti i problemi di nostro interesse hanno complessità bassa, lineare o al peggio polinomiale rispetto a |x|

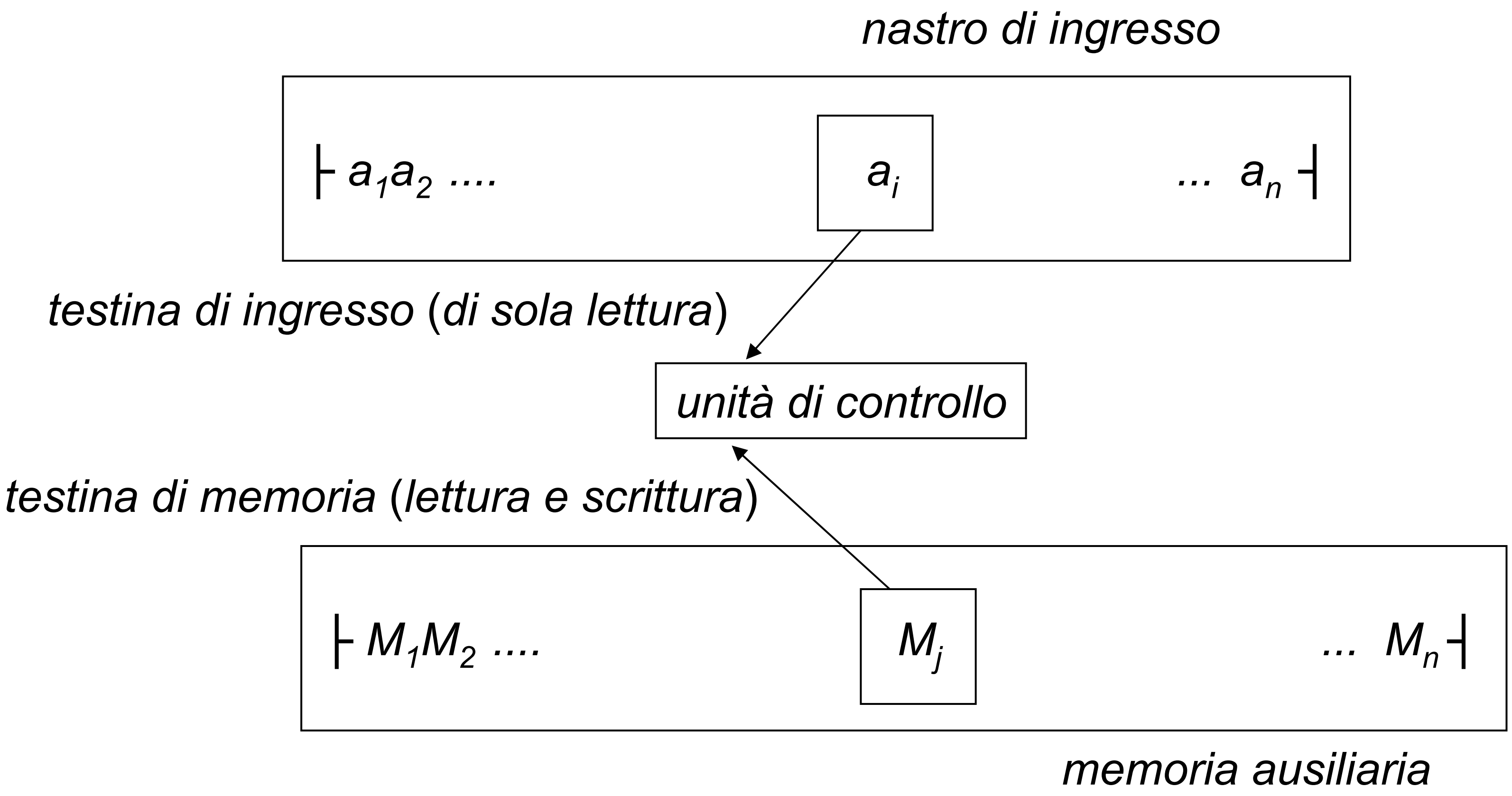
Nella teoria e ingegneria dei linguaggi passo computazionale è mossa di un automa o macchina astratta, capace solo di manipolare un simbolo.

