Esercizi di Informatica Teorica

Automi a stati finiti

1

Automa a stati finiti (ASF)

richiami

automa a stati finiti ASF $= <\Sigma, K, \delta, q_0, F>$ dove

- $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ è un *alfabeto* (finito) di input
- $K = \{q_0, q_1, \ldots, q_m\}$ è un insieme (finito e non vuoto) di *stati*
- q_0 è lo *stato iniziale*
- $F \subseteq K$ è l'insieme degli *stati finali*
- $\delta : K \times \Sigma \rightarrow K$ è la *funzione* (totale) *di transizione*
  si può rappresentare graficamente tramite una *tabella di transizione* o un *diagramma di stato*

2

Linguaggio riconosciuto da un ASF

richiami

si definisce ricorsivamente la *funzione di transizione estesa alle stringhe* $\underline{\delta} : K \times \Sigma^* \rightarrow K$ nel seguente modo:

- $\underline{\delta}(q, \varepsilon) = q$

- se $a \in \Sigma$ e $w \in \Sigma^*$
  $\underline{\delta}(q, aw) = \underline{\delta}(\delta(q, a), w)$

linguaggio riconosciuto da A : $L = \{w \in \Sigma^* : \underline{\delta}(q_0, w) \in F\}$

3

Configurazioni e computazioni

richiami

configurazione (istantanea) di un ASF: $<q, w>$ dove
- q è lo *stato corrente* dell'ASF
- w è la porzione di stringa che l'ASF deve ancora leggere

transizione tra configurazioni: $<q, w> \vdash <q', w'> \Leftrightarrow$
- $w = aw'$ $\quad a \in \Sigma$
- $\delta(q, a) = q'$

una configurazione $<q, w>$ è

iniziale	se $q = q_0$
finale	se $w = \varepsilon$
accettante	se $w = \varepsilon$ e $q \in F$

computazione: $\vdash^*$ è la chiusura transitiva e riflessiva di $\vdash$

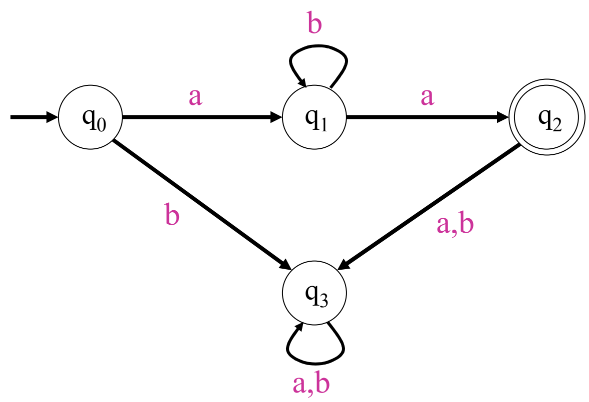
computazione accettante: $c_0 \vdash^* c_n$ dove c_0 è iniziale e c_n è accettante

4

Automi a stati finiti deterministici

esercizio 1

descrivere il linguaggio riconosciuto dal seguente ASF e trovare la corrispondente espressione regolare

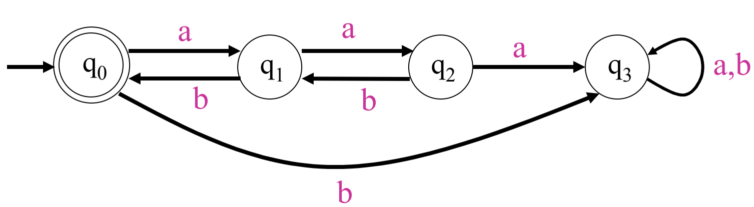


5

Automi a stati finiti deterministici

esercizio 2

si consideri il seguente AFS:



2.a mostrare le computazioni sulle stringhe “aaab” e “abaabb”

