Trasparenze del corso di

Informatica Teorica

Parte Terza

Linguaggi Regolari

1

Linguaggi regolari

- sono i linguaggi generati da grammatiche di Chomsky di tipo 3
- vari elementi sintattici di base dei linguaggi di programmazione sono regolari (es. identificatori)
- godono di varie interessanti proprietà algebriche
- rapporti con espressioni regolari

2

Lungo viaggio nei linguaggi regolari

automi a stati finiti

relazioni tra automi e linguaggi regolari

"pumping lemma"

chiusura dei linguaggi regolari

espressioni regolari e linguaggi regolari

decidibilità e linguaggi regolari

teorema di Myhill-Nerode

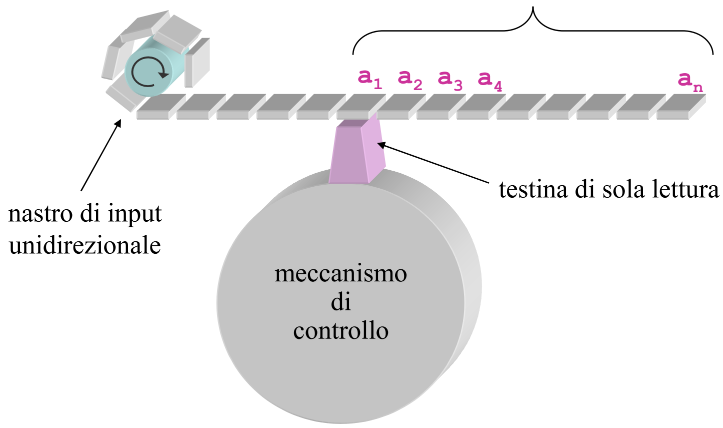
3

Automi a stati finiti (ASF)

- sono il tipo più semplice di macchina per riconoscere linguaggi
- dispositivi che leggono la stringa di input da un nastro unidirezionale e la elaborano usando una memoria limitata
- elaborazione un passo alla volta
- ad ogni passo: lettura di un carattere, spostamento della testina in avanti, aggiornamento dello stato

4

Schema di automa a stati finiti



5

Automi a stati finiti (ASF)

un automa a stati finiti è una quintupla

$$A = \langle \Sigma, K, \delta, q_0, F \rangle$$

$\Sigma = \{\sigma_1, \dots, \sigma_n\}$	alfabeto di input
$K = \{q_0, \dots, q_m\}$	insieme finito non vuoto di stati
$F \subseteq K$	insieme di stati finali
$q_0 \in K$	stato iniziale
$\delta : K \times \Sigma \rightarrow K$	funzione totale di transizione che determina lo stato successivo

può essere descritta tramite una *tabella di transizione* o un *diagramma di stato*

6

Esempio di ASF

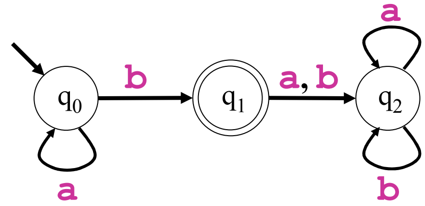
$A = \langle\Sigma, K, \delta, q_0, F\rangle$

$\Sigma = \{\mathbf{a}, \mathbf{b}\}$ $K = \{q_0, q_1, q_2\}$ $F = \{q_1\}$

tabella di transizione

δ	**a**	**b**
q_0	q_0	q_1
q_1	q_2	q_2
q_2	q_2	q_2

diagramma di stato



riconosce il linguaggio $\{a^n b \mid n \geq 0\}$

7

Linguaggio riconosciuto da un ASF

estendiamo la funzione di transizione alle stringhe:

$$\underline{\delta} : K \times \Sigma^* \to K$$

$$\begin{cases} \underline{\delta}(q,\varepsilon) = q \\ \underline{\delta}(q,x\mathbf{a}) = \delta(\underline{\delta}(q,x),\mathbf{a}) \end{cases}$$ con $x \in \Sigma^*$ ed $\mathbf{a} \in \Sigma$

linguaggio riconosciuto da un automa A:

$$L(A) = \{x \in \Sigma^* \mid \underline{\delta}(q_0,x) \in F\}$$

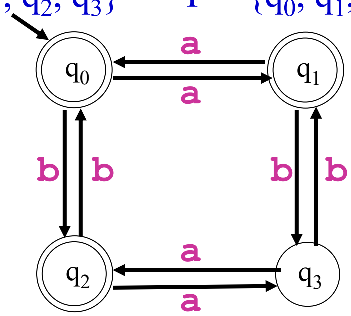
8

Esempio

automa che riconosce il linguaggio delle parole che contengono un numero pari (anche zero) di **a** o un numero pari (anche zero) di **b**