Consuetudine presentare gli algoritmi di riconoscimento e traduzione come automi, per:

* evidenziare correlazione tra famiglie di linguaggi e grammatiche
* evitare riferimenti prematuri ad aspetti realizzativi del software.

facile poi dare indicazioni essenziali per trasformare automi in programmi

SCHEMA DI UN AUTOMA RICONOSCITORE nella sua forma più generale:



UN AUTOMA ESAMINA LA STRINGA DI INGRESSO compiendo una serie di mosse: ogni mossa dipende dai simboli presenti sotto le testine e dallo stato del controllo. Una mossa può avere gli effetti seguenti:

- spostamento della testina di ingresso di una posizione a sinistra o a destra
- scrittura di un simbolo al posto del simbolo corrente nella memoria e spostamento della testina di memoria (di una posizione a destra o a sinistra)
- cambiamento di stato dell'unità di controllo.

Alcune di queste azioni possono mancare.

AUTOMA MONODIREZIONALE: la testina di ingresso si può spostare soltanto da sinistra verso destra. Si considererà questo caso, che corrisponde alle esigenze di analisi con una sola scansione.

MEMORIA AUSILIARIA MANCANTE: automa finito (macchina a stati finiti) – riconoscitore dei linguaggi regolari.

MEMORIA A PILA: automa a pila – riconoscitore dei linguaggi liberi.

CONFIGURAZIONE (istantanea) definita da tre componenti
che determinano il futuro comportamento dell'automa:

- la porzione di nastro d'ingresso non ancora letta (situata alla destra della testina di lettura)
- il contenuto della memoria ausiliaria e la posizione della testina di memoria
- lo stato dell'unità di controllo.

CONFIGURAZIONE INIZIALE:
- testina d'ingresso sul simbolo dopo la marca d'inizio
- l'unità di controllo in stato iniziale
- memoria contiene l'informazione iniziale (di solito un particolare simbolo).

automa evolve attraverso serie di mosse (<u>calcolo o computazione</u>) che cambiano la configurazione

DETERMINISMO – in ogni configurazione possibile non più di una mossa
NON DETERMINISMO (detto anche INDETERMINISMO) in caso contrario

CONFIGURAZIONE FINALE
- controllo in uno stato finale
- testina di ingresso sul terminatore della stringa (variante: condizione sulla memoria: essere vuota o contenere un simbolo specifico

UNA STRINGA SORGENTE x È ACCETTATA DALL'AUTOMA se esso, partendo dalla configurazione iniziale ├ x ┤ in ingresso, esegue una serie di mosse che lo porta in una configurazione finale (un automa non deterministico potrebbe raggiungere la configurazione finale in più modi diversi).

IL CALCOLO TERMINA
- l'automa ha raggiunto una configurazione finale, o
- perché non può eseguire alcuna mossa (mossa indefinita, stringa non accettata)

LINGUAGGIO RICONOSCIUTO (accettato, definito) dall'automa e\`
L'INSIEME DELLE STRINGHE riconosciute dall'automa

DUE AUTOMI che accettano lo stesso linguaggio sono detti EQUIVALENTI.
Non necessario che siano dello stesso tipo né che abbiano la stessa complessità.

I LINGUAGGI REGOLARI, riconosciuti dagli automi finiti, sono una sottofamiglia dei linguaggi riconoscibili in tempo reale da parte di una macchina di Turing.

Molte applicazioni informatiche utilizzano automi finiti:
progettazione digitale, sistemi di controllo, protocolli di comunicazione,
studio dell'affidabilità dei sistemi, …

SI VEDRANNO:

* diagrammi stato-transizione
* automi finiti come riconoscitori di linguaggi definiti da grammatiche unilineari
* comportamento deterministico e non deterministico
* passaggio da comportamento non deterministico a comportamento deterministico
* sottofamiglie caratterizzate da un test locale della memoria dell'automa

DIAGRAMMI STATO-TRANSIZIONE

In un automa finito vi sono:

* il nastro d'ingresso (con stringa da riconoscere $x \in \Sigma^*$)
* unità di controllo con la sua memoria limitata
* testina di lettura (posta inzialmente sul primo carattere di *x* e moventesi da sx a dx con una sola scansione fino alla fine di *x* o al primo errore).

Dopo la lettura di un carattere l'automa aggiorna lo stato dell'unità di controllo.

Alla fine della scansione, l'automa riconosce o meno *x* a seconda dello stato In cui si trova.

