ANDREA TOSTI - CORSO B - Mat n. 518111

Relazione secondo progetto v.1, 15 dicembre 2016

Scelte progettuali:

Dato che Etup e Pipe sono tuple di espressioni, mi è sembrato ragionevole fare una distinzione tra i due:

eval (**Etup**) prende una tupla e la espande senza restituire un valore numerico/booleano, mostrando quindi la sequenza di una funzione composta

eval (**Pipe**) prende una tupla, la espande, e la valuta fino ad ottenere un risultato numerico/booleano.

eval (**ManyTimes(n, f)**) restituisce un valore che è l'applicazione consecutiva della funzione, per n volte, su un parametro scelto.

In particolare ho implementato due funzioni :

tupleToSeq per trasformare una tupla in una sequenza di sole Seq :

Seq(Den(f), Seq(Den(g), ..., Seq(Den(z), IntTup(n)/BoolTup(b))

seqToApply per trasformare una sequenza di sole Seq in una concatenazione di sole Apply :

Apply(Den(f), Apply(Den(g), ..., Apply(Den(z), CstTrue/CstFalse/CstInt(n))

Un caso particolare è avere *Seq(Pipe(a), b)* o *Seq(Etup(a), b)*, dove decido di ignorare il parametro b, che è una potenziale sequenza, perché mi aspetto di trovare il parametro all'interno della Pipe/Etup.

L'ambiente è implementato con una lista di coppie, dove la prima componente contiene il nome di una funzione mentre la seconda contiene una struttura dati di tipo Chiusura con tre parametri per una funzione non ricorsiva oppure una ChiusuraRicorsiva con quattro parametri per una funzione ricorsiva.

L'applicazione di una funzione, *Apply(Den(f), eArg)* è valutata controllando dapprima che il nome della funzione f sia associata nell'ambiente ad una chiusura di funzione. Il parametro attuale è valutato nell'ambiente al momento della chiamata ottenendo un valore aVal e il body della funzione è valutato nell'ambiente di dichiarazione esteso con l'associazione del parametro formale.

In allegato ho messo anche dei testcase adatti a testare le situazioni comuni.