$\Sigma = \{$**a**, **b**$\}$ $K = \{q_0, q_1, q_2, q_3\}$ $F = \{q_0, q_1, q_2\}$

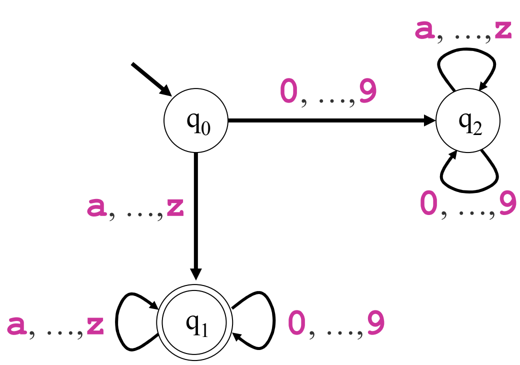
δ	**a**	**b**
q_0	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1



9

Esempio

automa che riconosce gli identificatori (stringhe alfanumeriche che cominciano con una lettera)



10

Automi a stati finiti non deterministici (ASFND)

un *automa a stati finiti non deterministico* è una quintupla

$$A = \langle \Sigma, K, \delta_N, q_0, F \rangle$$

$\Sigma = \{\sigma_1, \ldots, \sigma_n\}$	alfabeto di input
$K = \{q_0, \ldots, q_m\}$	insieme finito non vuoto di stati
$F \subseteq K$	insieme di stati finali
$q_0 \in K$	stato iniziale
$\delta_N : K \times \Sigma \rightarrow P(K)$	funzione totale di transizione che determina l'insieme (eventualmente vuoto) degli stati successivi

11

Linguaggio riconosciuto da un ASFND

estendiamo la funzione di transizione alle stringhe:

$$\underline{\delta}_N : K \times \Sigma^* \rightarrow P(K)$$

$$\begin{cases} \underline{\delta}_N(q,\varepsilon) = \{q\} \\ \underline{\delta}_N(q,x\mathbf{a}) = \bigcup_{p \in \underline{\delta}_N(q,x)} \delta_N(p,\mathbf{a}) \text{ con } x \in \Sigma^*, \mathbf{a} \in \Sigma, p \in K \end{cases}$$

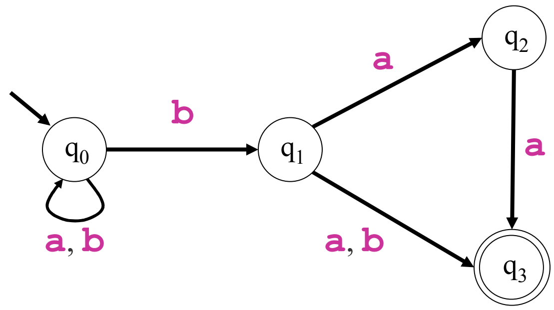
linguaggio riconosciuto da un automa A_N:

$$L(A_N) = \{x \in \Sigma^* \mid \underline{\delta}_N(q_0,x) \cap F \neq \emptyset\}$$

12

Esempio

ASFND che riconosce le stringhe che terminano con **bb** o **ba** o **baa**



13

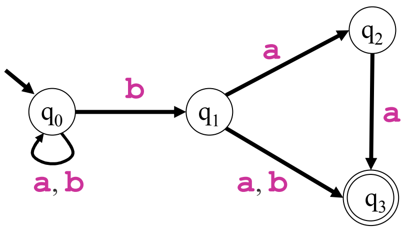
Osservazioni sugli ASFND

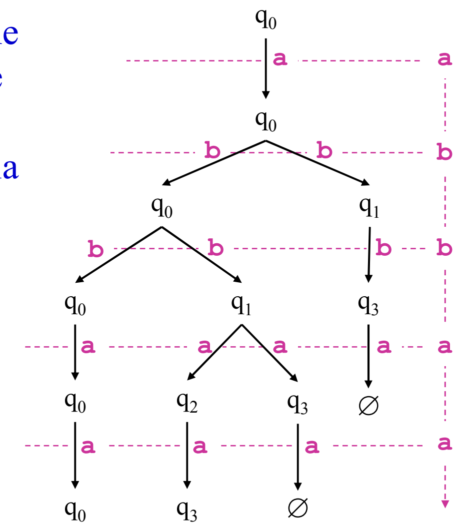
- attenzione a non confondere il non determinismo degli automi con altre accezioni del termine (per esempio aspetti probabilistici )
- il non determinismo degli automi consente di rappresentare una computazione come un albero nello spazio degli stati