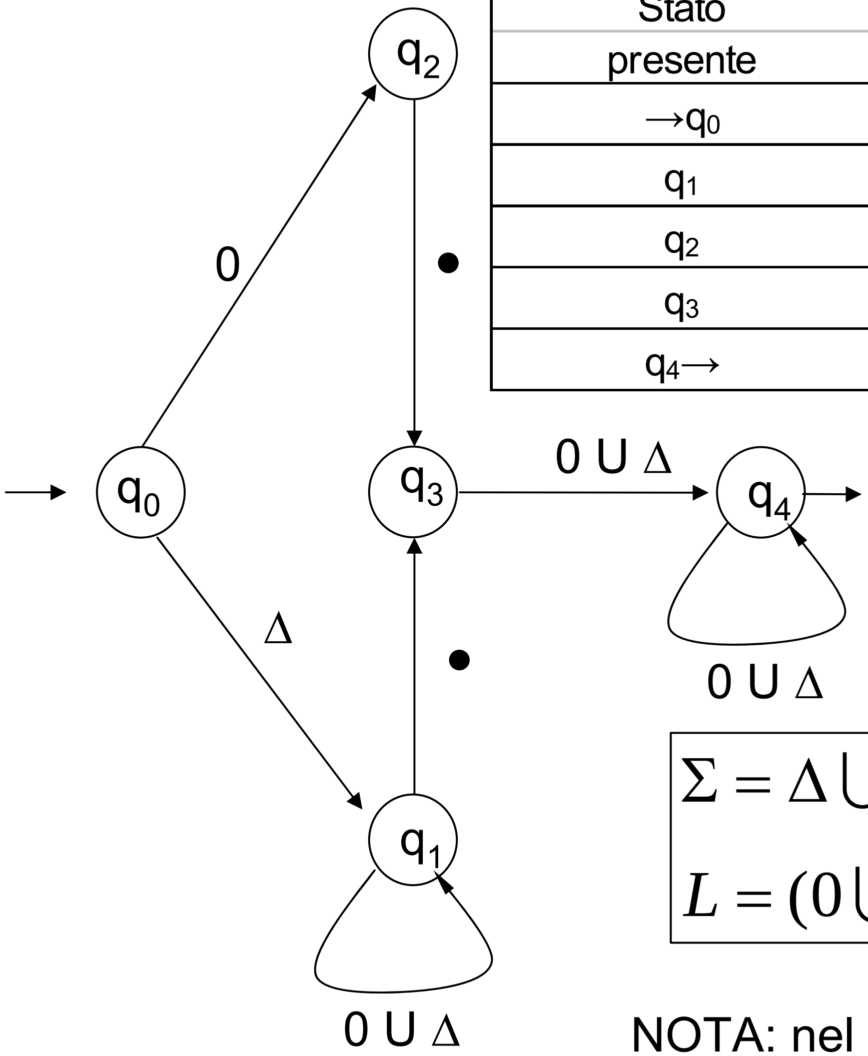
DIAGRAMMI STATO-TRANSIZIONE (continua)

DIAGRAMMA STATO-TRANSIZIONE: è un grafo orientato etichettato
NODI: gli stati dell'unità di controllo
ARCHI: etichettati con caratteri dell'alfabeto di ingresso, indicano transizioni

Il diagramma stato-transizione ha (nella versione deterministica) un solo STATO INIZIALE, ma può avere PIÙ STATI FINALI

La matrice delle incidenze del grafo forma la TABELLA DELLE TRANSIZIONI
Essa riporta per ogni coppia (stato presente, carattere corrente) il prossimo stato

ESEMPIO – Costanti numeriche decimali



Stato	Carattere corrente				
presente	0	1	...	9	•
$\rightarrow q_0$	q_2	q_1	...	q_1	–
q_1	q_1	q_1	...	q_1	q_3
q_2	–	–	...	–	q_3
q_3	q_4	q_4	...	q_4	–
$q_4 \rightarrow$	q_4	q_4	...	q_4	–

$$\Sigma = \Delta \bigcup \{0, \bullet\}, \text{ dove } \Delta = \{1,2,3,4,5,6,7,8,9\}$$

$$L = (0 \bigcup \Delta(0 \bigcup \Delta)^{*}) \bullet (0 \bigcup \Delta)^{+}$$

NOTA: nel disegno sono riuniti gli archi topologicamente uguali (da q_0 a q_1 vi è un fascio di 9 archi ....).

