

Leonardo Vona
Matricola 545042
Corso A
Programmazione II – Secondo Progetto Intermedio
06/02/2020

Con lo scopo di estendere il linguaggio funzionale didattico con la manipolazione di dizionari. Per fare ciò si è quindi estesa la sintassi astratta per permettere la gestione di dizionario, aggiungendo alle espressioni già definite:

```
type exp =  
  ...  
  | Fun2 of ide * ide * tval * tval * exp  
  | Apply2 of exp * exp * exp  
  | Edict of dict  
  | Insert of ide * exp * exp  
  | Delete of ide * exp  
  | Has_key of ide * exp  
  | Iterate of exp * exp  
  | Fold of exp * exp  
  | Filter of (ide list) * exp  
and dict =  
  | Empty  
  | Elem of ide * exp * dict;;
```

E aggiungendo inoltre ai tipi esprimibili:
type evT =

```
  ...  
  | Fun2Val of ide * ide * exp * evT env  
  | Dict of dict;;
```

dove Fun2Val, Fun2 e Apply2 sono necessari per implementare funzioni con due parametri (necessarie per la Fold).

Nota: il tipo tval è stato introdotto per la gestione del type checker statico.

Quindi si è esteso l'interprete con le funzioni Insert, Delete, Has_key, Iterate, Fold, Filter e Apply2.

Il type checker statico è stato implementato come un interprete separato che restituisce il tipo di una espressione.

Una scelta di progetto è stata quella di non imporre limiti sul tipo di un dizionario, quindi un dizionario può contenere espressioni eterogenee. Nel caso della Fold, la quale richiede che gli elementi del dizionario siano dello stesso tipo, il controllo viene fatto al momento dell'applicazione della funzione che, nel caso riscontri un errore di tipo, genera un'eccezione.