

Corso di laurea in Informatica Applicata

Fondamenti di Programmazione

Appello 6/2/03

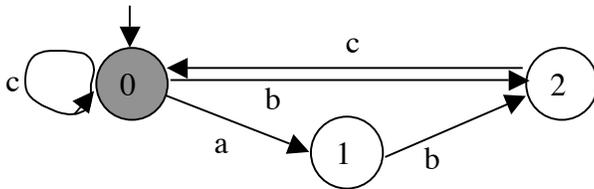
Prima parte

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' è sempre seguito dal simbolo 'c'. Si definiscano l'automa deterministico e la grammatica regolare per $L(A)$.

Soluzione

Automa per $L(A)$



Grammatica regolare

$G = \langle \{a,b,c\}, \{S\}, S, \{S ::= c^* \mid (bc)^* \mid (abc)^*\} \rangle$

Esercizio 1

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

	a	b	c	epsilon
-> 0	{1,2}	4	-	-
1	3	1	-	-
2	3	-	2	-
* 3	-	-	-	-
4	-	-	-	2

sull'alfabeto $\{a,b,c\}$, e avente stato iniziale 0 e stato finale 3.

- a) Si dia la rappresentazione grafica.
- b) Si costruisca un automa equivalente deterministico. Si mostrino i passi intermedi del processo di calcolo con i risultati del calcolo delle funzioni

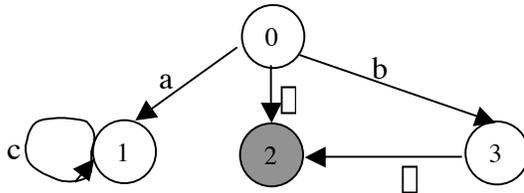
add, move e clos,

c) Si definisca la grammatica regolare equivalente all'automa minimo.

d) Si trasformi la grammatica regolare in una grammatica libera, eliminando gli operatori * e |.

Soluzione

a)



b)

$\text{add}(\text{clos}(0)) = \text{add}(\{0,2\})$ $\text{Map}(0) = \{0,2\}$ $D = \langle \{0\}, \{0\}, \square, \{ \} \rangle$

$\text{add}(\text{move}(\text{Map}(0), a)) = \text{add}(\text{move}(\{0,2\}, a)) = \text{add}(\{1\})$ $\text{Map}(1) = \{1\}$ $\text{edge}(0,1,a)$
 $D = \langle \{0,1\}, \{0\}, \square, \{ \langle 0,1,a \rangle \} \rangle$

$\text{add}(\text{move}(\text{Map}(0), b)) = \text{add}(\text{move}(\{0,2\}, b)) = \text{add}(\{3\})$ $\text{Map}(2) = \{3\}$ $\text{edge}(0,2,b)$
 $D = \langle \{0,1,2\}, \{0\}, \square, \{ \langle 0,1,a \rangle, \langle 0,2,b \rangle \} \rangle$

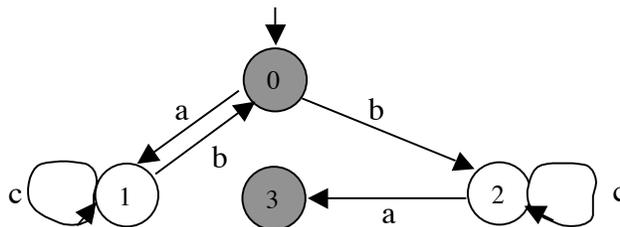
$\text{add}(\text{move}(\text{Map}(1), b)) = \text{add}(\text{move}(\{1\}, b)) = \text{add}(\text{close}(0) \square \text{clos}(2)) = \text{add}(\{0,2\})$ $\text{edge}(1,0,b)$
 $D = \langle \{0,1,2\}, \{0\}, \square, \{ \langle 0,1,a \rangle, \langle 0,2,b \rangle, \langle 1,0,b \rangle \} \rangle$

$\text{add}(\text{move}(\text{Map}(1), c)) = \text{add}(\text{move}(\{1\}, c)) = \text{add}(\text{close}(1)) = \text{add}(\{1\})$ $\text{edge}(1,1,c)$
 $D = \langle \{0,1,2\}, \{0\}, \square, \{ \langle 0,1,a \rangle, \langle 0,2,b \rangle, \langle 1,0,b \rangle, \langle 1,1,c \rangle \} \rangle$