DEFINIZIONE FORMALE DI AUTOMA FINITO DETERMINISTICO

cinque entità:

1. Q, *l'insieme degli stati* (finito e non vuoto)
2. Σ, *l'alfabeto di ingresso* (o terminale)
3. la *funzione di transizione* δ: $(Q \times \Sigma) \rightarrow Q$
4. $q_0 \in Q$, lo *stato iniziale*
5. $F \subseteq Q$, *l'insieme degli stati finali*

La funzione di transizione codifica le mosse dell'automa: $\delta(q_l,a)=q_j$
M, nello stato q_i, leggendo a va nello stato q_j
Se $\delta(q, a)$ non definita, l'automa si ferma (entra in stato di errore non finale e rifiuta)

Estensione δ^* di δ alle stringhe di lunghezza qualsiasi δ: $(Q \times \Sigma^*) \rightarrow Q$

definita induttivamente come $\delta^*(q, \varepsilon)=q$ e $\delta^*(q, xa)=\delta(\delta^*(q, x),a)$, $x \in \Sigma^*$, $a \in \Sigma$

Per brevita` si indica con δ anche la sua estensione δ^*

CALCOLO ESEGUITO DA PARTE UN AUTOMA = CAMMINO SUL GRAFO

RICONOSCIMENTO DI UNA STRINGA:

stringa x riconosciuta (o accettata) dall'automa M se scandendola M passa dallo stato iniziale a uno stato finale:

$$\boldsymbol{\delta(q_0, x) \in F}$$

Quindi la stringa vuota è riconosciuta solo se lo stato iniziale è anche stato finale.

LINGUAGGIO RICONOSCIUTO DALL'AUTOMA M:
INSIEME DELLE STRINGHE RICONOSCIUTE

$$L(M) = \left\{ x \in \Sigma^* \mid x \text{ è riconosciuta da M} \right\}$$

La famiglia dei linguaggi riconosciuti da automi del tipo considerato è detta A STATI FINITI.

Due automi sono equivalenti se riconoscono lo stesso linguaggio.

L'efficienza di un automa deterministico è la migliore possibile in assoluto: l'algoritmo decide l'appartenenza della stringa in tempo reale, scandendola una sola volta da sinistra a destra.

ESEMPIO – Costanti numeriche decimali (continua) - L'automa M è definito da:

$$Q = \{q_0, q_1, q_2, q_3, q_4\}$$

$$\Sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 0, \bullet\}$$

$$q_0 = q_0$$

$$F = \{q_4\}$$

esempi di transizioni:

$$\delta(q_0, 3{\bullet}1) = \delta(\delta(q_0, 3{\bullet}), 1) = \delta(\delta(\delta(q_0, 3), \bullet), 1) =$$

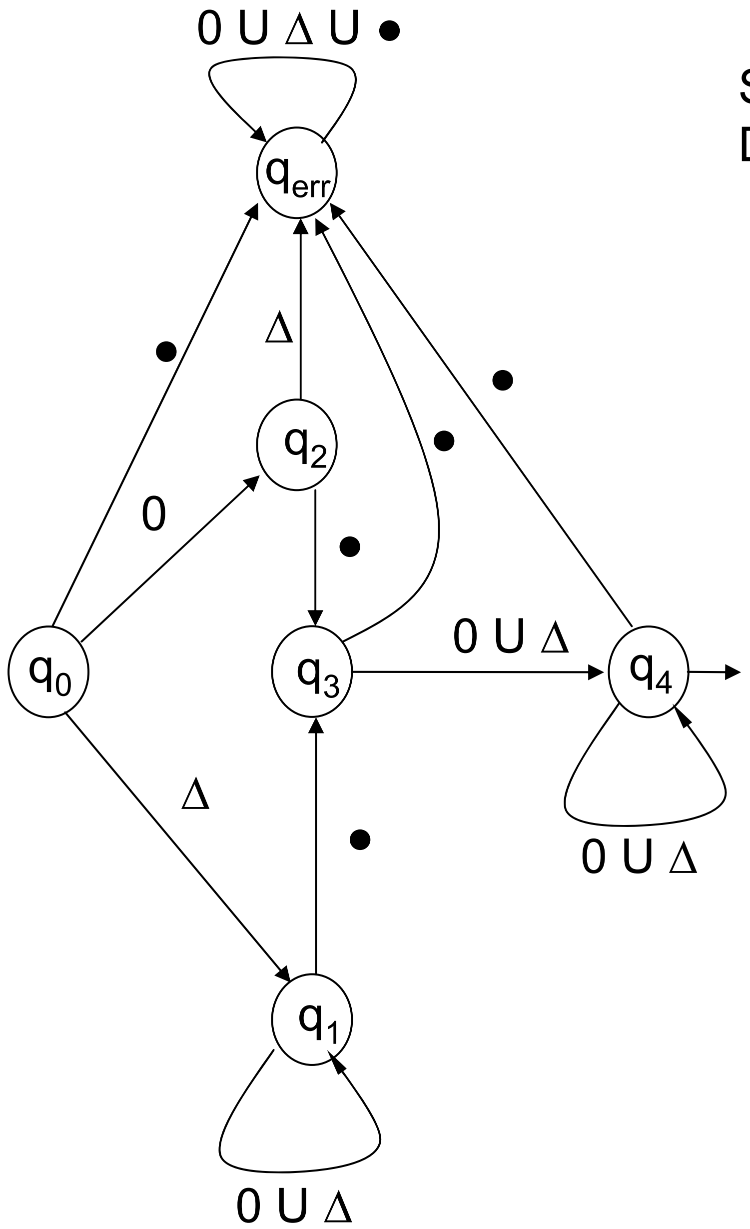
$$= \delta(\delta(q_1, \bullet), 1) = \delta(q_3, 1) = q_4$$