14

Le transizioni determinano un albero

transizioni dell'ASFND che riconosce il linguaggio che termina con **bb** o **ba** o **baa** quando riceve in ingresso la stringa **abbaa**





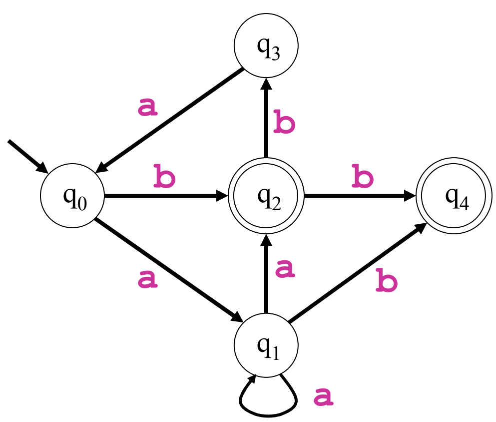
15

Esempio

ASFND che riconosce il linguaggio definito dall'espressione regolare:

$$((a^+a+b)ba)^*(b+bb+a^+(b+a+ab))$$

dove a^+ sta per aa^*



16

Automi a stati finiti
computazione

configurazione di un ASF:

coppia $<q,x>$

con $q \in K$ stato interno e $x \in \Sigma^*$ stringa da leggere in input

relazione di transizione di ASF:

relazione binaria sulle configurazioni $\vdash$

$$<q,x> \vdash <q',x'> \Leftrightarrow x=ax' \wedge \delta(q,a)=q'$$

configurazioni *iniziale*, *finale* e *accettante* di ASF:

$<q,x>$ è: iniziale se $q = q_0$

finale se $x = \varepsilon$

accettante se $x = \varepsilon$ e $q \in F$

17

Automi a stati finiti non deterministici
computazione

configurazione di un ASFND:

coppia $<Q,x>$

con $Q \subseteq K$ insieme di stati interni e $x \in \Sigma^*$ stringa da leggere in input

relazione di transizione di ASFND:

relazione binaria sulle configurazioni $\vdash$

$$<Q,x> \vdash <Q',x'> \Leftrightarrow x=ax' \wedge \bigcup_{p\in Q} \delta_N(p,a)=Q'$$

configurazioni *iniziale*, *finale* e *accettante* di ASFND:

$<q,x>$ è: iniziale se $Q = \{q_0\}$

finale se $x = \varepsilon$

accettante se $x = \varepsilon$ e $(Q \cap F) \neq \emptyset$

18

Automi a stati finiti deterministici (e non) computazione

computazione per ASF e ASFND:

chiusura transitiva e riflessiva di $\vdash$, indicata con $\vdash^*$

computazione accettante:

$c_0 \vdash^* c_n$ è accettante se c_0 è configurazione iniziale e c_n è configurazione accettante

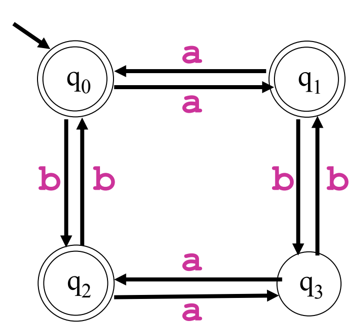
esempio:

l'ASF che riconosce le stringhe con un numero pari di **a** o di **b** accetta la stringa **abbaabb** tramite la computazione accettante

$<q_0,\mathbf{abbaabb}> \vdash^* <q_1,\varepsilon>$

19

Computazione accettante



$<q_0,\mathbf{abbaabb}> \vdash <q_1,\mathbf{bbaabb}>$

$\vdash <q_3,\mathbf{baabb}>$

$\vdash <q_1,\mathbf{aabb}>$

$\vdash <q_0,\mathbf{abb}>$

$\vdash <q_1,\mathbf{bb}>$

$\vdash <q_3,\mathbf{b}>$

$\vdash <q_1,\varepsilon>$

20

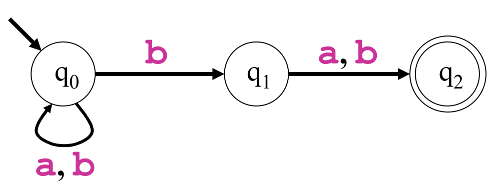
Automi a stati finiti non deterministici computazione

