Progetti di Paradigmi

dott. Ivano Salvo
Università di Roma "La Sapienza"
email: salvo@dsi.uniroma1.it

Anno Accademico 2002-03

Il progetto di Paradigmi consiste nell'implementazione in LISP di *uno* dei due seguenti moduli. La data di consegna è il **3 giugno 2003**.

1. (MicroLisp)

   (a) Implementare una funzione LISP `mylisp` che attiva un micro-interprete LISP interattivo. Il linguaggio interpretato dal micro-interprete dovrà almeno contenere le espressioni: `mydefun, mysetq, myquote, myif`, le costanti numeriche e `mynil, mytrue`, e i predicati `mynull, myatom, mynumberp, mylistp`.

   Facoltativi sono: `mylet, mylambda, ...` e le funzioni predefinite sulle liste.

   (b) (**Facoltativo**) Implementare più di un interprete, tra le quattro possibilità offerte dalle scelte: lazy/eager evaluation e static/dynamic binding, esibendo per ciascuna coppia di interpreti almeno un esempio di espressione che viene valutata in modo diverso.

2. (Type Inference)

   (a) Implementare in LISP un algortimo di type inference per F1, che interattivamente accetta una espressione di F1 e restituisce in output il tipo principale. I termini di F1 vengono scritti in sintassi LISP-like, e cioè sono termini generati dalla seguente sintassi:

   $$exp = x \mid (\mathsf{lambda}\ x\ exp) \mid (exp\ exp)$$

   (b) (**Facoltativo**) Implementare l'algoritmo $\mathcal{W}$ di Milner di type inference sul core calculus di ML, cioè su F1 arricchito con un'espressione:

   $$exp = (\mathsf{let}\ (x\ exp)\ exp)$$

   (c) (**Facoltativo**) Estendere la sintassi con la possibilità di avere definizioni simultanee (da trattarsi come ML tratta le definizioni legate dalla parola chiave `and`:

   $$exp = (\mathsf{let}\ ((x_1\ exp_1) \ldots (x_n\ exp_n))\ exp)$$