$q_4 \in F$ la stringa $3{\bullet}1$ è accettata

non accettate:

$\delta(q_0, 3{\bullet}) = q_3$ non è finale - $3{\bullet} \notin L$

$\delta(q_0, 02) = \delta(\delta(q_0, 0), 2) = \delta(q_2, 2)$ non è definita - $02 \notin L$



STATO DI ERRORE E COMPLETAMENTO DELL'AUTOMA

q_{err} = stato pozzo

La funzione di transizione può sempre essere completata con lo stato di errore, senza modificare il linguaggio accettato.

$\forall$ stato $q \in Q$ e $\forall$ car. $a \in \Sigma$

se $\delta(q,a)$ è indefinita

poni $\delta(q,a) = q_{err}$

$\forall$ car. $a \in \Sigma$ poni $\delta(q_{err},a) = q_{err}$

AUTOMA PULITO

Un automa potrebbe contenere delle parti inutili, che non danno contributo al linguaggio riconosciuto e vanno normalmente eliminate.

Uno stato q È RAGGIUNGIBILE da uno stato p se esiste un calcolo che porta l'automa da p a q

Uno stato È ACCESSIBILE è raggiungibile dallo stato iniziale.

Uno stato È POST-ACCESSIBILE da esso uno stato finale è raggiungibile.

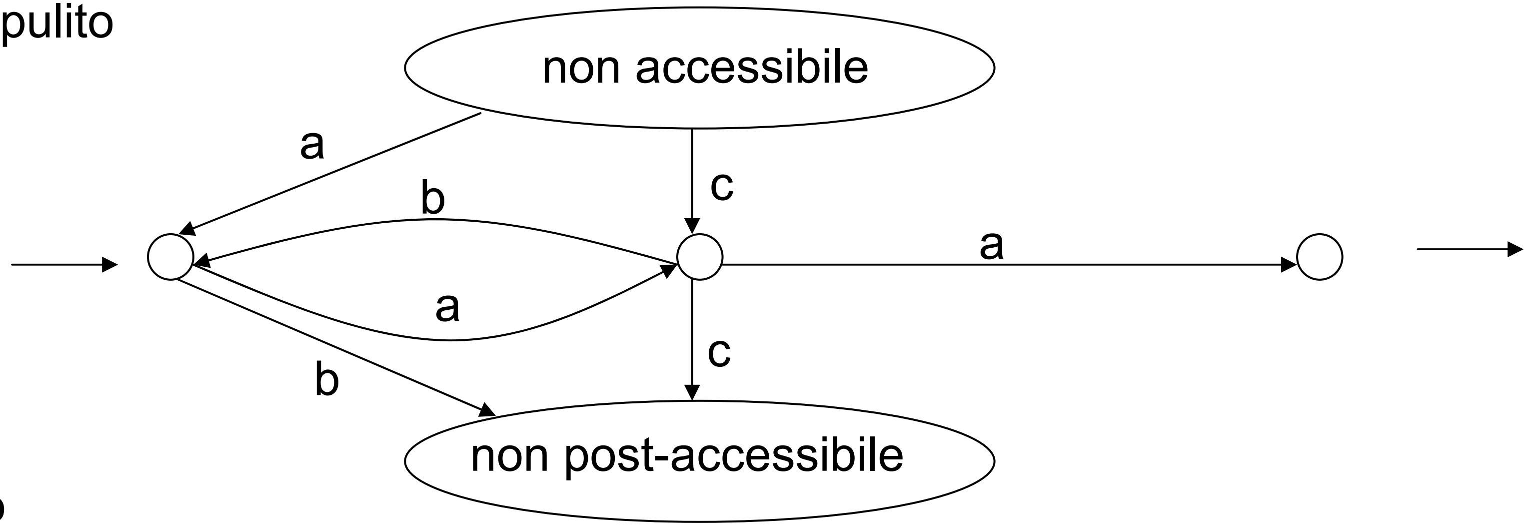
Uno stato è detto UTILE se è accessibile e post-accessibile (cioè giace su un cammino tra lo stato iniziale e finale).

