RELAZIONE PROGETTO PR2

Luca Cataldo

1 Scopo del progetto

1.1 Traccia:

Si consideri un'estensione del linguaggio funzionale che permetta di manipolare come dati primitivi dizionari di elementi, ovvero una collezione di coppie (chiave, valore). Assumiamo che la chiave sia un identificatore. Si definisca la sinstassi concreta del linguaggio e la sinstassi asatratta.Si definisca l'interprete in Ocaml assumendo la regola dello scoping statico.

2 Sintassi concreta

```
Dictionary::= | Item

Item::=Ide:Exp,Item | Ide:Exp

Exp::=Dict |Select(Exp, Ide) |Insert(Exp, Ide, Exp) |Remove(Exp, Ide) |Clear(Exp)
|ApplyOver(Exp, Exp)
```

3 Sintassi astratta

```
Dictionary of item

Insert of exp * ide * exp

Remove of exp * ide

Search of exp * ide

Clear of exp

 ApplyOver of exp * exp

and item = Unbound | KeyValue of ide * exp * item
```

4 Scelte progettuali

Il type checker che ho usato è dinamico.Ho usato un interprete ricostrutito in base alle mie esigenze.
E ogni espressione è stata sviluppata in questo modo:

(i) `Dictionary`: Per la creazione del dizionario ho usato principalmente due funzioni:`evalDict` e `check` le quali rispettivamente in ordine di apparizione eseguono le seguenti operazioni: `evalDict` valuta tutto ciò che gli viene passato e mette tutti questi elementi valutati in una variabile la quale sarà poi usata dalla funzione seguente. `Check` non fa altro che prendere gli elementi valutati da `evalDict` e controllare che non ci siano chiavi ripetute all'interno del dizionario. Nel caso ci fossero chiavi ripetute all'interno del dizionario il risultato finale di tale funzione sarà il sollevamento di una eccezione.

(ii) `Search`: Per quanto riguarda la ricerca di un elemento all'interno del dizionario ho usato una funzione interna `Searching` la quale banalmente cerca tramite dei parametri in ingresso, se tale elmento è presente all'interno oppure no.

(iii) `Insert`: Per quanto rigaurda l'inserimento di un elemento all'intero del dizionario ho usufruito di una funzione `Add` la quale in base a dei parametri in ingresso controlla prima se tale elemento esiste gia all'interno del dizionario, se ciò dovesse rivelarsi esatto allora la funzione non fa altro che sovrascrivere tale elemento con quello gia presente nel dizionario, altrimenti tale elemento se dopo la scansione non si rivela essere prensente nel dizionario essa lo aggiunge.

(iv) `Remove`: Per quanto riguarda l'eliminazione dei un elemento all'interno del dizionario ho usato una funzione `Delete`, la quale non fa altro che prendere dei parametri in entrata, valutare tali parametri cercando se all'interno del dizionario sono presenti elementi che corrispondono a tali parametri,se cosi fosse allora tale elemeto verra rimosso dal dizionario.

(v) `Clear` :Invece per quanto riguarda la pulizia del dizionario non faccio altro che dare come risultato un dizionario che non conterrà nulla

(vi) `ApplyOver`: Per quanto riguarda questa espressione essa ha due parametri, una fuzione e un dizionario. Tale funzione sara applicata a ogni elemento del dizionario per poi restituire un dizionario nuovo nel quale a tutti gli elementi è stata applicata la funzione descritta nel parametro d'ingresso. Per la creazione di questa espressione ho usato una funzione `Apply` che prende in ingresso tre parametri e

5 Scelte sull'implementazione dei metodi