Corso di laurea in Informatica Applicata
Fondamenti di Programmazione
Prima verifica intermedia

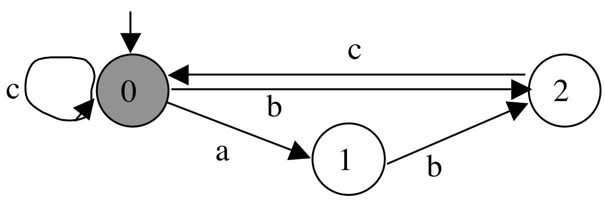
6/11/2002

Esercizio 1

Sia L(A) il linguaggio sull'alfabeto {a,b,c} che riconosce le sequenze (anche vuote) tali che il simbolo 'a' è sempre seguito dal simbolo 'b' e il simbolo 'b' e sempre seguito dal simbolo 'c'. Si definiscano l'automa deterministico e la grammatica regolare per L(A).

Soluzione

Automa per L(A)



Grammatica regolare

G=<{a,b,c}, {S},S,{S::=c*| (bc)* | (abc)*}>

Esercizio 2

Si consideri l'automa nondeterministico descritto dalla tabella di transizione sotto riportata:

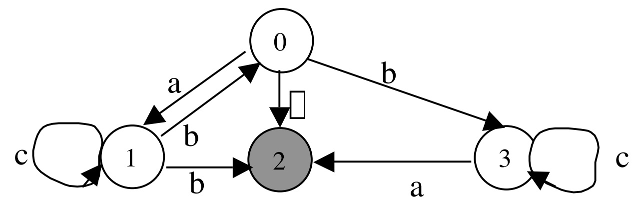
	a	**b**	**c**	**epsilon**
-->0	{1}	{3}		{2}
1		{0,2}	{1}	
* **2**	–	–	–	–
3	{2}	–	{3}	–

avente stato iniziale 0 e stato finale 2.

a) Si dia la rappresentazione grafica.
b) Si costruisca un automa equivalente deterministico
c) Si definisca la grammatica regolare equivalente all’automa deterministico.
d) Si trasformi la grammatical regolare in una grammatical libera, eliminando gli operatori * e | .

Soluzione

a)



b)

add(clos(0))=add({0,2}) Map(0)={0,2} D=<{0},{0},□, {}>

add(move(Map(0),a)=add(move({0,2},a)=add({1}) Map(1)={1} edge(0,1,a)
D=<{0,1},{0},□, {<0,1,a>}>

add(move(Map(0),b)=add(move({0,2},b)=add({3}) Map(2)={3} edge(0,2,b)
D=<{0,1,2},{0},□, {<0,1,a>,<0,2,b>}>

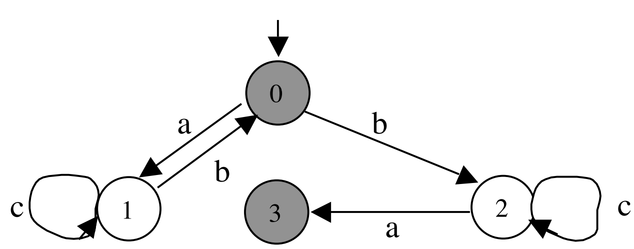
add(move(Map(1),b))=add(move({1},b)=add(close(0) □ clos(2))=add({0,2}) edge(1,0,b)
D =<{0,1,2},{0},□, {<0,1,a>,<0,2,b>,<1,0,b>}>}>

add(move(Map(1),c))=add(move({1},c))=add(close(1))=add({1}) edge(1,1,c)
D=<{0,1,2},{0},□, {<0,1,a>,<0,2,b>,<1,0,b>,<1,1,c>}>

add(move(Map(2),a))=add(move({3},a)=add(close(2))=add({2}) Map(3)={2}
edge(2,3,a) D=<{0,1,2,3},{0},□, {<0,1,a>,<0,2,b>,<1,0,b>,<1,1,c>,<2,3,a>}>

add(move(Map(2),c))=add(move({3},c))=add(clos(3))=add({3}) edge(2,3,c)
D=<{0,1,2,3},{0},{0}, {<0,1,a>,<0,2,b>,<1,0,b>,<1,1,c>,<2,3,a>,<2,2,c>}>

