Linguaggi di Programmazione (A.A. 2006-07)

Tipi co-induttivi in SML
Specifiche del progetto

Introduzione. La sezione "Nozioni preliminari" delle dispense del corso introduce le nozioni di insieme definito *induttivamente* e *co-induttivamente*. Alla prima di queste nozioni si ispira il costrutto `datatype` del linguaggio *Standard ML* (SML). Mediante tale costrutto è possibile definire un tipo "induttivo" T specificando un insieme di costruttori. Questo consente la definizione "per casi" di funzioni su T. Come spiegato nelle dispense, una tale definizione non risulta invece fondata per insiemi co-induttivi, i cui elementi non possono essere analizzati in base alla loro struttura (mascheramento) ma solo "testati" mediante un insieme di *osservatori*. Lo SML realizza una forma di mascheramento mediante il costrutto `abstype`, ma non supporta appieno la nozione di tipo co-induttivo. In particolare, non è possibile in SML definire una funzione su un `abstype` specificando le osservazioni sui valori da essa prodotti. Scopo di questo progetto è quello di estendere lo SML con un costrutto per la specifca di tipi co-induttivi che consenta definizioni di funzioni "per osservazioni".

Sintassi. La sintassi del nuovo costrutto per tipi co-induttivi potrebbe essere:

```
cotype T = F1 to T1 | ... | Fn to Tn ;
```

dove `T` è il tipo che viene definito, e ciascuna clausola `Fi to Ti` specifica un osservatore `Fi:T -> Ti` sui valori di `T`. Data una tale definizione, il programmatore può definire funzioni per osservazioni mediante la sintassi:

```
fun F is F1(F(x)) = v1
        | ...
        | Fn(F(x)) = vn ;
```

Specifiche.

1. Definire la sintassi astratta di un sottoinsieme minimo (ma significativo) *Fun-co* dello SML (il nome suggerisce una estensione del frammento *Fun*) che includa i costrutti sopra specificati.

2. Scrivere un traduttore *Fun-co* → SML che, ricevendo in input un file contenente un programma nel linguaggio *Fun-co* restituisca in output codice SML *eseguibile*. L'esecuzione deve produrre i risultati attesi secondo la semantica dell'SML e dei tipi co-induttivi.

Suggerimenti.

- *Lex* e *Yacc* sono tecnologia standard per generare traduttori. Consultate la pagina `http://dinosaur.compilertools.net/`.
- Una idea per implementare i tipi co-induttivi è quella di far uso del *sistema dei moduli* di SML. In particolare, la definizione di un tipo co-induttivo potrebbe essere tradotta nella definizione di una *segnatura*. I valori del tipo sarebbero in tal caso *strutture*, che verrebbero manipolate da funzioni (definite per osservazioni) implementate come *funtori*.

  È solo un suggerimento...

Buon lavoro!