$\text{add}(\text{move}(\text{Map}(2), a)) = \text{add}(\text{move}(\{3\}, a)) = \text{add}(\text{close}(2)) = \text{add}(\{2\})$ $\text{Map}(3) = \{2\}$
 $\text{edge}(2,3,a)$ $D = \langle \{0,1,2,3\}, \{0\}, \square, \{ \langle 0,1,a \rangle, \langle 0,2,b \rangle, \langle 1,0,b \rangle, \langle 1,1,c \rangle, \langle 2,3,a \rangle \} \rangle$

$\text{add}(\text{move}(\text{Map}(2), c)) = \text{add}(\text{move}(\{3\}, c)) = \text{add}(\text{clos}(3)) = \text{add}(\{3\})$ $\text{edge}(2,3,c)$
 $D = \langle \{0,1,2,3\}, \{0\}, \{0\}, \{ \langle 0,1,a \rangle, \langle 0,2,b \rangle, \langle 1,0,b \rangle, \langle 1,1,c \rangle, \langle 2,3,a \rangle, \langle 2,2,c \rangle \} \rangle$

Graficamente l'automa deterministico:



$$c) G = \langle \{a,b,c\}, \{S\}, S \{S ::= (a c^* b)^* (\epsilon | bc^* a)\} \rangle$$

$$d) G = \langle \{a,b,c\}, \{S\}, S \{S ::= AB, A ::= \epsilon, A ::= a C b A, B ::= \epsilon, B ::= b C a, C ::= \epsilon, C ::= cC\} \rangle$$

Esercizio 3

Data la seguente grammatica:

$$G = \langle \{a,b,c\}, \{S,A,B\}, S, \{S ::= AcB, A ::= aA, B ::= bB\} \rangle$$

Soluzione

Il sistema di transizioni per G è dato da $\langle \Sigma, T, Rel \rangle$

$$\Sigma = \{ \epsilon \mid \epsilon \epsilon (\{a,b,c\} \cup \{S,A,B\})^* \}$$

$$T = \{ \epsilon \mid \epsilon \epsilon \{a,b,c\}^* \}$$

Rel = {

$$\frac{\epsilon, \epsilon \epsilon (\{a,b,c\} \cup \{S,A,B\})}{\epsilon S \epsilon \epsilon \epsilon AcB \epsilon}$$

$$\frac{\epsilon, \epsilon \epsilon (\{a,b,c\} \cup \{S,A,B\})}{\epsilon A \epsilon \epsilon \epsilon aA \epsilon}$$

$$\frac{\epsilon, \epsilon \epsilon (\{a,b,c\} \cup \{S,A,B\})}{\epsilon B \epsilon \epsilon \epsilon bB \epsilon}$$

}

Esercizio 4

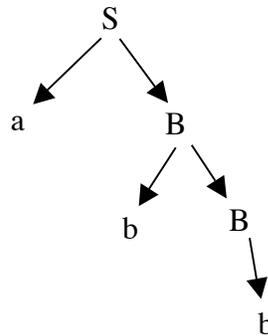
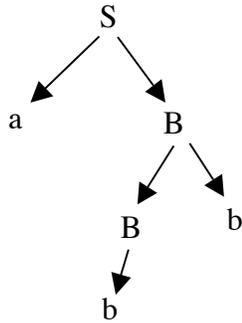
Si mostri che la seguente grammatica è ambigua:

$$G = \langle \{a,b\}, \{S,B\}, S, \{S ::= aB, B ::= bB, B ::= Bbb\} \rangle$$

Soluzione:

Ad esempio la sequenza di simboli abb è frontiera di entrambi i seguenti alberi di

derivazione sintattica:



Esercizio 5

Si consideri il linguaggio L definito dalla seguente espressione regolare:

$(abc)^* \cup (a^*bc^*)d(c^*ba^*) \cup (clba)^*$

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) $(clba)^*$

allora abbiamo

- 1) Non appartiene al sotto linguaggio a) che contiene solo \square 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 \square stringhe che contengono solo c e b. Quindi 1) non appartiene al linguaggio

