PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2013/2014
Verifica scritta del 4/11/2013

SOLUZIONI PROPOSTE

ESERCIZIO 1 (6 punti)
Definire una grammatica libera che genera il seguente linguaggio sull'alfabeto $\Lambda = \{a, b\}$.

$$\mathcal{L} = \left\{ a^n b^k \mid n, k > 0 \wedge n \neq k + 1 \right\}$$

Dimostrare poi che la stringa *aabb* appartiene al linguaggio generato dalla grammatica utilizzando il teorema di ricorsione.

Soluzione

```
S ::= aSb | aBb | aAb | ab
A ::= aa | aA
B ::= b | bB
```

Il sistema di equazioni ricorsive corrispondente alla grammatica costruita è il seguente:

$\mathtt{S} = \{\mathtt{a}\}\mathtt{S}\{\mathtt{b}\} \cup \{\mathtt{a}\}\mathtt{B}\{\mathtt{b}\} \cup \{\mathtt{a}\}\mathtt{A}\{\mathtt{b}\} \cup \{\mathtt{a}\}\{\mathtt{b}\}$
$\mathtt{A} = \{\mathtt{a}\}\{\mathtt{a}\} \cup \{\mathtt{a}\}\mathtt{A}$
$\mathtt{B} = \{\mathtt{b}\} \cup \{\mathtt{b}\}\mathtt{B}$

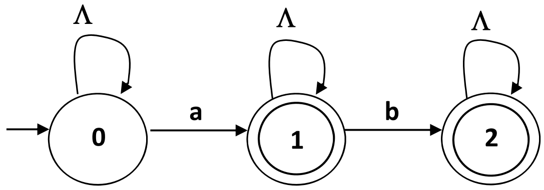
Calcoliamo le approssimazioni del minimo punto fisso del sistema fino a quando la stringa `aabb` compare nell'approssimazione associata alla categoria sintattica principale.

$\mathtt{S}^0 = \{\}$
$\mathtt{A}^0 = \{\}$
$\mathtt{B}^0 = \{\}$

$\mathtt{S}^1 = \{\mathtt{a}\}\mathtt{S}^0\{\mathtt{b}\} \cup \{\mathtt{a}\}\mathtt{B}^0\{\mathtt{b}\} \cup \{\mathtt{a}\}\mathtt{A}^0\{\mathtt{b}\} \cup \{\mathtt{a}\}\{\mathtt{b}\} = \{\} \cup \{\} \cup \{\} \cup \{\mathtt{ab}\} = \{\mathtt{ab}\}$
$\mathtt{A}^1 = \{\mathtt{a}\}\{\mathtt{a}\} \cup \{\mathtt{a}\}\mathtt{A}^0 = \{\mathtt{a}\}\{\mathtt{a}\} \cup \{\mathtt{a}\}\{\} = \{\mathtt{aa}\}$
$\mathtt{B}^1 = \{\mathtt{b}\} \cup \{\mathtt{b}\}\mathtt{B}^0 = \{\mathtt{b}\} \cup \{\mathtt{b}\}\{\} = \{\mathtt{b}\}$

$\mathtt{S}^2 = \{\mathtt{a}\}\mathtt{S}^1\{\mathtt{b}\} \cup \{\mathtt{a}\}\mathtt{B}^1\{\mathtt{b}\} \cup \{\mathtt{a}\}\mathtt{A}^1\{\mathtt{b}\} \cup \{\mathtt{a}\}\{\mathtt{b}\} = \{\mathtt{a}\}\{\mathtt{ab}\}\{\mathtt{b}\} \cup \{\mathtt{a}\}\mathtt{B}^1\{\mathtt{b}\} \cup \{\mathtt{a}\}\mathtt{A}^1\{\mathtt{b}\} \cup \{\mathtt{a}\}\{\mathtt{b}\} = \{\mathtt{aabb}\} \cup \dots$

ESERCIZIO 2 (6 punti)
Sia Λ un alfabeto con $a, b \in \Lambda$. Dato il seguente automa a stati finiti



(i) Indicare formalmente il linguaggio riconosciuto dall'automa

(ii) Utilizzare la tecnica della costruzione dei sottinsiemi per costruire un automa **deterministico** che riconosce lo stesso linguaggio

Soluzione
Il linguaggio riconosciuto dall'automa può essere espresso come l'unione di due linguaggi $\mathcal{L}_1$ e $\mathcal{L}_2$: il primo corrisponde alle stringhe il cui cammino di riconoscimento termina nello stato **1** ed il secondo corrisponde alle stringhe il cui cammino di riconoscimento termina nello stato **2**.

