

# Linguaggi e Traduttori

Prof. Marco Gavanelli

26 luglio 2018

## Esercizio 1 (punti 16)

Si scriva in Haskell una funzione di ordine superiore

```
longest :: (a -> a -> Bool) -> [a] -> [a]
```

che prende come parametri

- una funzione  $f$
- una lista  $xs$

e fornisce la sottosequenza più lunga della lista  $xs$  tale che  $f(x_i, x_{i+1})$  sia vero (se  $x_i$  e  $x_{i+1}$  appartengono alla sottosequenza  $xs$ ).

Ad esempio, nell'invocazione

```
longest (<) [1,3,65,4,3,45,3,5,4,5,6,8,9,0,1,3]
```

si cercano le sequenze in cui ogni elemento è in relazione  $<$  con il successivo; tali sequenze sono quindi

- [1, 3, 65]
- [3, 45]
- [3, 5]
- [4, 5, 6, 8, 9]
- [0, 1, 3]

Di queste, la sequenza più lunga è [4, 5, 6, 8, 9], per cui la funzione dovrà fornire questo risultato.

Altri esempi:

```
*Main> longest (==) "abaaabdddcdeeefffffffdssawwwaasa"  
"ffffff"  
*Main> longest (\x y -> x+1==y) [1,2,3,5,7,9,10,11,12,13,15,17,18]  
[9,10,11,12,13]
```

```

    oppure, supponendo di definire
divisibile a b = mod a b == 0

    allora
*Main> longest divisibile [10,5,1,16,16,8,8,2,2,1,7,3,15,5]
[16,16,8,8,2,2,1]

```

## Esercizio 2 (punti 3)

Si desidera creare un programma che visualizza il codice di un programma mettendo in evidenza il livello di annidamento delle parentesi, utilizzando colori diversi. Più in dettaglio, quando viene aperta una parentesi (che può essere una parentesi tonda '(', quadra '[' o graffa '{') si deve cambiare il colore di visualizzazione. Il colore viene ripristinato quando si incontra una parentesi chiusa (tonda ')', quadra ']' o graffa '}').

Ad esempio, il codice di una funzione C potrebbe essere visualizzato come segue:

```

int f(int A[])
{ int i,B[10];
  for(i=0;i<g(A[i]);i++)
  { B[i] = A[i]; }
  return i;
}

```

Per cambiare colore, nei programmi che vengono eseguiti nella shell di Unix, è sufficiente stampare a video una speciale stringa, costruita come segue:

- inizialmente si stampa la stringa "\033["
- poi si visualizza un intero compreso fra 30 e 37
- poi si visualizza il carattere 'm'

A seconda dell'intero visualizzato, si otterrà un colore diverso:

black	30
red	31
green	32
yellow	33
blue	34
magenta	35
cyan	36
white	37

Ad esempio, se viene eseguita la seguente istruzione C:

```
printf("il \033[31mrosso\033[30m e il \033[34mblu\033[30m.");
```

verrà visualizzato:

il rosso e il blu.

## Soluzione 1

```
longest f (x:xs) =
    let sequenze = split_sequences f (x:xs)
    in massimo length sequenze

split_sequences f (x:xs) =
    let coppie = zip (x:xs) xs
    in map estraiSequenza (split_sequences_pairs (\(a,b) -> f a b) coppie)

-- data una sequenza costituita da coppie
-- es [(1,3),(3,4),(4,5)]
-- fornisce tutti gli elementi: [1,3,4,5]
estraiSequenza [] = []
estraiSequenza (a:as) = (fst a):map snd (a:as)

split_sequences_pairs f [] = []
split_sequences_pairs f coppie =
    let elemento = takeWhile f coppie
        resto = dropWhile (not.f) (dropWhile f coppie)
    in elemento:split_sequences_pairs f resto

massimo f [x] = x
massimo f (x:xs) =
    let m1 = massimo f xs
    in if (f m1 > f x) then m1 else x
```

## Soluzione 2

```
%{
    int depth=0;
%}

OPEN    [\{\[ (
CLOSED  [\}\] )

%%

{OPEN}    {depth++; printf("\033[%dm%s",depth+30,yytext); }
{CLOSED}  {depth--; printf("%s\033[%dm",yytext,depth+30); }
```