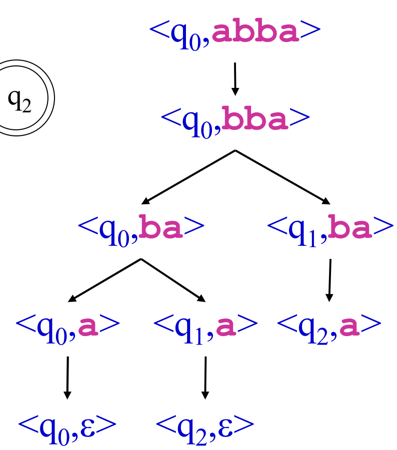
- un altro modo per definire una computazione per ASFND è il seguente:

1. definire la configurazione come per gli ASF
2. rappresentare l'albero delle configurazioni

21

ASFND che riconosce $(a+b)^*b(a+b)$





$$
\begin{aligned}
\langle\{q_0\},\mathbf{abba}\rangle & \vdash \langle\{q_0\},\mathbf{bba}\rangle \\
& \vdash \langle\{q_0,q_1\},\mathbf{ba}\rangle \\
& \vdash \langle\{q_0,q_1,q_2\},\mathbf{a}\rangle \\
& \vdash \langle\{q_0,q_2\},\varepsilon\rangle
\end{aligned}
$$

computazione

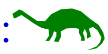
22

Relazione tra ASF e ASFND

sono computazionalmente più potenti gli ASF o gli ASFND?

teorema:

dato un ASF che riconosce il linguaggio L, esiste un ASFND che riconosce L, viceversa, dato un ASFND che riconosce il linguaggio L′, esiste un ASF che riconosce L′

dimostrazione: 

la simulazione di un ASF con un ASFND è banale, i due automi sostanzialmente coincidono

la simulazione di un ASFND con un ASF sfrutta la finitezza di P(K)

23

Dimostrazione (ASFND → ASF)

dato $A_N = <\Sigma, K, \delta_N, q_0, F>$

costruiamo $A' = <\Sigma', K', \delta', q'_0, F'>$

come segue:

- $\Sigma' = \Sigma$
- K' è in corrispondenza biunivoca con 2^K
  - i nomi degli stati di K' sono i sottoinsiemi in []
  - $K' = \{[q_{i1},\ldots,q_{ik}] \mid \{q_{i1},\ldots,q_{ik}\} \in 2^K\}$
- $q'_0 = [q_0]$
- $F' = \{[q_{i1},\ldots,q_{ik}] \mid \{q_{i1},\ldots,q_{ik}\} \in 2^K \wedge \{q_{i1},\ldots,q_{ik}\} \cap F \neq \emptyset\}$

24

Dimostrazione (ASFND → ASF)

$\delta'([q_{i1},\ldots,q_{ik}],\mathbf{a}) = [q_{j1},\ldots,q_{jk}]$
$\{q_{j1},\ldots,q_{jk}\} = \delta_N(q_{i1},\mathbf{a}) \cup \ldots \cup \delta_N(q_{ik},\mathbf{a})$

resta da mostrare che i due automi A_N ed A' si comportano allo stesso modo

resta cioè da mostrare che

$\underline{\delta}'([q_0],x) = q_i$ con $q_i \in F'$ e $q_i=[q_{j1},\ldots,q_{jh}]$

se e solo se

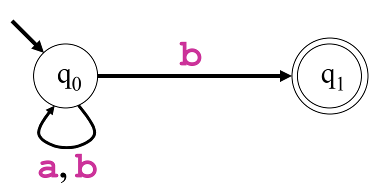
$\underline{\delta}_N(\{q_0\},x) = \{q_{j1},\ldots,q_{jh}\}$ con $\{q_{j1},\ldots,q_{jh}\} \cap F \neq \emptyset$

ciò è vero per costruzione

25

Esempio

costruire l'ASF corrispondente all'ASFND seguente:



che accetta le stringhe di $\{a,b\}^*$ che terminano con b

una strategia per la costruzione della funzione di transizione dell'ASF è quella di visitare gli stati a partire da $[q_0]$ introducendo nuovi stati (e relative transizioni) non appena ciò si renda necessario

26

Esempio (continua)

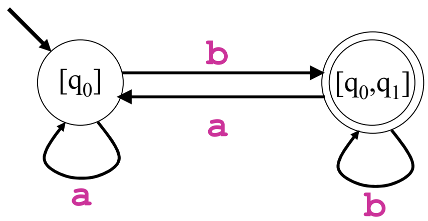
$\delta'([q_0],\mathtt{a}) = [\delta_N(q_0,\mathtt{a})] = [q_0]$