2.b dire qual’è il linguaggio riconosciuto dall’automa

2.c descrivere il linguaggio attraverso una espressione regolare

6

Automi a stati finiti deterministici

esercizio 3

costruire un AFS che riconosce il linguaggio dei numeri naturali pari in base 3, compresa la stringa vuota; si modifichi poi l'automa in modo che non accetti la stringa vuota.

esercizio 4

scrivere la tabella di transizione per l'automa dell'esercizio 3

7

Automi a stati finiti deterministici

esercizio 5

costruire gli AFS che riconoscono i seguenti linguaggi

5.a $L_1=\{w \in \{a,b\}^* : w$ non contiene mai tre 'b' consecutive$\}$

5.b $L_2=\{w \in \{a,b\}^* : w$ contiene tre 'b' consecutive$\}$

5.c $L_3=\{w \in \{a,b\}^* : w$ contiene almeno tre 'b'$\}$

8

Automi non deterministici

richiami

automa a stati finiti non deterministico ASFND=< Σ, K , δ_N, q_0, F >
dove:

- $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ è un *alfabeto* (finito) di input
- $K = \{q_0, q_1, \ldots, q_m\}$ è un insieme (finito e non vuoto) di *stati*
- q_0 è lo *stato iniziale*
- $F \subseteq K$ è l'insieme degli *stati finali*
- $\delta_N : K \times \Sigma \rightarrow \mathcal{P}(K)$ è la *funzione (totale) di transizione*

9

Linguaggio riconosciuto da un ASFND

richiami

si definisce ricorsivamente la *funzione di transizione estesa alle stringhe* $\underline{\delta}_N : K \times \Sigma^* \rightarrow \mathcal{P}(K)$ nel seguente modo:

- $\underline{\delta}_N(q, \varepsilon) = \{q\}$
- se $a \in \Sigma$ e $w \in \Sigma^*$

$$\underline{\delta}_N(q, aw) = \bigcup_{p \in \delta_N(q,a)} \underline{\delta}_N(p, w)$$

linguaggio riconosciuto da A : $L = \{w \in \Sigma^* : \underline{\delta}_N(q_0, w) \cap F \neq \emptyset\}$

10

Configurazioni e computazioni di un ASFND

richiami

configurazione (istantanea) di un ASFND: $<Q, w>$ dove

- $Q \subseteq K$ è l'insieme degli *stati correnti* dell'ASFND
- w è la porzione di stringa che l'ASFND deve ancora leggere

transizione tra configurazioni: $<Q, w> \vdash <Q', w'> \quad \Leftrightarrow$

- $w = a\, w' \quad a \in \Sigma$
- $Q' = \bigcup_{q \in Q} \delta_N(q, a)$

una configurazione $<Q, w>$ è

- *iniziale* se $Q = \{q_0\}$
- *finale* se $w = \varepsilon$
- *accettante* se $w = \varepsilon$ e $Q \cap F \neq \emptyset$

computazione: $\vdash^*$ è la chiusura transitiva e riflessiva di $\vdash$

computazione accettante: $c_0 \vdash^* c_n$ dove c_0 è iniziale e c_n è accettante

11

Automi non deterministici

esercizio 6

costruire un ASFND che riconosce il linguaggio descritto dall'espressione regolare $(ab+aba)^*$

esercizio 7

costruire un ASF che riconosce lo stesso linguaggio dell'esercizio 6

12

Automi non deterministici

esercizio 8

costruire un ASFND che riconosce il linguaggio L = (ab)*a(ab)*

esercizio 9

mostrare la computazione sulla stringa "ababaab" dell'automa soluzione dell'esercizio 8

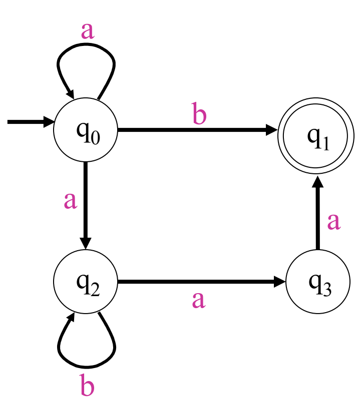
13

Automi non deterministici

esercizio 10

quali stringhe tra "aabb", "ab", "abbaa" ed "aabbaa" sono riconosciute dal seguente ASFND?

quale linguaggio riconosce?



