PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2009-2010
Esercitazione del 27/10/2011

ESERCIZIO 1
Definire una grammatica libera che genera il seguente linguaggio sull'alfabeto $\Lambda = \{a, b, c\}$

$$\mathcal{L} = \left\{a^n b^k \mid 0 < n \leq k\right\} \cup \left\{b^n c^k \mid 0 < k \leq n\right\}$$

Soluzione
Definiamo dapprima sepratamente due grammatiche G_1 e G_2 che generano rispettivamente il linguaggio $\left\{a^n b^k \mid 0 < n \leq k\right\}$ e il linguaggio $\left\{b^n c^k \mid 0 < k \leq n\right\}$.

- $G_1 = \langle \Lambda, \{A\}, A, P_1 \rangle$, dove P_1 contiene le seguenti produzioni:

  ```
  A ::= aAb | Ab | ab
  ```

- $G_2 = \langle \Lambda, \{B\}, B, P_2 \rangle$, dove P_2 contiene le seguenti produzioni:

  ```
  B ::= bBc | bB | bc
  ```

La grammatica richiesta è allora $G = \langle \Lambda, \{S, A, B\}, S, P \cup P_1 \cup P_2 \rangle$ dove P contiene le seguenti produzioni:

```
S ::= A | B
```

ESERCIZIO 2
Data la seguente grammatica libera sull'alfabeto $\Lambda = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

```
S ::= AB
A ::= aAb | aA | a
B ::= bBc | Bc | c
```

(i) Indicare formalmente il linguaggio generato

(ii) Trasformare la grammatica in un sistema di equazioni ricorsive

(iii) Calcolare le prime quattro approssimazioni del minimo punto fisso del sistema determinato al punto precedente

Soluzione

(i) Osserviamo che le stringhe generate dalla categoria sintattica principale sono della forma $\alpha_1\alpha_2$, dove α_1 è una stringa generata dalla categoria sintattica `A` e α_2 è una stringa generata dalla categoria sintattica `B`. In altre parole il linguaggio generato può essere visto come

$$\mathcal{L} = \mathcal{L}_A \mathcal{L}_B$$

dove $\mathcal{L}_A$ e $\mathcal{L}_B$ sono, rispettivamente, i linguaggi generati da `A` e `B`. È facile convincersi che essi sono:

$$\mathcal{L}_A = \left\{\mathtt{a}^n \mathtt{b}^k \mid 0 \leq k < n\right\}$$

e

$$\mathcal{L}_B = \left\{\mathtt{b}^m \mathtt{c}^h \mid 0 \leq m < h\right\}$$

Dunque

$$\mathcal{L} = \left\{\mathtt{a}^n \mathtt{b}^k \mathtt{b}^m \mathtt{c}^h \mid 0 \leq k < n \wedge 0 \leq m < h\right\}$$

che può essere riscritto in modo più compatto come

$$\mathcal{L} = \left\{\mathtt{a}^n \mathtt{b}^t \mathtt{c}^h \mid 0 \leq t \leq n + h - 2\right\}$$

(ii) Il sistema di equazioni ricorsive è il seguente:

$S = AB$

$A = \{a\}A\{b\} \cup \{a\}A \cup \{a\}$

$B = \{b\}B\{c\} \cup B\{c\} \cup \{c\}$

(iii) Calcolo delle prime quattro approssimazioni del minimo punto fisso:

$S^0 = \{\}$

$A^0 = \{\}$

$B^0 = \{\}$

$S^1 = A^0B^0 = \{\}$

$A^1 = \{a\}A^0\{b\} \cup \{a\}A^0 \cup \{a\} = \{a\}$

$B^1 = \{b\}B^0\{c\} \cup B^0\{c\} \cup \{c\} = \{c\}$

$S^2 = A^1B^1 = \{ac\}$

$A^2 = \{a\}A^1\{b\} \cup \{a\}A^1 \cup \{a\} = \{a, aa, aab\}$

$B^2 = \{b\}B^1\{c\} \cup B^1\{c\} \cup \{c\} = \{c, cc, bcc\}$

$S^3 = A^2B^2 = \{ac, acc, abcc, aac, aacc, aabcc, aabc, aabcc, aabbcc\}$

$A^3 = \{a\}A^2\{b\} \cup \{a\}A^2 \cup \{a\} = \{a, aa, aaa, aab, aaab, aaabb\}$