$\delta'([q_0],\mathtt{b}) = [\delta_N(q_0,\mathtt{b})] = [q_0,q_1]$ nuovo stato introdotto

$\delta'([q_0,q_1],\mathtt{a}) = [\delta_N(q_0,\mathtt{a})] \cup [\delta_N(q_1,\mathtt{a})] = [q_0,\varnothing] = [q_0]$

$\delta'([q_0,q_1],\mathtt{b}) = [\delta_N(q_0,\mathtt{b})] \cup [\delta_N(q_1,\mathtt{b})] = [q_0,q_1,\varnothing] = [q_0,q_1]$

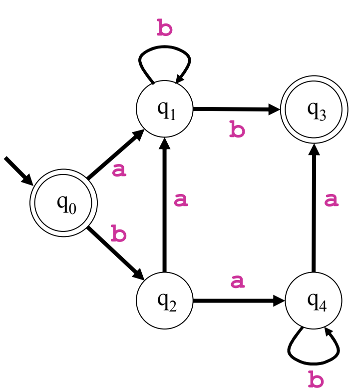
quindi $K' = \{[q_0], [q_0,q_1]\}$
dato che $F=\{q_1\}$
abbiamo $F'=\{[q_0,q_1]\}$

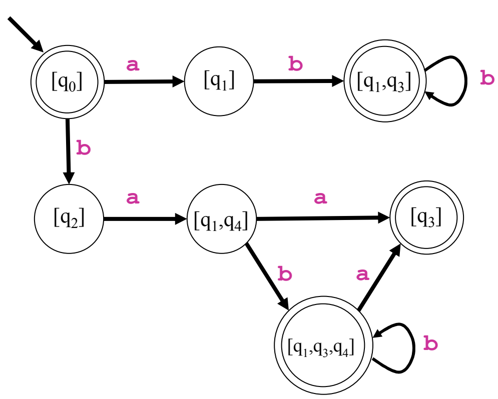


27

Esempio

costruzione di un ASF con 8 stati (compreso lo stato pozzo) a partire da un ASFND con 5 stati



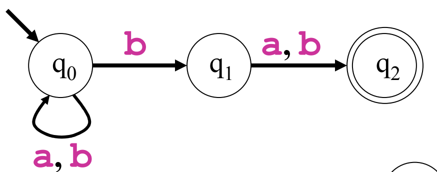


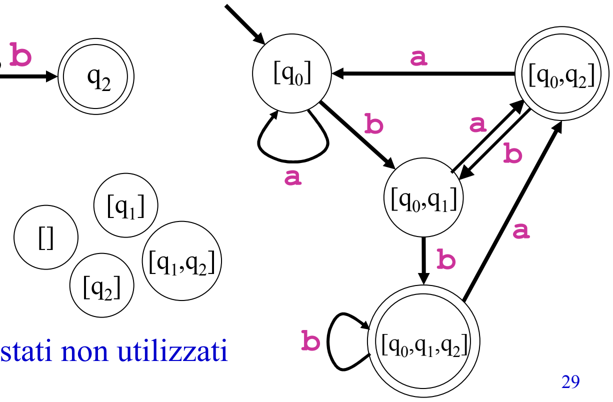
28

Osservazione

con il metodo di costruzione introdotto, l'ASF può (ma non è detto) avere un numero esponenziale di stati rispetto all'ASFND di partenza

esempio: automi che riconoscono $(a+b)^*b(a+b)$

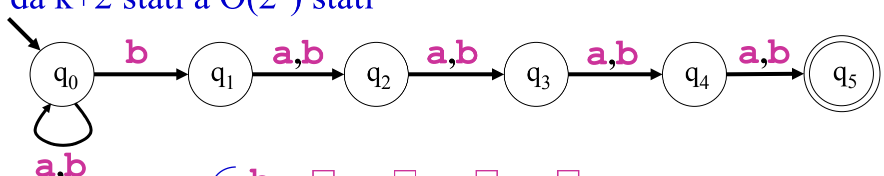


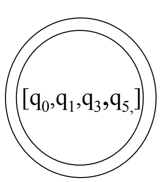


29

Osservazione

generalizzando a stringhe del tipo $(a+b)^*b(a+b)^k$, si passa da k+2 stati a $O(2^k)$ stati





b	□	□	□	□	raggiunge q_5
□	a	□	□	□	non raggiunge q_4
□	□	b	□	□	raggiunge q_3
□	□	□	a	□	non raggiunge q_2
□	□	□	□	b	raggiunge q_1
b	a	b	a	b	raggiunge q_0,q_1,q_3,q_5

