

Esempi di Programmazione Curry

Marco Comini

24 marzo 2006

In questo scritto intendo mostrare alcuni esempi di programmi in Curry per impratichirsi con le features tipiche del linguaggio. Per la parte “deterministica” di Curry valgono in buona sostanza esempi ed esercizi di Haskell, che vi invito a fare prima di prendere in considerazione quanto segue in quanto qui intendo focalizzarmi sulla parte che sfrutta la potenza espressiva delle variabili logiche trattando contemporaneamente programmi più “sostanziosi”.

1 Soluzione di Problemi di Ricerca mediante opportuna combinazione di constraints e generazione non-deterministica

1. Si vuole trovare una soluzione al gioco dei Missionari e Cannibali. Ci sono 3 missionari e 3 cannibali che vogliono attraversare un fiume con una barca che può contenere al massimo 2 persone. I missionari su ogni lato del fiume non devono trovarsi in minoranza altrimenti verrebbero mangiati.

Per risolvere impostiamo un possibile stato con una tripla con il numero dei missionari e cannibali sul lato iniziale (l'altro si ottiene per differenza) e un booleano per codificare la posizione della barca.

La funzione `move` prova tutte le possibili mosse e tramite `extend` andiamo ad aggiungere una mossa alla storia delle mosse avendo cura di non generare cicli infiniti.

La funzione `move` prova a spostare persone compatibilmente con la capienza della barca e tramite `newState` permette di generare solo stati plausibili che rispettano il vincolo di maggioranza dei missionari ($m \leq c \wedge (3 - m) \leq (3 - c) \Rightarrow m = c$).

```
solution = solve [first]
  where
    solve xs@(x:_) = if x==last then xs
                      else solve (extend (move x) xs)

    extend y ys | all (y/=) ys = y:ys

first = newState 3 3 True
last  = newState 0 0 False

newState m c b | valid && safe = (m,c,b)
  where valid = 0 <= m && m <= 3 && 0 <= c && c <= 3
        safe  = m == 3 || m == 0 || m == c
```

```

move (m,c,b) = move' 0 1
              ? move' 0 2
              ? move' 1 0
              ? move' 1 1
              ? move' 2 0

where
  move' i j = if b then newState (m-i) (c-j) False
              else newState (m+i) (c+j) True

```

se si volesse generare la lista di tutte le possibili soluzioni basta (ovviamente)

```
solutions = findall (\x -> solution == x)
```