RELAZIONE PROGRAMMAZIONE II

A.A. 2017-18: Secondo Progetto Intermedio (OCaml)

Guidi Elena 537336

Per lo sviluppo del progetto sono stati aggiunti alcuni tipi oltre a quelli richiesti nella specifica del progetto:

- RecVFun (type evT) che indica una funzione ricorsiva valutata.
- Tree (type exp e type evT) che indica un albero.
- VTree = VEmpty o VNode (type evT) che indica un albero valutato (con il suo ambiente).

I controlli di tipi basici (boolean e int nel nostro caso) sono svolti dalla funzione typecheck mentre il controllo di tipo degli alberi è svolto in modo ricorsivo (controllando ogni volta il sottoalbero destro e sinistro) dalla funzione isTree. Typecheck e isTree vengono chiamate da funzioni per determinare che comportamento far assumere a quest'ultime in base al risultato della chiamata (risultato offerto a run time implementando un type checker dinamico).

Il controllo degli errori sugli input da passare alle funzioni viene principalmente svolto da eval (con l'aiuto delle funzioni sopra citate) . Ogni funzione ha comunque un controllo dei risultati di computazioni e chiamate di altre funzioni.

ApplyOver, Update e Select richieste dalla specifica sono implementate mediante chiamate alle rispettive funzioni apply_over_tree, update_tree e select_in_tree (tutte e tre ricorsive) assegnando previamente la funzione input all'ambiente e passando quindi una funzione valutata (prima della chiamata si controlla anche la correttezza della funzione, ovvero che sia ternaria).

Nella seconda parte del file si trova la batteria di test che compre la creazione di alberi (vuoti e non) e la chiamata alle tre funzioni implementate. I test cercano di comprire tutti i vari casi d'uso ovvero tutti i comportamenti del codice implementato.

Particolare attenzione è stata posta nella scelta dei nomi dei test e delle variabili del codice, si è cercato di dare nomi significativi cercando di dare il massimo della leggibilità al codice.