- 2) Non appartiene al sottolinguaggio a) che contiene solo ϵ e stringhe ab ripetute un qualunque numero di volte, non appartiene al sottolinguaggio b) 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 ϵ stringhe che contengono solo c e b. Quindi 2) non appartiene al linguaggio
- 3) Appartiene al sottolinguaggio b) nel caso $(\epsilon bc^*)d(\epsilon b \epsilon)$. Quindi 3) appartiene al linguaggio
- 4) Appartiene al sottolinguaggio c) $(ba)^* \epsilon (c \mid ba)^*$. Quindi 4) appartiene al linguaggio

Seconda parte

ESERCIZIO 1

Si supponga di estendere la sintassi dei comandi con la dichiarazione multipla così definita:

$$\text{Decl} ::= \text{Type } \text{Ide}_1, \text{Ide}_2, \dots, \text{Ide}_k = \text{Exp}_1, \text{Exp}_2, \dots, \text{Exp}_k$$

Con il significato informale che gli identificatori Ide_i sono tutti del medesimo tipo Type e all'identificatore i -esimo viene assegnato il valore dell'espressione i -esima. I valori di tutte le espressioni vengono valutate nello stato in cui viene valutata la dichiarazione.

Dare la semantica operativa della nuova dichiarazione, con riferimento al modello in cui lo stato è composto solo da stack di frames.

Soluzione

$$\text{mul} = \frac{\langle \text{Exp}_1, \epsilon \rangle \stackrel{\text{exp}}{\vdash} v_1 \dots \langle \text{Exp}_k, \epsilon \rangle \stackrel{\text{exp}}{\vdash} v_k \quad \epsilon' = \epsilon [v_1 / \text{Ide}_1] \dots [v_k / \text{Ide}_k]}{\langle \text{Type } \text{Ide}_1, \text{Ide}_2, \dots, \text{Ide}_k = \text{Exp}_1, \text{Exp}_2, \dots, \text{Exp}_k, \epsilon \rangle \stackrel{\text{com}}{\vdash} \epsilon'}$$

ESERCIZIO 2

Con riferimento alle regole definite per la dichiarazione multipla si dimostri che i seguenti frammenti di programma sono equivalenti a partire da un generico stato ϵ :

- I. `int x,y=3,4; y=x+y;`
- II. `int x=3; int y=x+4;`

Soluzione

```

I:C[[y=x+y;]] □[3/x][4/y] □
      E[[x+y]] □[3/x][4/y] □ 7
      □[3/x][4/y][7/y]
II:C[[y=x+4]] □[3/x] □
      E[[x+4]] □[3/x] □ 7
      □[3/x][7/y]

```

ESERCIZIO 3

Si vuole aggiungere alla classe Arrays vista a lezione, un nuovo metodo statico CambiaElem. L'intestazione di tale metodo è:

```

public static boolean IsOrd (int [ ] a)

/** calcolatruce se l'array a e` ordinato in senso decrescente .

param a: un array di interi..*/

```

Si definisca il corpo del metodo, in modo che calcoli true se a e` ordinato in senso decrescente , false altrimenti. Ad esempio la chiamata Arrays.Isord(a) calcola false se a e` rappresentato dalla seguente tabella:

23	0	-9	-96	6	7	-4	6
----	---	----	-----	---	---	----	---

mentre la chiamata Arrays.Isord(a) calcola true se a e` rappresentato dalla seguente tabella:

123	15	10	7	0	-1	-4	-6
-----	----	----	---	---	----	----	----

Soluzione

```

public static boolean IsOrd (int [ ] a){
    boolean ord=true;
    int i=0;
    while (i<a.length-1 & ord)
        if a[i]<=a[i+1] i++;
        else ord=false;
    return ord; }

```

ESERCIZIO 4

Dato il seguente programma:

```

prog {class ClasseA{
public int x;
    }
class ClasseB{

```

```

public int y;
public int x;
public void UpdMe(int i) {
int og1=i;
if (og1 >0) this.y=i;      (5)
}
}
}                               (1)
{ClasseA og1= new ClasseA();
ClasseB og2= new ClasseB(); (2)
og1.x=100;
og2.x=0;
og2.y=1;                      (3)
og2.UpdMe(og1.x);             (4)
}}

```

rappresentare graficamente:

