

RELAZIONE PROGETTO OCAML PROGRAMMAZIONE 2

Il progetto parte dall'interprete del linguaggio didattico funzionale visto a lezione, aggiungendo le espressioni richieste riguardanti i dizionari e una valutazione che utilizzi lo scoping dinamico delle espressioni:

- Dict, Insert, Get, Remove, Clear, ApplyOver per quanto riguarda i dizionari
- DynamicFun, DynamicFunCall, DynamicLetRec, DynamicApplyOver per quanto riguarda lo scoping dinamico
- È stato inoltre necessario aggiungere il tipo String al linguaggio funzionale, che prima prevedeva solo l'uso di interi e booleani, anche se non sono state aggiunte operazioni per manipolarle. Il tipo String viene utilizzato solo nei dizionari per aggiungere coppie che contengono, appunto, questo valore

Di conseguenza è stato necessario aggiungere anche dei tipi esprimibili:

- Dict of (ide * exp) list e DictVal of (ide * evT) list per l'uso dei dizionari
(ide * exp) denota le coppie identificatore-valore presenti appunto nel dizionario
- DynamicFunVal of ide * exp e DynamicRecFunVal of ide * (ide * exp) per l'uso di funzioni con scoping dinamico

Queste non hanno, al contrario delle funzioni con scoping statico, un evFun, che denotava l'ambiente non locale della funzione (fDecEnv). Infatti le funzioni prendono come tale ambiente, la valutazione dell'espressione stessa

VALUTAZIONE DELLE NUOVE ESPRESSIONI

Durante la valutazione si controlla che i tipi siano quelli previsti. Se così non fosse l'interprete dà errore

- DIZIONARI valutazione del tipo: dict deve essere un dizionario
 - DICT: per valutare Dict(list) si utilizza una funzione ricorsiva che, presa una lista come input, restituisce una lista di coppie
 - CLEAR: la valutazione restituisce un dizionario vuoto
 - INSERT: la valutazione di insert utilizza una funzione ricorsiva che, presi in input una lista, un nome e un valore, controlla che nella lista non siano presenti altri valori associati al nome, in qual caso restituisce un errore, altrimenti associa il nome al valore inserendoli nella lista
 - GET: la valutazione del get utilizza una funzione ricorsiva la quale presi in input una lista e un nome, verifica che nella lista sia presente un'associazione al nome. Se è presente restituisce il valore, altrimenti ritorna errore
 - REMOVE: la valutazione di remove utilizza una funzione ricorsiva la quale, presi in input una lista e un nome, elimina il nome dalla lista se presente, altrimenti non fa nulla
 - APPLYOVER la valutazione di applyover fa uso di due funzioni, una non ricorsiva e una ricorsiva.
La prima, presi in ingresso una funzione e un valore, verifica che sia una funzione e la applica al valore passato. Se dà errore di tipo cattura l'eccezione e restituisce il valore passato, altrimenti aggiorna il valore con quello restituito dalla funzione. La

seconda funzione, presi in input una lista di coppie nome-valore e una funzione, ad ogni elemento della lista applica la funzione ad ogni valore mediante la prima funzione, restituendo la lista aggiornata

- SCOPING DINAMICO
 - DYNAMICFUN
 - DYNAMICFUNCALL
 - DYNAMICLETREC
 - DYNAMICAPPLYOVER

Queste quattro funzioni non sono presenti se si utilizza eval e una valutazione mediante eval restituisce, appunto, un errore.

Queste possono essere valutate solo mediante rteval. Questa funzione differisce da quella originale solo per queste funzioni, e solo per l'ambiente della funzione, che in questo caso è quello della valutazione dell'espressione intera, non viene passato come input alla funzione. In rteval non è possibile invece una valutazione della versione statica delle funzioni appena elencate

TEST

Le batterie di test presenti alla fine dell'interprete sono divise in 3 parti:

quelle per testare l'interprete di base, quelle per testare i dizionari e quelle per testare lo scoping dinamico, facendo vedere le differenze con lo scoping statico avendo delle espressioni simili tra loro.

Per compilare ed eseguire il file del progetto si utilizzi il sito <https://try.ocamlpro.com/>

Marcello Matteucci 546273