Un automa è PULITO se ogni suo stato è utile.

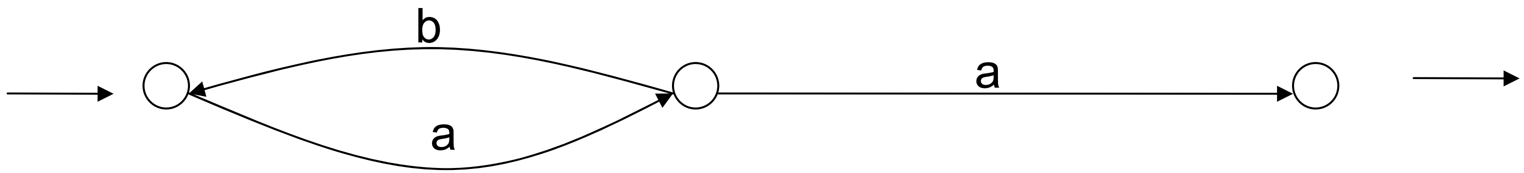
PROPRIETÀ – Ogni automa finito ammette un automa pulito equivalente.
Pulizia di una automa: individuati gli stati non utili, li si sopprime con tutti gli archi incidenti (sia entranti sia uscenti).

ESEMPIO – eliminazione degli stati inutili

non pulito



pulito



AUTOMA MINIMO

PROPRIETÀ – Per ogni linguaggio a stati finiti, il riconoscitore deterministico minimo rispetto al numero degli stati è unico (a meno di una ridenominazione degli stati).

STATI INDISTINGUIBILI - Lo stato p È INDISTINGUIBILE dallo stato q,
se e solo se, per ogni stringa x,
$\delta(p, x)$ e $\delta(q, x)$ sono entrambi finali, o entrambi non finali, oppure
scandendo x da p e q, non è possibile giungere a due stati, uno finale e l'altro no

L'INDISTINGUIBILITÀ è una relazione binaria simmetrica, transitiva e riflessiva

Quindi è una relazione di equivalenza

Due stati indistinguibili possono essere FUSI INSIEME, riducendo così il numero degli stati, senza alterare il comportamento dell'automa.

Per calcolare ralzione di indistinguibilità si passa dal suo complemento:

relazione di ***distinguibilità***

q_{err} è distinguibile da ogni altro stato p, perché certamente esiste una stringa x per cui risulta $\delta(p, x)$ appartenente a F, mentre per ogni stringa x, è $\delta(q_{err}, x)=q_{err}$