14

Automi non deterministici

esercizio 11

costruire un ASFND che riconosce il linguaggio delle stringhe su {a,b} con un numero dispari di ‘a’ e un numero pari di ‘b’

esercizio 12

modificare l’automa in modo che riconosca le stringhe con un numero dispari di ‘a’ o un numero pari di ‘b’

15

Esercizi sugli automi

esercizio 13

costruire degli automi a stati finiti (deterministici o non deterministici) che riconoscono i seguenti linguaggi:

13.a L = (ab*a)*
13.b L = (ab*a*b)*
13.c L = a*b*(aa + bb)
13.d L = a (bc)*a
13.e stringhe su {a,b} terminanti con “baa” o con “abb”
13.f stringhe su {a,b} terminanti con un numero dispari di “a”

16

Algoritmo: ASFND → ASF

richiami

input: un ASFND $<\Sigma, K, \delta_N, q_0, F>$
ouput: un ASF $<\Sigma', K', \delta', q'_0, F'>$

costruzione:

- $\Sigma' = \Sigma$
- K' = contiene un superstato $[q_i \ldots q_j]$ per ciascun elemento $\{q_i,..,q_j\}$ di $P(K)$
- $q'_0 = [q_0]$
- $F' \subseteq K'$ è l'insieme dei superstati che contengono almeno uno stato di F
- $\delta'([q_i \ldots q_j], a) = [q_h \ldots q_k]$ dove $\{q_h \ldots q_k\} = \delta_N(q_i,a) \cup \ldots \cup \delta_N(q_j,a)$

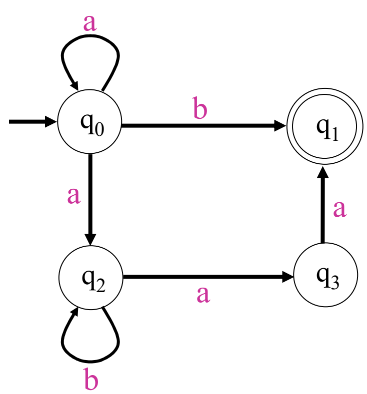
semplificazione: per costruire K' si considerano solo superstati raggiungibili a partire dal superstato $[q_0]$

17

ASFND → ASF

esercizio 14

costruire un ASF che riconosce lo stesso linguaggio del seguente ASFND:



δ_N	a	b
q_0	$\{q_0,q_2\}$	$\{q_1\}$
q_1	$\varnothing$	$\varnothing$
q_2	$\{q_3\}$	$\{q_2\}$
q_3	$\{q_1\}$	$\varnothing$

18

Esercizi su ASFND → ASF

esercizio 15

utilizzando l'algoritmo ASFND → ASF, costruire gli ASF che riconoscono gli stessi linguaggi dei seguenti automi:

15.a AFSND dell'esercizio 6

15.b AFSND dell'esercizio 8

19

Algoritmo: ASFND → grammatica regolare

richiami

input: un ASFND $<\Sigma, K, \delta_N, q_0, F>$ (o un ASF $<\Sigma, K, \delta, q_0, F>$)
ouput: una grammatica regolare $<V_T, V_N, P, S>$

costruzione:

- $V_T = \Sigma$
- V_N contiene un non terminale A_i per ogni stato $q_i \in K$; se q_0 è uno stato finale, si aggiunge un ulteriore non terminale A'_0
- $S = A_0$ se q_0 non è uno stato finale, altrimenti $S = A'_0$
- P contiene le seguenti produzioni $\forall q_k \in \delta_N(q_i,a)$ (o se $\delta(q_i,a) = q_k$):
  - $A_i \rightarrow aA_k$
  - $A_i \rightarrow a$ se q_k è uno stato finale

