

# Secondo progetto intermedio. 2020/21

Marco Marinò - 596440 - Corso A

## Esecuzione del progetto:

Per eseguire il progetto è sufficiente eseguire nel terminale:

```
ocaml progetto.ml
```

## Scelte d'implementazione:

Si estende il linguaggio didattico presentato a lezione introducendo la possibilità di rappresentare insiemi. Per insieme dunque si intende una collezione di dati che devono avere lo stesso tipo e non devono contenere duplicati. Si è scelto di implementare i Set mediante l'utilizzo del tipo di dato list nativo di OCaml. Con tale sistema di implementazione tuttavia, si incorre in alcune problematiche:

- È possibile creare una lista di espressione che risultano in tipi non omogenei, rendendo necessario il controllo sulla congruenza del tipo degli elementi.
- Bisognerà controllare la presenza di duplicati

## Of, Empty, Singleton

I tre costruttori di set. Il primo permette di creare un set a partire da una lista passata come parametro ed un tipo che rappresenta il tipo del set. Of controlla in primo luogo (come anche per Singleton) che ogni elemento in lista abbia il tipo indicato come parametro ed in secondo luogo (solo Of) controlla la presenza di duplicati. In altre parole, quando operiamo sui set sappiamo già a priori che non vi sono duplicati ed errori di tipo.

## Union, Intersection, Difference, Subset, Contains, Insert, Remove, IsEmpty

In queste operazioni si controlla che i tipi dei due insiemi (nel caso di Unione, Int., Diff. e Subset) siano gli stessi, failwith nel caso contrario. Stessa cosa per Contains, Insert, Remove in cui controlliamo che il parametro abbia lo stesso tipo (la Insert controlla anche la presenza di duplicati con la funzione esterna set\_insert, Remove controlla che l'elemento da eliminare esista nel set con set\_remove, stessa cosa con set\_subset).

## MinValue, MaxValue

Si controlla se l'insieme è vuoto e poi si itera la lista per trovare min e max.

## Forall, Exists, Filter, Map

Un dettaglio degno di nota di queste funzioni è sicuramente che per assicurare internamente la correttezza del set passato come parametro lo si valuta, così in caso di errore non andrebbe a buon fine. Nelle prime tre si suppone che le funzioni passate come parametro restituiscano un tipo booleano, dato che non possiamo verificarne la natura a priori. Per quanto riguarda la Map è necessario controllare che il set di ritorno rispetti tutte le proprietà dei set. Siccome non possiamo sapere a priori il tipo di ritorno della funzione dobbiamo effettuare i controlli a posteriori. L'eventuale presenza di duplicati si verifica mentre si calcola la funzione passata come parametro sugli elementi dell'insieme. Successivamente si controlla che anche tutti gli altri elementi dell'insieme restituito siano dello stesso tipo. Anche qui in caso vi siano delle incongruenze sull'insieme da restituire si lanciano le eccezioni corrispondenti.