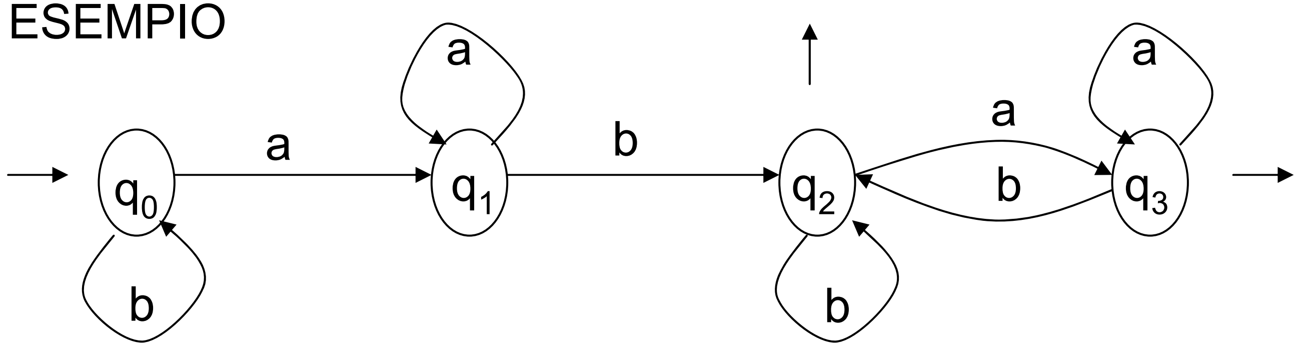
p È DISTINGUIBILE DA q se

1. p è finale e q no (o viceversa)
2. $\delta(p, a)$ è distinguibile da $\delta(q,a)$

IN PARTICOLARE - p È DISTINGUIBILE da q se gli insiemi delle etichette uscenti da p e q sono diversi (NB: non necessariamente disgiunti)

Infatti allora esiste *a* tale che $\delta(p,a)=p'\neq q_{err}$, mentre $\delta(p,a)=q_{err}$
E lo stato q_{err} e` distinguibile da tutti gli altri

ESEMPIO



<u>Tabella della relazione di indistinguibilità</u>: inizialmente distinguibili stati finali e non finali

q1				
q2	X	X		
q3	X	X		
	q0	q1	q2	q3

<u>(δ(q0,a),δ(q1,a))=(q1,q1)</u>
<u>(δ(q0,b),δ(q1,b))=(q0,q2)</u>
<u>q0 dist. q2 ➔ q0 dist. q1</u>
<u>(δ(q2,a),δ(q3,a))=(q3,q3)</u>
<u>(δ(q2,b),δ(q3,b))=(q2,q2)</u>
<u>q3=q3, q2=q2 ➔ q2 indist. q3</u>

q1	(1,1)(0,2)			
q2	X	X		
q3	X	X	(3,3)(2,2)	
	q0	q1	q2	q3

<u>Coppie indistinguibili</u>: q_2 e q_3

<u>Classi di equivalenza</u> della relazione di indistinguibilità: $[q_0]$, $[q_1]$, $[q_2, q_3]$.

q1	X			
q2	X	X		
q3	X	X		
	q0	q1	q2	q3

MINIMIZZAZIONE

L'automa minimo M' ha come stati le classi di equivalenza della relazione di indistinguibilità. Per definire la sua funzione di transizione basta dire che vi è in M' un arco tra due classi di equivalenza:

$$\boxed{[\ldots, p_r, \ldots] \xrightarrow{b} [\ldots, q_s, \ldots]}$$

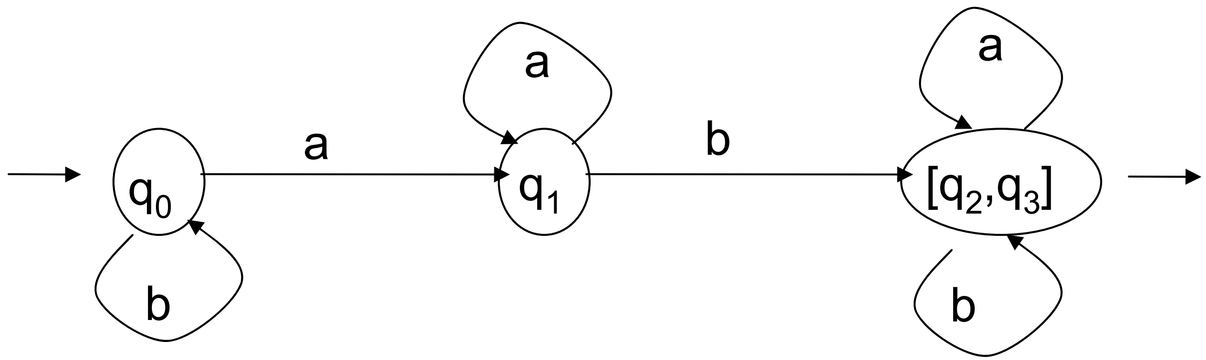
se e solo se in M vi è un arco:

$$\boxed{p_r \xrightarrow{b} q_s}$$

tra due stati, ordinatamente appartenenti alle due classi di equivalenza.
Si noti che lo stesso arco di M' potrebbe derivare da più archi di M.

