Satta Marcello - 580495 - B 14/01/2020

RELAZIONE SECONDO PROGETTO PROGRAMMAZIONE II

Scopo di questo documento è fornire la documentazione relativa alla progettazione e realizzazione del secondo progetto proposto dai docenti del Corso di Programmazione II, che ha lo scopo di applicare i concetti e le tecniche di programmazione OCaml relative agli interpreti esaminate durante le lezioni.

ISTRUZIONI PER ESEGUIRE IL PROGRAMMA

Per eseguire il programma è necessario copiare e incollare sull'interprete OCaml il file ProgettoPR2INTERPRETE.ml presente nella cartella: …SecondoProgettoPR2ZIP/ SecondoProgettoPR2.

Per eseguire la batteria di test copiare e incollare sull'interprete OCaml il file ProgettoPR2TEST.ml presente nella medesima cartella.

SCELTE PROGETTUALI

Ho scelto di rappresentare i dizionari come una lista di coppie chiave-valore con le seguenti proprietà:

- i dizionari sono omogenei, ovvero contengono valori di uno e un solo tipo;
- il tipo dei valori presenti all'interno dei dizionari può essere Int o Bool;
- i dizionari non ammettono chiavi duplicate;
- ogni chiave è associata ad uno ed un solo valore.

Regole operazionali

Dictionary: il costruttore prende come parametro un'espressione di tipo Dictionary e valuta i valori presenti all'interno del dizionario, restituendo una lista di coppie chiave-valore

((key * evT) list). Viene controllato che il dizionario non presenti chiavi duplicate e che sia omogeneo.

Insert: questa operazione inserisce una coppia chiave-valore all'interno del dizionario su cui viene chiamata, verificando che il tipo del valore da inserire e i tipi dei valori del dizionario corrispondano. L'inserzione avviene correttamente se la chiave dell'elemento da inserire non è già presente all'interno del dizionario.

Delete: questa operazione rimuove una coppia chiave-valore dal dizionario, riconoscendola tramite il parametro rappresentante la chiave. Precondizione affinché l'operazione abbia successo è che il dizionario sia omogeneo e la coppia chiave-valore identificata sia presente nel dizionario.

HasKey: questa operazione controlla la presenza della coppia chiave-valore identificata dalla chiave (passata come parametro) all'interno del dizionario. Il dizionario deve essere omogeneo e non ci devono essere chiavi duplicate.

Iterate: questa operazione applica la funzione passata come parametro ad ogni valore del dizionario. Il dizionario deve essere omogeneo e non ci devono essere chiavi duplicate. Inoltre il tipo del valore restituito dalla funzione deve corrispondere al tipo dei valori presenti all'interno del dizionario.

Fold: questa operazione applica la funzione passata come parametro ad ogni valore del dizionario e restituisce la somma di tutti i valori aggiornati. Il dizionario deve essere omogeneo e non ci devono essere chiavi duplicate. Inoltre il tipo del valore restituito dalla funzione deve corrispondere al tipo dei valori presenti all'interno del dizionario (solo dizionari di tipo Int).

Filter: questa operazione filtra la lista, eliminando dal dizionario tutte le chiavi che non sono presenti nella lista di chiavi passata come parametro. Il dizionario deve essere omogeneo e le chiavi non devono avere duplicati.

Type checker

Il type checker esteso per i dizionari controlla che ogni dizionario sia omogeneo e che non ci siano chiavi duplicate. Viene chiamato alla valutazione di ogni dizionario, nel costruttore, nella Iterate e nella Fold.