



UNIVERSITÀ DI PISA

# Corso di laurea in Informatica

## RELAZIONE DEL PROGETTO INTERMEDIO DEL CORSO DI PROGRAMMAZIONE II

**Studente:** Lorenzo Massagli

**Matricola:** 579418

**Corso:** A

### **INSEGNANTI:**

Prof. Gianluigi Ferrari

Prof.essa Francesca Levi

**Obiettivo:** Estendere il linguaggio didattico funzionale in modo che permetta di manipolare i "Dizionari"

Anno Accademico 2019/2020

# Capitolo 1

## 1.1 Istruzioni per eseguire il codice

Per eseguire il codice si può utilizzare il sito “<https://try.ocamlpro.com/>” che permette di eseguire codice scritto nel linguaggio OCAML direttamente da browser.

## 1.2 Dettagli progettuali

Il linguaggio didattico è stato esteso con i Dizionari definendo essi ricorsivamente tramite un valore “Empty” che indica che il dizionario è vuoto, ed un valore “Val” che indica la coppia (chiave, valore) attuale e le altre chiavi contenute nel dizionario.

I valori nel dizionario non sono omogenei, ma la loro omogeneità viene controllata tramite un typechecker dinamico, alla valutazione da parte dell’interprete delle operazioni “fold” e “iterate”. Nel caso il dizionario non fosse omogeneo alla valutazione di una di queste due operazioni, viene generato un errore.

Per definizione di dizionario, essendo che usa le chiavi come accesso ai valori, le chiavi devono essere univoche, quindi c’è un controllo di doppia chiave sia quando si valuta un dizionario, sia quando viene valutata una Insert.

Infine, è stato progettato di estendere il linguaggio in modo da gestire funzioni binarie. Questo perchè la fold prende in ingresso 2 parametri che sono l’accumulatore ed un valore del dizionario.

L’interprete deve essere in grado in caso di una fold di valutare l’applicazione della funzione binaria con il giusto accumulatore in base all’operazioni nel body della funzione.

Il typechecker è stato progettato in modo che possa fare il typechecking dinamico di tutti i tipi di evT. Nel caso del Dizionario, controlla se è un dizionario e se tutti i tipi all’interno sono omogenei.

## 1.3 Dettagli di realizzazione

I dizionari sono stati realizzati attraverso una definizione ricorsiva data da:

- Empty (Valore che indica che il dizionario è vuoto)
- Val (ide \* exp \* dictarg)

Ide = key      exp = valore      dictarg = resto delle coppie (ide, valore) nel dizionario

Sono state estese inoltre le operazioni associate ad un dizionario:

Insert, Delete, Has\_key, Iterate, Fold, Filter.

Per estendere l’operazione di Fold, la quale calcola il valore ottenuto applicando la funzione sequenzialmente a tutti gli elementi del dizionario, è stato esteso ulteriormente il linguaggio con le funzioni binarie.

Nel caso della fold, la funzione binaria deve prendere un accumulatore e un valore, ed applicare la funzione ad ogni valore salvando i risultati intermedi nell’accumulatore.

Le funzioni binarie sono state estese così:

- FunAcc (ide \* ide \* exp) → Prende due parametri e il body della funzione;
- FunCallAcc(exp\*exp\*exp) → Prende in ingresso i valori dei due parametri e applica la funzione.

In più sono stati aggiunti ai tipi esprimibili  $evT$ :

- Dictvalues of ( $ide * evT$ ) list  $\rightarrow$  Tipo esprimibile dato dalla valutazione di un dizionario (Una lista che contiene le coppie ( $key*value$ )).
- FunValAcc of ( $ide * ide * exp * evT\ env$ )  $\rightarrow$  Chiusura data dalla valutazione di una funzione binaria FunAcc. La chiusura è caratterizzata da due parametri (accumulatore e value del dizionario), il body della funzione e l'ambiente al momento della dichiarazione.

L'interprete è stato modificato in aggiunta all'estensione del linguaggio.

Per valutare un dizionario viene utilizzata la funzione ricorsiva `evalList` che controlla se ci sono duplicati nel dizionario e genera l'insieme delle coppie ( $key * val$ ).

Per valutare l'operazione di Insert viene applicata una funzione ricorsiva che verifica se esiste già la chiave associata al valore che vogliamo inserire. Nel caso ci fosse genera un errore, sennò la inserisce infondo al Dizionario.

Per valutare l'operazione di delete viene applicata una funzione ricorsiva che elimina l'elemento nel dizionario che ha la chiave uguale alla chiave passata nell'operazione.

Per valutare l'operazione di haskey viene applicata una funzione ricorsiva che verifica se esiste almeno una chiave che è uguale alla chiave passata nell'operazione.

Per valutare l'operazione Iterate, bisogna utilizzare il typechecker dinamico che va a controllare se tutti gli elementi del dizionario sono omogenei. Nel caso non fossero omogenei, viene dato un errore. In caso contrario viene applicata una funzione che valuta ogni elemento del dizionario nella funzione passata nell'operazione, restituendo un nuovo dizionario con i valori ottenuti come risultato della funzione. Viene valutata dall'interprete l'operazione "FunCall" valutando il valore del dizionario all'interno del corpo della funzione.

Per valutare l'operazione di fold, bisogna utilizzare il typechecker dinamico che va a controllare se tutti gli elementi del dizionario sono omogenei. Nel caso non fossero omogenei, viene dato un errore. Nel caso fossero tutti omogenei, bisogna passare alla funzione il corretto accumulatore che dipende dalle operazioni contenute nel body della funzione. Ogni valore del dizionario e dell'accumulatore viene valutato nella valutazione dell'operazione "FunCallAcc" che valuta i valori nel body della funzione e salva il risultato intermedio nell'accumulatore.

Per valutare l'operazione di filter, bisogna applicare una funzione ricorsiva al dizionario, la quale restituisce un nuovo dizionario con le coppie ( $key,value$ ) le quali hanno le chiavi uguali ad almeno una chiave della lista passata all'operazione di filter.

Il typechecker come già detto, è stato realizzato in modo che possa controllare a run-time tutti i tipi di  $evT$  e soprattutto che possa controllare l'omogeneità del dizionario, in modo da poter valutare le operazioni di fold e iterate.

## 1.4 Batteria di test

Nella batteria di test sono stati inseriti dei test sufficienti a testare tutti gli operatori aggiunti dal dizionario.

Oltre ai test che verificano che ogni operazione venga valutata nel modo corretto, sono stati inseriti test che vanno a verificare i vari errori dati dalla valutazione dell'interprete, ad esempio:

- Chiave duplicata nel dizionario;
- Tipi non omogenei nella valutazione di un operazione di fold/iterate.

Infine sono stati aggiunti alcuni test che vanno a testare il typechecker dinamico.