



UNIVERSITÀ DEGLI STUDI DI PISA

Corso di Programmazione II

Corso A

A.A 2019/2020

Documentazione

Secondo Progetto-Ocaml

*Autore:*

Lorenzo Angeli

539036

L'ambiente scelto per questo linguaggio di programmazione è una lista, il che comporta un pattern matching di un identificatore (descritto come stringa) per ogni bind al suo interno.

I tipi di dato esprimibili dal linguaggio sono quelli essenziali per il funzionamento del dizionario:

1. Int
2. Booleani
3. Stringhe

## Sintassi Concreta

### Sintassi del Let:

$\text{Let Iden} = \text{Espr in BodyLet}$

Viene valutato il body del Let nell'ambiente, che contiene un bind tra l'identificatore e la valutazione dell'espressione.

### Sintassi del Let Ricorsivo:

$\text{LetRec FunIden Param} = \text{Body\_Fun in Let\_Body}$

Viene valutato il Body del Let all'interno della funzione nell'ambiente, contenente l'identificatore della funzione ricorsiva, i suoi parametri e il suo body.

## Operazioni di base:

- **Moltiplicazione (Prod):**  $\text{Int1} * \text{Int2} = \text{Int3}$
- **Somma (Sum):**  $\text{Int1} + \text{Int2} = \text{Int3}$
- **Differenza (Diff):**  $\text{Int1} - \text{Int2} = \text{Int3}$
- **Uguaglianza (Eq):**
  - $\text{Int1} == \text{Int2} = \text{Boolean3}$
  - $\text{Boolean1} == \text{Boolean2} = \text{Boolean3}$
  - $\text{String1} == \text{String2} = \text{Boolean3}$
- **Not (Minus):**
  - $\text{Int1} = -\text{Int2}$
  - $-\text{Int1} = \text{Int2}$
  - $\text{Boolean1} = \neg \text{Boolean2}$
- **And (And):**  $\text{Boolean1 AND Boolean2} = \text{Boolean3}$
- **Or (Or):**  $\text{Boolean1 OR Boolean2} = \text{Boolean3}$
- **IsZero (IsZero):** if  $\text{Int1} == 0$  then true else false
- **Minimo (Min):** if  $\text{Int1} < \text{Int2}$  then true else false

- **IfthenElse(Ifthenelse):** if esp then C1 else C2
  - Valuta l'espressione esp, se vera esegue il blocco di codice C1, altrimenti C2

### **Operazioni Dictionary:**

Per il Dizionario sono state aggiunte le operazioni elencate di seguito:

- **Insert:** inserisce coppia <chiave,valore> nel dizionario
- **Remove:** prende come argomenti il dizionario e una chiave k e rimuove la coppia <k,valore> dal dizionario
- **Select:** prende come argomenti il dizionario e una chiave k e restituisce il valore associato a quella chiave k
- **Clear:** prende in ingresso un dizionario e restituisce un dizionario vuoto
- **Has\_key:** prende come argomenti un dizionario e una chiave k e controlla l'esistenza di quella chiave k all'interno del dizionario restituendo true o false di conseguenza.
- **Iterate:** prende come argomenti una funzione e un dizionario e applica la funzione a tutti gli elementi del dizionario
- **Fold:** prende come argomenti una funziona e un dizionario e applica la funzione sequenzialmente a tutti gli elementi del dizionario e restituisce il valore ottenuto