

PROGRAMMAZIONE II - A.A. 2019 - 20

Secondo Progetto - sessione autunnale

Studente: Simone Ianniciello

Matricola: 581201
Corso A

Obiettivo:

Implementare un tipo Dizionario nel linguaggio didattico presentato a lezione.

Scelte di implementazione:

- Il tipo *Edict* è stato implementato in modo ricorsivo con il modulo *Val(id, e, next)* dove *id* ed *e* rappresentano la coppia *chiave -> valore* e *next* il prossimo valore nel dizionario; la lista è terminata dal valore *Empty*.
- Il tipo dei valori all'interno del dizionario deve essere omogeneo
- La funzione *Insert(id, e, d)* controlla che *id* non sia già presente nel dizionario e che *e* sia dello stesso tipo dei valori già in esso; se 'le condizioni vengono rispettate ritorna il dizionario originale con l'aggiunta di (*id*, *e*)
- *Delete(id, d)* ritorna $d \setminus \{(id, -)\}$
- La funzione *Has_key(id, d)* ritorna *Bool true* se $(id, -) \in d$
- *Iterate(f, d)* applica, se possibile, la funzione *f* a tutti i valori in *d*
- La funzione *Fold(f, d, a)* applica ricorsivamente la funzione *f* all'*n*-esimo valore del dizionario e al risultato della chiamata *n-1-esima* di *fold*; l'accumulatore parte da *a* (il risultato della chiamata 0 è *a*)
- *Filter(is, d)* ritorna una porzione del dizionario *d* con le sole chiavi contenute in *is*

Typechecker statico:

Il typechecker statico deve controllare la validità del codice scritto prima di valutarlo.

Per fare ciò esegue un controllo ad albero sugli argomenti del comando da eseguire e si accerta che rispettino il tipo richiesto dalle funzioni ritornandone il tipo.

TestSet:

Il file *test.ml* contiene una batteria di test per verificare il corretto funzionamento del typechecker (statico e dinamico)