$\mathcal{L}_1 = \{\alpha \mathtt{a} \beta \mid \alpha, \beta \in \Lambda^*\}$

$\mathcal{L}_2 = \{\alpha \mathtt{a} \beta \mathtt{b} \gamma \mid \alpha, \beta, \gamma \in \Lambda^*\}$

Poiché è evidente che $\mathcal{L}_2 \subseteq \mathcal{L}_1$, il linguaggio riconosciuto dall'automa è $\mathcal{L}_1$.

Utilizziamo la tecnica della costruzione dei sottinsiemi.

Stato	a	b	$\Lambda \setminus \{\mathtt{a}, \mathtt{b}\}$
{0}	{0,1}	{0}	{0}
{0,1} *	{0,1}	{0,1,2}	{0,1}
{0,1,2} *	{0,1,2}	{0,1,2}	{0,1,2}

Gli stati evidenziati con * sono stati di riconoscimento. La rappresentazione grafica è lasciata per esercizio.

ESERCIZIO 3 (6 punti)
Indicare il tipo delle seguenti funzioni `CAML`

(i) `let f x y z = (y x) = (z y);;`

(ii) `let f x y z = x ::(z x)::(y x);;`

(iii)
```
let f x y = match x with
  [] -> [y]
  | w::ws when (y w)>1 -> []
  | _ -> [y;y];;
```

Soluzione

(i) `f :  'a -> ('a -> 'b) -> (('a -> 'b) -> 'b) -> bool`

(ii) `f :  'a -> ('a -> 'a list) -> ('a -> 'a) -> 'a list`

(iii) `f :  'a list -> ('a -> int) -> ('a -> int) list`

ESERCIZIO 4 (6 punti)
Definire una funzione ricorsiva f da coppie di naturali in naturali che soddisfi la seguente proprietà

$$\forall n, m \in \mathbb{N}.\ f(n,m) = n + m + 2$$

in modo che la relazione di precedenza indotta $\prec_f$ sia la seguente

$$(x,y) \prec_f (n,m) \equiv (n \geq 2 \ \wedge \ m \geq 1 \ \wedge \ x = n-2 \ \wedge \ y = m-1)$$

Dimostrare per induzione ben fondata la correttezza della soluzione proposta.

Soluzione
Una possibile rappresentazione grafica della relazione indicata è la seguente

```
     .            .            .
     .            .            .
     .            .            .
 (4,x+2)      (5,y+2)      (y+4,2)
     |            |            |
 (2,x+1)      (3,y+1)      (y+2,1)
     |            |            |
   (0,x)        (1,y)        (y,0)
```

dove `x` indica un numero naturale e `y` un numero naturale diverso da `0`.

Dunque:

$$f(n,m) = \begin{cases} m+2 & \text{se } n = 0 \wedge m > 0 \\ m+3 & \text{se } n = 1 \wedge m > 0 \\ n+2 & \text{se } m = 0 \\ f(n-2, m-1) + E & \text{se } n > 1 \wedge m > 0 \end{cases}$$

Per determinare E ci facciamo guidare dalla dimostrazione per induzione ben fondata (caso induttivo). Determiniamo cioè un valore di E in modo che valga la seguente implicazione, supponendo $n > 1 \wedge m > 0$:

$$f(n-2, m-1) = (n-2) + (m-1) + 2 \Longrightarrow f(n,m) = n + m + 2$$

$f(n,m)$
$=$ {def. di f, quarto caso}
$f(n-2, m-1) + E$
$=$ {Ip. induttiva}
$(n-2) + (m-1) + 2 + E$
$=$ {calcolo}
$n + m - 1 + E$
$=$ {**E=3**}
$n + m + 2$

Dunque la precedente definizione è corretta con $E = 3$. La dimostrazione di correttezza nei tre casi base è ovvia e dunque omessa.

ESERCIZIO 5 (6 punti)
Definire una funzione `f` con tipo

```
foo : 'a list -> 'a -> 'a list
```

in modo che (`foo` ℓ x) sia la lista ottenuta da ℓ aggiungendo x in fondo alla lista, se x non è già presente in ℓ, lasciando la lista inalterata altrimenti. Ad esempio

```
foo [50; 20; 10] 80 = [50; 20; 10; 80]

foo [50; 20; 10] 20 = [50; 20; 10]
```

Soluzione
Una possibile soluzione è la seguente:

```
let rec foo lis x = match lis with
       [] -> [x]
    |  z::zs when x=z -> z::zs
    |  z::zs when x<>z -> z :: (foo zs x);;
```