

## 2° Progetto Intermedio - Relazione

Il progetto consiste in un'estensione del linguaggio didattico visto a lezione che permetta di manipolare insiemi. Un insieme è una collezione di valori, di tipo omogeneo, non ordinati, che non contiene valori duplicati.

Gli elementi di un insieme possono avere tipo **Int**, **Bool** o **String**. Nel caso di insieme di Bool, la caratteristica di non contenere valori duplicati implica che l'insieme possa contenere al massimo due elementi (*CstTrue* e *CstFalse*). Inoltre, per quanto concerne le funzioni di Min e Max, il valore di CstTrue è considerato 1, mentre quello di CstFalse 0.

Si è deciso di non supportare insiemi di funzioni poiché, nonostante nel linguaggio esse siano considerate dei normali valori, sarebbe assai complesso controllare la presenza di funzioni duplicate; infatti, matematicamente due funzioni  $f$  e  $g$  sono la stessa funzione se e solo se:

- $f$  e  $g$  hanno lo stesso dominio;
- $f$  e  $g$  hanno lo stesso codominio;
- $\forall x : f(x) = g(x)$ .

Verificare ciò implica una computazione ovviamente ingestibile.

Si sono prese in considerazione soluzioni alternative, ma anch'esse hanno dimostrato di avere notevoli criticità: per esempio, se come criterio di uguaglianza tra funzioni considerassimo il corpo della funzione (*fbody*), allora due funzioni  $f$  e  $g$  così dichiarate (in notazione matematica standard):

1.  $f(x) = x + 1$

2.  $g(y) = y + 1$

sarebbero considerate come distinte, quando trattasi invece della medesima funzione con un argomento di nome diverso.

Il linguaggio didattico di partenza supportava **If-Then-Else** che restituivano risultati di tipo diverso a seconda del ramo.

Questo comportamento è stato modificato introducendo un controllo tramite

la funzione ausiliaria *same.type*, rendendolo in linea con quello di molti altri linguaggi.

Ciò permette alla funzione **Map** di inferire, tramite la funzione ausiliaria *find.type*, il tipo dell'insieme risultante.

All'interno del progetto si fa uso (in poche occasioni) di funzioni interne al module *List*.

Per eseguire la batteria di test è necessario eseguire in un ambiente REPL prima il codice interno al file *progetto.ml* e poi quello interno al file *test.ml*.

*Luca Lombardo*  
Mat. 546688  
13/12/2020