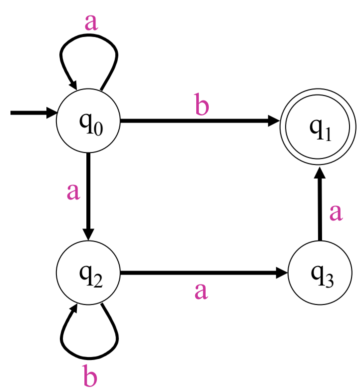
inoltre, se q_0 è uno stato finale, P contiene le seguenti produzioni:
  - $A'_0 \rightarrow \varepsilon$
  - $A'_0 \rightarrow aA_k$ $\forall A_0 \rightarrow aA_k$ $A'_0 \rightarrow a$ $\forall A_0 \rightarrow a$

20

ASFND → grammatica regolare

esercizio 16

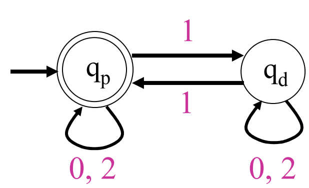
determinare una grammatica regolare equivalente al seguente ASFND



ASFND → grammatica regolare

esercizio 17

determinare una grammatica regolare equivalente al seguente ASFND



Algoritmo: grammatica regolare → ASFND

richiami

input: una grammatica regolare $<V_T, V_N, P, S>$
ouput: un ASFND $<\Sigma, K, \delta_N, q_0, F>$

costruzione:

- $\Sigma = V_T$
- K contiene uno stato q_A per ogni $A \in V_N$, più uno stato finale q_F
- $q_0 = q_S$
- F contiene q_F, più uno stato q_B per ogni ε-produzione $B \to \varepsilon$
- $\delta_N(q_B,a)$ contiene:
  - q_C se $B \to aC$
  - q_F se $B \to a$

23

Grammatica regolare → ASFND

esercizio 18

dimostrare che per ogni linguaggio regolare L che non contiene la stringa vuota ε esiste un ASFND con un solo stato finale che riconosce L

24

Grammatica regolare → ASFND

esercizio 19

19.a determinare un ASFND equivalente alla seguente grammatica regolare:

$V_T = \{a, b, c\}$
$V_N = \{S, A, C\}$, dove S è l'assioma

produzioni (1) $S \to aA$ (2) $S \to bC$
(3) $A \to aA$ (4) $A \to bC$
(5) $C \to cC$ (6) $C \to c$

19.b descrivere il linguaggio riconosciuto dall'ASFND

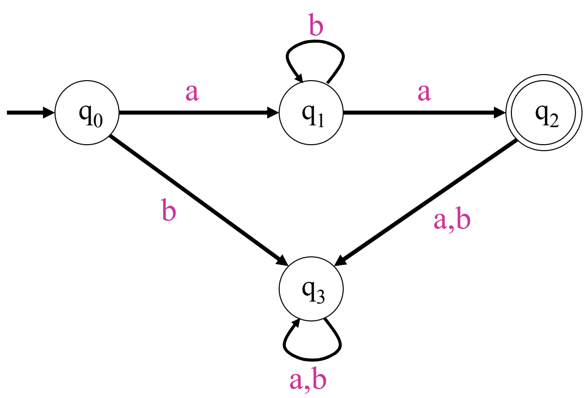
25

Soluzioni

soluzione esercizio 1

$L=\{ab^na : n \geq 0\}$

l'espressione regolare corrispondente è **ab*a**



26

Soluzioni

soluzione esercizio 2

2.a computazioni sulle stringhe “aaab” e “abaabb”

<q_0,aaab> $\vdash$ <q_1,aab> $\vdash$ <q_2,ab> $\vdash$ <q_3,b> $\vdash$ <q_3,ε> (non accettante)