30

Relazione tra ASFND e grammatiche regolari

teorema:

data una grammatica regolare $G=<V_T, V_N, P, S>$ esiste un ASFND $A_N=<\Sigma, K, \delta_N, q_0, F>$ che riconosce il linguaggio generato da G. Viceversa, dato un ASFND esiste una grammatica regolare che genera il linguaggio che esso riconosce

dimostrazione:

costruzione dell'automa a partire dalla grammatica e viceversa

31

Dimostrazione (grammatica → ASFND)

data $G=<V_T,V_N, P, S>$ costruiamo $A_N=<\Sigma, K, \delta_N, q_0, F>$ come segue:

$\Sigma = V_T$

$K=\{q_I | I \in V_N\} \cup \{q_F\}$ stato aggiuntivo

$q_0 = q_S$ corrispondenza tra stato iniziale e assioma

$F = \{q_F\} \cup \{q_B | B \rightarrow \varepsilon \in P\}$

$\delta_N(q_B, \mathbf{a}) = \{q_C | B \rightarrow \mathbf{a}C \in P\} \cup \{q_F \text{ se } B \rightarrow \mathbf{a} \in P\}$

Si noti che l'automa non è deterministico

32

Dimostrazione (grammatica → ASFND)

dimostriamo che esiste una derivazione $S \Rightarrow xZ$ di G se e solo se $q_Z \in \underline{\delta}_N(q_S,x)$ e se Z è finale allora $S \Rightarrow x$

per induzione:

se $|x|=1$ allora, per costruzione,

$S \Rightarrow xZ$ se e solo se $q_Z \in \delta_N(q_S,x)$

se $x=y\mathbf{a}$ e $|y|=n \geq 1$

allora se il risultato è valido per y è valido anche per x; infatti una derivazione $S \Rightarrow yZ \Rightarrow y\mathbf{a}Z'$ esiste se e solo se $q_Z \in \underline{\delta}_N(q_S,y)$ e $q_{Z'} \in \delta_N(q_Z,\mathbf{a})$ e ciò è vero se e solo se

$q_{Z'} \in \bigcup_{p \in \underline{\delta}_N(q_S,y)} \delta_N(p,\mathbf{a})$ cioè se $q_{Z'} \in \underline{\delta}_N(q_S,x)$

Si può mostrare che esiste una derivazione $S \Rightarrow x$ di G se e solo se $\underline{\delta}_N(q_S,x) \cap F \neq \emptyset$

33

Esempio (grammatica → ASFND)

il linguaggio $\{a(a+ba)^*a\}$ è generato dalla grammatica

$$G \begin{cases} S \rightarrow aB \\ B \rightarrow aB|bS|a \end{cases}$$

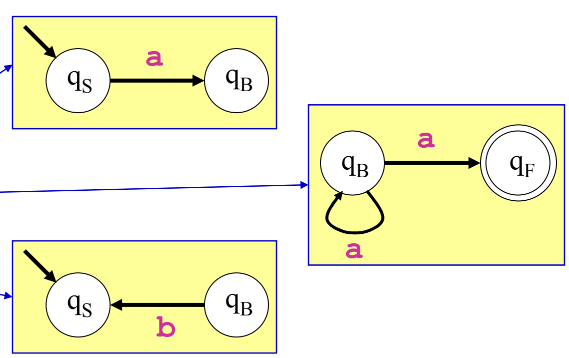
il corrispondente ASFND è definito da:

$K = \{q_S,q_B,q_F\}$

$\delta_N(q_S,\mathbf{a}) = \{q_B\}$

$\delta_N(q_B,\mathbf{a}) = \{q_B,q_F\}$

$\delta_N(q_B,\mathbf{b}) = \{q_S\}$

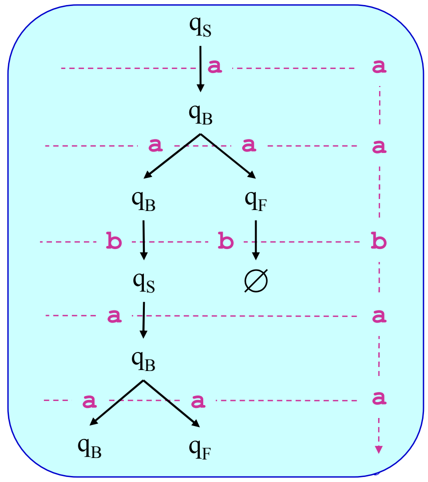


34

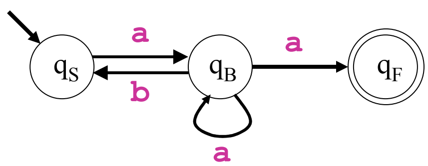
Esempio (computazione)