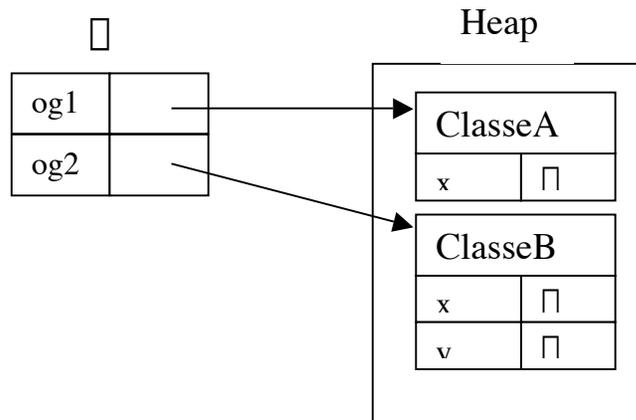
- I. l'ambiente delle classi al punto (1);
- II. lo stack di frames e lo heap dopo l'esecuzione del comando (2),
- III. lo stack di frames e lo heap dopo l'esecuzione del comando (3),
- IV. lo stack di frames e lo heap dopo l'esecuzione del comando (4),
- V. lo stack di frame e lo heap prima e dell'esecuzione del comando (5) (esecuzione del metodo UpdMe invocato in (4)).
- VI. lo stack di frame e lo heap dopo l'esecuzione del comando (5) (esecuzione del metodo UpdMe invocato in (4)).

Soluzione

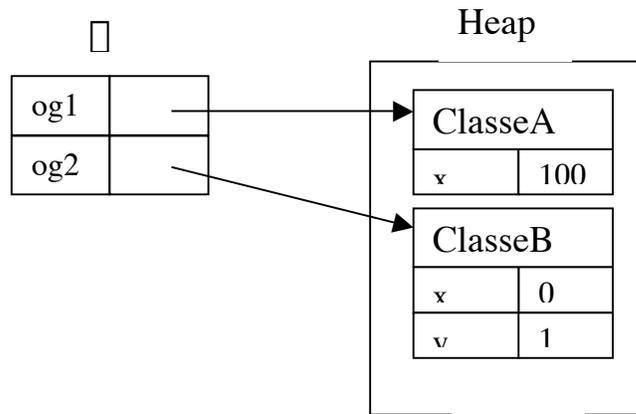
I.(1)

ClasseA	{(x, □)}	□
ClasseB	{(y, □), (x, □)}	UpdMe i int og1=i; if (og1>0) {this.y=i;}

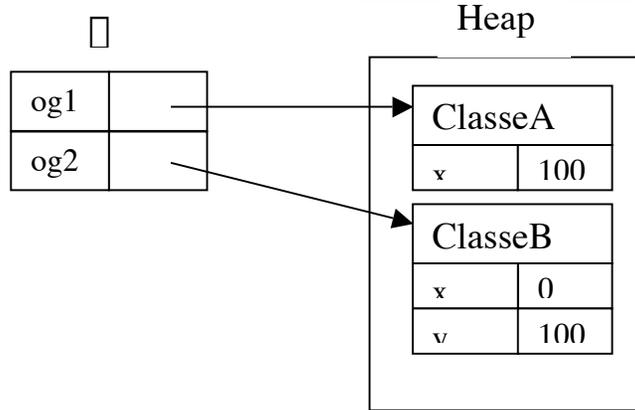
II (2)



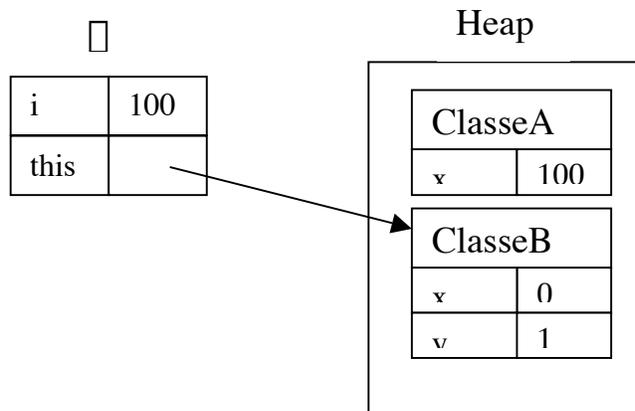
III(3)



IV(4)



V



VI