<q_0,abaabb> $\vdash$ <q_1,baabb> $\vdash$ <q_0,aabb> $\vdash$ <q_1,abb> $\vdash$ <q_2,bb> $\vdash$ <q_1,b> $\vdash$ <q_0,ε> (accettante)

2.b linguaggio riconosciuto dall’automa: stringhe su {a,b} tali che:

- numero di ‘a’ = numero di ‘b’
- sottosequenze massimali di sole ‘a’ o di sole ‘b’ di lunghezza al più 2
- iniziano per ‘a’ e finiscono per ‘b’
- in ogni punto, la sequenza è sbilanciata di al più due ‘a’

più la stringa vuota

2.c espressione regolare: (a(ab)*b)*

27

Soluzioni

soluzione esercizio 3

- AFS che riconosce il linguaggio dei numeri naturali in base 3, compresa la stringa vuota

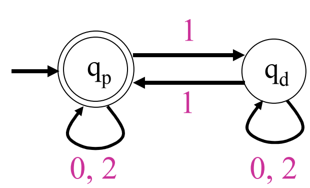
$\Sigma = \{0, 1, 2\}$

$K = \{q_p, q_p\}$

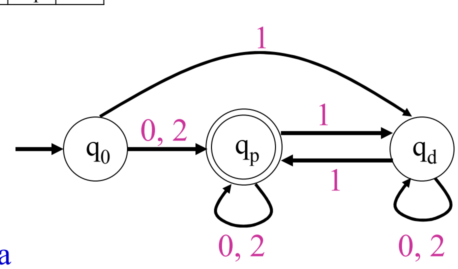
$F = \{q_p\}$

$q_0 = q_p$

δ	0	1	2
q_p	q_p	q_d	q_p
q_d	q_d	q_p	q_d



- AFS che riconosce il linguaggio dei numeri naturali pari in base 3, esclusa la stringa vuota

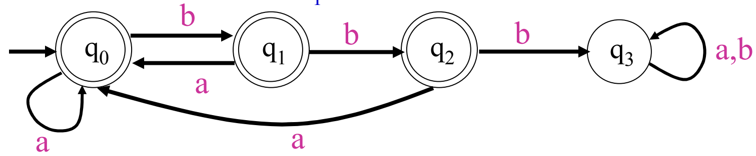


28

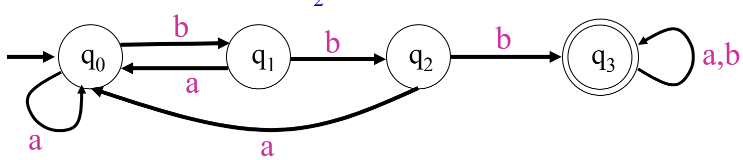
Soluzioni

soluzione esercizio 5

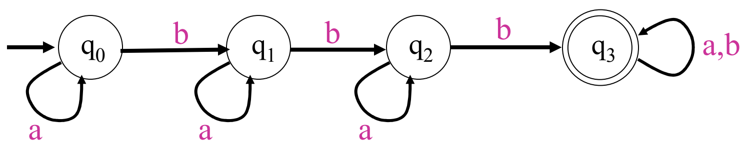
5.a ASF che riconosce L_1



5.b ASF che riconosce L_2



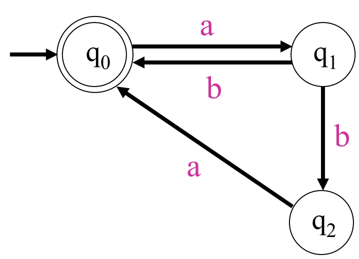
5.c ASF che riconosce L_3



Soluzioni

soluzione esercizio 6

δ_N	a	b
q_0	$\{q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_0,q_2\}$
q_2	$\{q_0\}$	$\varnothing$