$B^3 = \{b\}B^2\{c\} \cup B^2\{c\} \cup \{c\} = \{c, cc, ccc, bcc, bccc, bbccc\}$

ESERCIZIO 3

Indicare il tipo delle seguenti funzioni

(i) `let f x y z = 1 + (z (x y) y);;` (ii) `let f x y = let g z = if z>1 then x else y in g;;`

Soluzione

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

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

ESERCIZIO 4
Definire una funzione ricorsiva f da coppie di naturali in naturali che soddisfi la proprietà

$$\forall n, m \in \mathbb{N}.\ f(n,m) = 5$$

in modo che la relazione di precedenza indotta sia quella definita formalmente come segue

$$\forall n, m, n', m' \in \mathbb{N}.(n', m') \sqsubset (n, m) \equiv m \neq 0 \wedge m \neq 1 \wedge m' = m - 2 \wedge n' = n + 1$$

Disegnare il diagramma della relazione indotta e dimostrare per induzione ben fondata la correttezza della funzione proposta.

Soluzione
Il diagramma della relazione indotta si può rappresentare come segue

```
m pari     (n,m)              m dispari  (n,m)
             |                              |
         (n+1,m-2)                      (n+1,m-2)
              |                              |
          (n+2, m-4)                    (n+2, m-4)
              |                              |
              .                              .
              .                              .
              |                              |
       (n+(m-2)/2, 2)                (n+(m-2)/2, 3)
             |                               |
          (n+m/2, 0)                     (n+m/2, 1)
```

dove `/` indica il quoziente della divisione intera.

La funzione è definita come segue:

$$f(x,y) = \begin{cases} 5 & \text{se } y = 0 \vee y = 1 \\ f(x+1, y-2) & \text{altrimenti} \end{cases}$$

Infine, la dimostrazione per induzione ben fondata prevede due casi base e un caso induttivo.

Caso base 1
$\quad f(n,0)$
$= \{\text{def. di } f\}$
$\quad 5$

Caso base 2
$\quad f(n,1)$
$= \{\text{def. di } f\}$
$\quad 5$

Caso induttivo
$\quad f(n,m)$
$= \{m \neq 0 \text{ e } m \neq 1, \text{ def. di } f\}$
$\quad f(n+1, m-2)$
$= \{(n+1, m-2) \sqsubset (n,m), \text{ Ipotesi induttiva}\}$
$\quad 5$

ESERCIZIO 5
Definire una funzione `del` con tipo

```
del : 'a list-> ('a -> bool) -> int -> 'a list
```

in modo che (`del` ℓ `p n`) sia la lista ottenuta eliminando da ℓ i primi n elementi che soddisfano p. Ad esempio,
`del` $[4; 1; 3; 5; 6]$ *dispari* $2 = [4; 5; 6]$
`del` $[4; 1; 3; 5; 6]$ *dispari* $10 = [4; 6]$

Soluzione

```
let rec del xs p n = match (xs, n) with
     [], _ -> []            |
     ys, 0 -> ys            |
     y::ys, n when n>0 & (p y) -> del ys p (n-1) |
     y::ys, n when n>0 & not (p y) -> y::(del ys p n) ;;
```

ESERCIZIO 6
Completare la seguente definizione della funzione `foo` con tipo `int list -> int -> (int * int list)`:

```
let foo lis n = let f x y = ...
  in foldr f ... lis
```

in modo che `foo` ℓ `x = (n,` $\ell 1$`)` se e solo se $\ell 1$ è la lista ottenuta da ℓ eliminando tutti gli elementi maggiori di `x` e `n` è il numero di elementi cancellati.

Soluzione

```
let foo lis n =
      let f x y = let (m,ys)=y in if (x>n) then (m+1,ys) else (m, x::ys)
      in
         foldr f (0,[]) lis
```