Graficamente l'automa deterministico:



c) G=<{a,b,c},{S},S{S::=(a c* b)*(□| bc*a)

d) G=<{a,b,c},{S},S{S::=AB, A::= □, A::=a C b A, B::=□, B::= b C a, C::= □,C::=cC}

Esercizio 3

Data la seguente grammatica:

G=<{a,b,c}, {S,A,B},S,{S::=AcB, A::=aA, B::=bB} si definisca il sistema di transizioni

Soluzione

Il sistema di transizioni per G e dato da <□,T,Rel>

□ ={□| □□ ({a,b,c}□{S,A.B})*}

T ={□| □□ {a,b,c}*}

Rel={

□,□□ ({a,b,c}□{S,A,B}))}

□S□□ □AcB□

□,□□ ({a,b,c}□{S,A,B}))}

□A□□ □aA□

□,□□ ({a,b,c}□{S,A,B}))}

□B□□ □bB□

}

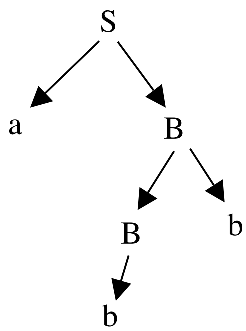
Esercizio 4

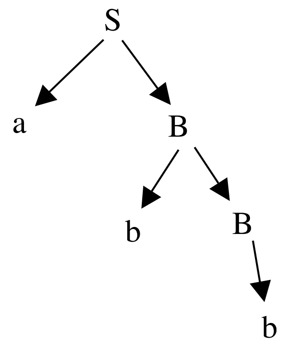
Si mostri che la seguente grammatica è ambigua:

G=<{a,b}, {S,B},S,{S::=aB, B::=bB, B::=Bb|b}

Soluzione:

Ad esempio la sequenza di simboli abb è frontiera di entrambi i seguenti aberi di derivazione sintattica:





Esercizio 5

Si consideri il linguaggio L definito dalla seguente espressione regolare:

(abc)* |(a*bc*)d(c*ba*)|(c|ba)*

Quali delle espressioni seguenti definisce un linguaggio contenuto in L?

1) (a*bc)*

2) (ab)*

3) (bc*)db

4) (ba)*

Soluzione

Il linguaggio è l'unione di tre sottolinguaggi:

a) (abc)*
b) (a*bc*)d(c*ba*)
c) (c|ba)*

allora abbiamo

1) Non appartiene al sotto linguaggio a) che contiene solo ε e stringhe abc ripetute un qualunque numero di volte, mentre 1) contiene anche stringhe del tipo aaabc, aabbc ecc. Non appartiene al sotto linguaggio b) perchè questo contiene stringhe che dopo elementi di (a*bc)* hanno una d e almeno una b. Infine non appartiene a c) perchè quest'ultimo contiene oltre ad ε, stringhe che contengono solo c e b. Quindi 1) non appartiene al linguaggio

2) Non appartiene al sottolinguaggio a) che contiene solo □e stringhe ab ripetute un qualunque numero di volte, non appartiene al sottolinguaggiob) perchè questo contiene stringhe che dopo elementi di (a*bc)* abbiano una d e almeno una b. non appartiene a c) perchè quest'ultimo contiene oltre ad □stringhe che contengono solo c e b. Quindi 2) non appartiene al linguaggio
3) Appartiene al sottolinguaggio b) nel caso (□bc*)d(□b □). Quindi 3) appartiene al linguaggio
4) Appartiene al sottolinguaggio c) (ba)* □ (c | ba)*. Quindi 4) appartiene al linguaggio