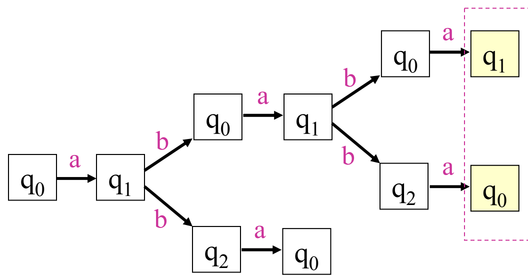


Soluzioni

esempio di computazione sulla stringa “ababa”

$<\{q_0\},ababa> \vdash <\{q_1\},baba> \vdash <\{q_0,q_2\}, aba> \vdash <\{q_0,q_1\}, ba> \vdash$
$<\{q_0,q_2\}, a> \vdash <\{q_0,q_1\}, \varepsilon>$

albero delle transizioni per la stringa “ababa”



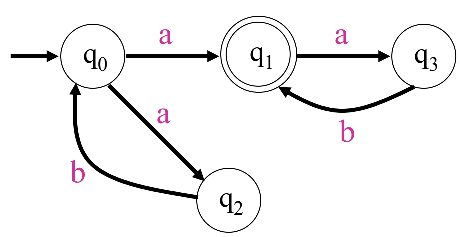
stati al termine della computazione

31

Soluzioni

soluzione esercizio 8

δ_N	a	b
q_0	$\{q_1,q_2\}$	$\emptyset$
q_1	$\{q_3\}$	$\emptyset$
q_2	$\emptyset$	$\{q_0\}$
q_3	$\emptyset$	$\{q_1\}$



soluzione esercizio 10

$L = (a^*b+a^*ab^*aa) = a^*(b+ab^*aa)$

32

Soluzioni

soluzione esercizio 11

logica costruttiva:

- si usano quattro stati con i seguenti significati q_0 = pari 'a' e pari 'b', q_1 = dispari 'a' e pari 'b', q_2 = dispari 'a' e dispari 'b', q_3 = pari 'a' e dispari 'b';
- si costruisce la funzione di transizione, osservando che da ciascuno stato si può passare direttamente solo a stati adiacenti;
- si decidono gli stati accettanti sulla base della classificazione fatta e delle stringhe che si vogliono riconoscere

33

Soluzioni

soluzione esercizio 14

- costruzione incrementale della funzione di transizione δ dell'ASF

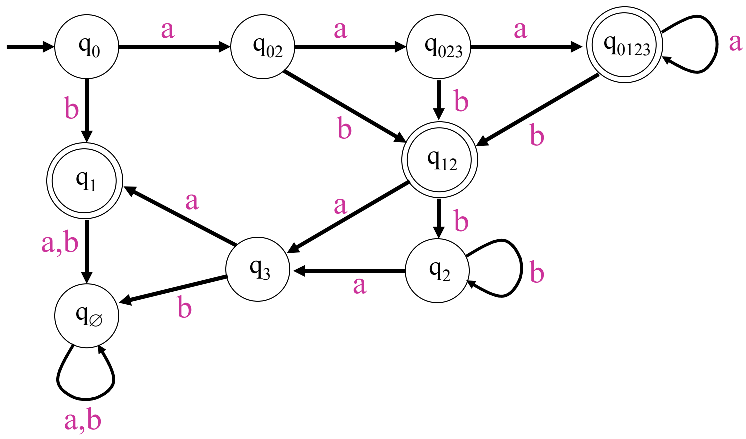
δ_N	a	b
q_0	$\{q_0,q_2\}$	$\{q_1\}$
q_1	$\varnothing$	$\varnothing$
q_2	$\{q_3\}$	$\{q_2\}$
q_3	$\{q_1\}$	$\varnothing$