- generazione di **aabaa** tramite la grammatica G $\begin{cases} S \rightarrow aB \\ B \rightarrow aB|bS|a \end{cases}$

$S \Rightarrow aB$
$aB \Rightarrow aaB$
$aaB \Rightarrow aabS$
$aabS \Rightarrow aabaB$
$aabaB \Rightarrow aabaa$



- riconoscimento di **aabaa** tramite l'ASFND



Dimostrazione (ASF → grammatica)

dato $A=<\Sigma, K, \delta, q_0, F>$
costruiamo $G=<V_T, V_N, P, S>$ come segue:

$V_T = \Sigma$
$V_N = \{A_i|$ costruiamo un A_i per ogni $q_i \in K\}$
$S = A_0$
per ogni regola di transizione $\delta(q_i, a) = q_j$ costruiamo la produzione $A_i \rightarrow aA_j$ e se $q_j \in F$ anche $A_i \rightarrow a$
se $q_0 \in F$ (ε appartiene al linguaggio) inseriamo in V_N anche A_0' e per ogni $A_0 \rightarrow aA_i$ aggiungiamo $A_0' \rightarrow aA_i$ e $A_0' \rightarrow \varepsilon$
l'assioma diventa A_0'

36

Dimostrazione (ASF → grammatica)

Hp: $\underline{\delta}(q_i,x)=q_j$

Th: esiste la derivazione $A_i \Rightarrow xA_j$

inoltre, se $q_j \in F$, esiste la derivazione $A_i \Rightarrow x$

lo dimostriamo per induzione sulla lunghezza di x

[|x|=0]

è ovvio che:

- $\delta(q_i,\varepsilon)=q_i$ ed esiste $A_i \Rightarrow A_i$ (ricorda che "$\Rightarrow$" sta per "$\Rightarrow^*$")
- se $q_i=q_0 \in F$ (cioè se $\varepsilon \in L$) allora esiste sia $A_0' \Rightarrow A_0'$ che $A_0' \Rightarrow \varepsilon$

[passo base: |x|=1]

poniamo x=**a**

- per ipotesi $\delta(q_i,\mathbf{a})=q_j$
- allora per costruzione esiste $A_i \Rightarrow \mathbf{a}A_j$
- inoltre se $q_j \in F$, sempre per costruzione esiste $A_i \Rightarrow \mathbf{a}$

37

Dimostrazione (ASF → grammatica)

[passo induttivo: |x|>1]

Hp induttiva: se $\underline{\delta}(q_i,y)=q_k$ con $1 \leq |y| < |x|$

allora esiste $A_i \Rightarrow yA_k$

e se $q_k \in F$, esiste $A_i \Rightarrow y$

Th induttiva: esiste la derivazione $A_i \Rightarrow xA_j$

e se $q_j \in F$, esiste $A_i \Rightarrow x$

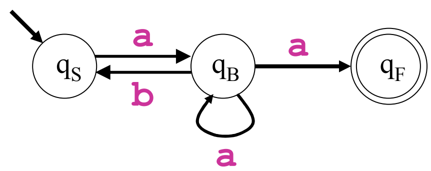
poniamo x=y**a** con $|y|=n \geq 1$

- $\underline{\delta}(q_i,y\mathbf{a})=\delta(\underline{\delta}(q_i,y),\mathbf{a})=q_j$
- per ipotesi induttiva esiste $A_i \Rightarrow yA_k$
- per costruzione, poiché $\delta(q_k,\mathbf{a})=q_j$ esiste $A_k \Rightarrow \mathbf{a}A_j$
- per costruzione se $q_j \in F$ esiste anche $A_k \Rightarrow \mathbf{a}$
- quindi $A_i \Rightarrow xA_j$ ed, eventualmente, $A_i \Rightarrow x$

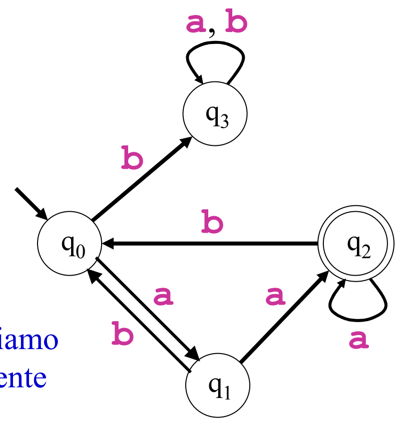
38

Esempio (ASFND → grammatica)

dato l'ASFND che riconosce il linguaggio $\{a(a+ba)^*a\}$ trovare la grammatica corrispondente



