

# Relazione per il progetto di Programmazione 2

Marco Colavita - secondo progetto intermedio

June 25, 2020

## 1 Introduzione

Scopo di tale progetto è quello di estendere l'interprete didattico fornito dal docente, in modo da gestire una collezione di coppie  $\langle \textit{chiave}, \textit{valore} \rangle$  che rappresentano un dizionario di elementi non omogenei, dove la chiave è un identificatore.

## 2 Sintassi Concreta

Viene definita la sintassi concreta qui di seguito:

$$\textit{Dict} := \epsilon | (\textit{Ide}, \textit{Exp}); \textit{Dict} \quad (1)$$

$$\textit{Insert} := (\textit{Ide}, \textit{Exp}, \textit{Dict}) \quad (2)$$

$$\textit{Delete} := (\textit{Dict}, \textit{Ide}) \quad (3)$$

$$\textit{HasKey} := (\textit{Ide}, \textit{Exp}) \quad (4)$$

$$\textit{Iterate} := (\textit{Exp}, \textit{Exp}) \quad (5)$$

$$\textit{Fold} := (\textit{Exp}, \textit{Exp}) \quad (6)$$

$$\textit{Filter} := (\textit{IdeList}, \textit{Dict}) \quad (7)$$

## 3 Scelte progettuali

Gli operatori richiesti dal docente sono stati implementati secondo il seguente criterio:

**Dict** - viene utilizzato per la creazione di un nuovo dizionario. Un dizionario alla sua creazione potrebbe essere vuoto oppure costruito con un insieme di coppie  $\langle \textit{chiave}, \textit{valore} \rangle$ . La sua valutazione restituisce le coppie  $\langle \textit{ide}, \textit{evT} \rangle$  a significare che i tipi di elementi all'interno del dizionario non sono omogenei, quindi di qualsiasi tipo.

**Insert** - che con una coppia  $\langle \textit{chiave}, \textit{valore} \rangle$  restituisce un nuovo dizionario, identico al precedente con l'aggiunta di tale coppia (siccome questi ultimi sono

immutabili). La chiamata di tale operatore può terminare con successo in caso di chiave non presente nel dizionario, o con errore, nel caso in cui la chiave da inserire compare all'interno del dizionario.

**Delete** - con la quale viene restituito un nuovo dizionario, eliminando la coppia  $\langle ide, exp \rangle$  se è presente nel dizionario passato. Nel caso in cui l'identificatore non è presente nel dizionario, viene restituito un dizionario identico.

**HasKey** - che prendendo come parametro l'identificatore e il dizionario, restituisce true se la chiave è presente in quest'ultimo, false altrimenti.

**Iterate and Fold** - la prima applica una funzione  $f$  a tutte le coppie  $\langle ide, exp \rangle$  nel dizionario, restituendo un nuovo dizionario aggiornato. La Fold invece applica una funzione  $f$  *sequenzialmente* anch'essa a tutte le coppie  $\langle ide, exp \rangle$  nel dizionario, restituendo però un unico risultato.

**Filter** - viene utilizzata per restituire un nuovo dizionario. Preso come argomento una lista di identificatori, tale metodo rimuove le coppie  $\langle ide, exp \rangle$  se l'identificatore non compare nella lista.

## 4 Istruzioni per l'esecuzione

Nel file *progetto.ml* sono presenti una serie di dichiarazioni di espressioni seguite dalla loro valutazione, con un commento che riporta il risultato da stampare. Nel file sono state testate tutte le operazioni del dizionario.