ESEMPIO (continua)

Automa minimo dell'esempio precedente:



Come si comporta l'algoritmo di minimizzazione quando la funzione di transizione non è totale ? Supponiamo di modificare l'automa M cancellando l'autoanello $\delta(q_3, a) = q_3$. Ridefiniamo come $\delta(q_3, a) = q_{err}$. q_2 e q_3 risultano così distinguibili, perché $\delta(q_2, a) = q_3$ e $\delta(q_3, a) = q_{err}$ e q_3 è distinguibile da q_{err}.
M risulta dunque minimo.

Il metodo di minimizzazione dimostra la proprietà di esistenza e unicità dell'automa minimo equivalente a un automa deterministico dato.
La proprietà di esistenza e unicità dell'automa minimo non vale in generale per gli automi non deterministici.

Per verificare se due automi siano equivalenti si riducono, si minimizzano e si confrontano, per vedere se siano identici a meno di un cambiamento di nome degli stati.

DALL’AUTOMA ALLA GRAMMATICA

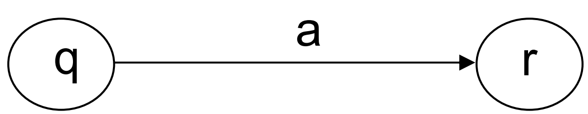
GLI AUTOMI A STATI FINITI E LE GRAMMATICHE UNILINEARI
(tipo 3 di Chomsky) trattano esattamente gli stessi linguaggi.

<u>Si mostra come costruire una grammatica lineare a destra equivalente a un automa.</u>

Passagio inverso (grammatica $\Rightarrow$ automa) richiede nondeterminismo che vedremo poi

La grammatica: simboli n.t. gli stati Q dell’automa,
assioma: lo stato iniziale.

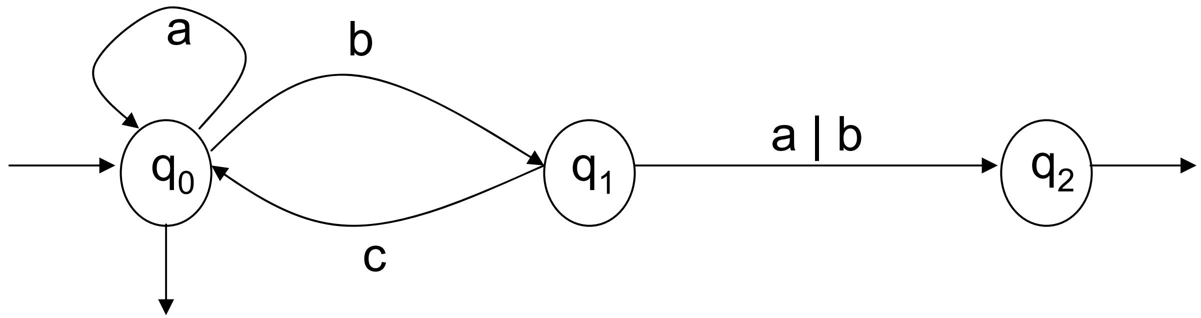
$$q \to ar$$
se $q \in F$
$$q \to ar \quad q \to \varepsilon$$



corrispondenza biunivoca: calcoli automa $\leftrightarrow$ derivazioni della grammatica

stringa x è accettata dall’automa se e solo se $q_0 \overset{+}{\Rightarrow} x$ (NB e` una derivazione)

ESEMPIO



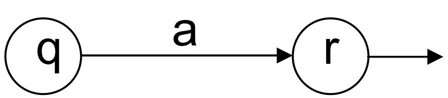
$$
\begin{aligned}
q_0 &\rightarrow aq_0 \mid bq_1 \mid \varepsilon \\
q_1 &\rightarrow cq_0 \mid aq_2 \mid bq_2 \\
q_2 &\rightarrow \varepsilon
\end{aligned}
$$

riconoscimento di *bca*

$$q_0 \Rightarrow bq_1 \Rightarrow bcq_0 \Rightarrow bcaq_0 \Rightarrow bca\varepsilon$$

<u>Forma non annullabile della grammatica</u>:

In questa versione della grammatica
a una mossa entrante in uno stato finale
corrispondono le due regole:



$$q \rightarrow ar \mid a$$

$$
\begin{aligned}
&Null = \{q_0, q_2\} \\
&q_0 \rightarrow aq_0 \mid bq_1 \mid a \mid \varepsilon \\
&q_1 \rightarrow cq_0 \mid aq_2 \mid bq_2 \mid a \mid b \mid c
\end{aligned}
$$