ASFND



per prima cosa otteniamo l'ASF corrispondente

Esempio (continua)

traduciamo le transizioni in produzioni

$A_0 \rightarrow \mathtt{a}A_1$
$A_1 \rightarrow \mathtt{b}A_0$
$A_1 \rightarrow \mathtt{a}A_2$
$A_1 \rightarrow \mathtt{a}$
$A_2 \rightarrow \mathtt{a}A_2$
$A_2 \rightarrow \mathtt{a}$
$A_2 \rightarrow \mathtt{b}A_0$

$$\left.\begin{array}{l} A_0 \rightarrow \mathtt{b}A_3 \\ A_3 \rightarrow \mathtt{b}A_3 \\ A_3 \rightarrow \mathtt{a}A_3 \end{array}\right\}$$ non sfruttabili per produrre stringhe

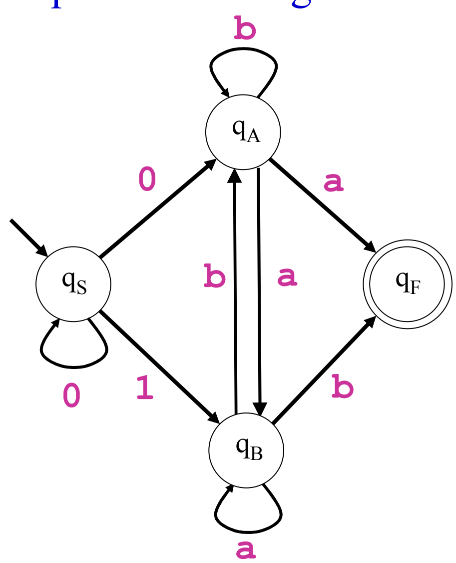
Esempio

trovare l'ASFND corrispondente alla grammatica

$S \rightarrow$ `0`A | `1`B | `0`S

$A \rightarrow$ `a`B | `b`A | `a`

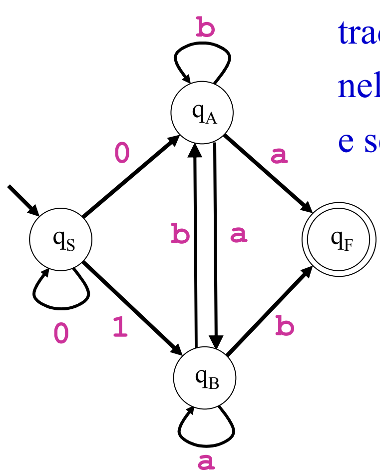
$B \rightarrow$ `b`A | `a`B | `b`



41

Esempio (continua)

si può passare direttamente dall'ASFND alla grammatica (senza passare per l'ASF)



traduciamo $\delta_N(q_i,$`a`$) = \{q_{j1},\ldots,q_{jh}\}$

nelle produzioni $A_i \rightarrow$ `a`A_{j1} | … | `a`A_{jh}

e se $A_{jk} \in F$ aggiungiamo $A_i \rightarrow$ `a`

$S \rightarrow$ `0`A | `1`B | `0`S

$A \rightarrow$ `a`B | `b`A | `a`

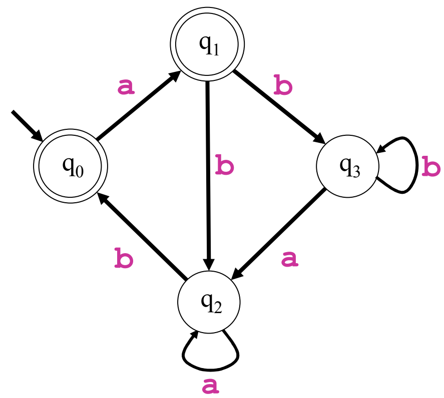
$B \rightarrow$ `b`A | `a`B | `b`

42

Esempio

ASFND che riconosce

$((ab(b^*a+\varepsilon)a^*b)^{+}+\varepsilon)\ \ (ab(b^*a+\varepsilon)a^*b+a)+\varepsilon$



$A_0' \to \varepsilon \mid aA_1 \mid a$

$A_0 \to aA_1 \mid a$

$A_1 \to bA_3 \mid bA_2$

$A_2 \to aA_2 \mid bA_0 \mid b$

$A_3 \to bA_3 \mid aA_2$

grammatica

43