- $\delta([q_0],a) = [q_0q_2]$ $\quad\delta([q_0],b) = [q_1]$
- $\delta([q_0q_2],a) = [q_0q_2q_3]$ $\quad\delta([q_0q_2],b) = [q_1q_2]$
- $\delta([q_1],a) = []$ $\quad\delta([q_1],b) = []$
- $\delta([q_0q_2q_3],a) = [q_0q_1q_2q_3]$ $\quad\delta([q_0q_2q_3],b) = [q_1q_2]$
- $\delta([q_1q_2],a) = [q_3]$ $\quad\delta([q_1q_2],b) = [q_2]$
- $\delta([q_0q_1q_2q_3],a) = [q_0q_1q_2q_3]$ $\quad\delta([q_0q_1q_2q_3],b) = [q_1q_2]$
- $\delta([q_3],a) = [q_1]$ $\quad\delta([q_3],b) = []$
- $\delta([q_2],a) = [q_3]$ $\quad\delta([q_2],b) = [q_2]$

34

Soluzioni

• grafo dell'ASF con funzione di transizione δ (per semplicità si scrive $[q_{i} \ldots q_{j}] = [q_{i \ldots j}]$)



35

Soluzioni

soluzione esercizio 16

$V_T=\{a,b\}$ $V_N=\{A_0, A_1, A_2, A_3\}$ $S = A_0$

insieme P delle produzioni

• produzioni per A_0: $A_0 \to aA_0$ $A_0 \to aA_2$ $A_0 \to bA_1$ $A_0 \to b$

• produzioni per A_1: nessuna

• produzioni per A_2: $A_2 \to aA_3$ $A_2 \to bA_2$

• produzioni per A_3: $A_3 \to aA_1$ $A_3 \to a$

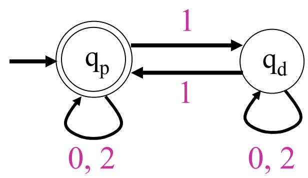
osservazione: poiché A_1 non ha produzioni, le produzioni $A_0 \to bA_1$ e $A_3 \to aA_1$ diventano inutili per la grammatica, e possono quindi essere tolte

36

Soluzioni

soluzione esercizio 17

$V_T=\{0,1,2\}$ $V_N=\{A_p, A_d, A'\}$ $S = A'$



insieme P delle produzioni

- produzioni per A_p: $A_p \to 0A_p$ $A_p \to 2A_p$ $A_p \to 1A_d$ $A_p \to 0$ $A_p \to 2$
- produzioni per A_d: $A_d \to 0A_d$ $A_d \to 2A_d$ $A_d \to 1A_p$ $A_d \to 1$
- produzioni per A’: $A' \to 0A_p$ $A' \to 2A_p$ $A' \to 1A_d$ $A' \to 0$ $A' \to 2$

$A' \to \varepsilon$

soluzione esercizio 18

- se L non contiene ε allora esiste per L una grammatica regolare G senza ε-produzioni
- applicando l’algoritmo che da G calcola un AFSND, ricaviamo un automa con un solo stato finale
- tale automa, essendo equivalente a G, riconosce L

37

Soluzioni

soluzione esercizio 19

19.a l’ASFND è definito come segue

$\Sigma = \{a,b,c\}$

$K=\{q_S, q_A, q_C, q_F\}$, q_S iniziale

$F = \{q_F\}$

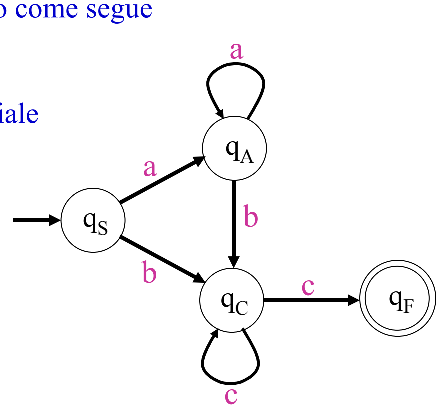
$\delta_N(q_S, a) = \{q_A\}$

$\delta_N(q_S, b) = \{q_C\}$

$\delta_N(q_A, a) = \{q_A\}$

$\delta_N(q_A, b) = \{q_C\}$

$\delta_N(q_C, c) = \{q_C, q_F\}$



19.b il linguaggio riconosciuto è $L = a^*bc^*c$

38