**Relazione del progetto in OCaml del corso di Programmazione 2**

**A.A. 2019/20**

**SIMONE IOZIA N.549108**

**Obiettivo**

Realizzare un’estensione del linguaggio didattico funzionale che permetta di manipolare il tipo del *dizionario*.

**Scelte d’implementazione**

Il linguaggio didattico è esteso con Dict, che rappresenta un dizionario implementato come una lista di coppie chiave-valore ((ide\*exp)list).

Inoltre sono aggiunte le operazioni primitive che supportano i dizionari: Insert(aggiunta di una coppia non già presente), Delete(rimozione di una coppia esistente), Has\_Key(controllo della presenza di una coppia), Iterate(applicazione di una funzione a tutti i valori delle chiavi), Fold(applicazione di una funzione che prevede l’uso di un accumulatore che salva via via tutti i risultati temporanei) , Filter(filtrare dal dizionario solo le coppie la cui chiave è presente in una lista di stringhe).

I valori di un dizionario devono essere omogenei, in modo particolare al fine di applicare le funzioni Iterate e Fold senza alcun problema di tipo; per far ciò, l’omogeneità di un dizionario è controllata da un typechecker dinamico che controlla i tipi dei valori quando si crea un dizionario, o quando si aggiunge una nuova coppia o alla chiamata delle funzioni sopra dette.

Un dizionario è caratterizzato da coppie chiave-valore dove la chiave è unica; per questo motivo, quando viene creato un dizionario con chiavi duplicate o aggiunta una nuova coppia la cui chiave è già presente, si lancia un’eccezione.

Per l’implementazione della funzione Fold, che prevede due parametri (un valore e un accumulatore), il linguaggio didattico è esteso anche con FunBin, che definisce le funzioni binarie (ide\*ide\*exp).

**Test**

Appena sotto l’interprete, si può trovare una batteria di test che provano il funzionamento del dizionario e di tutte le sue operazioni, anche in presenza di situazioni non consentite(lancio di un’eccezione). Il tutto è valutato all’interno di un ambiente (env) inizialmente vuoto.