

Premessa

Il progetto richiede l'estensione del linguaggio didattico funzionale presentato a lezione, in modo che permetta la creazione e la manipolazione di insiemi.

Un insieme è una collezione di valori non ordinati e non duplicati (l'insieme vuoto è consentito).

Nel mio linguaggio, un Set è caratterizzato da una coppia { lista valori – tipo Set } i quali rappresentano rispettivamente:

- I valori dell'insieme non ripetuti e dello stesso tipo
- I valori ammessi all'interno dell'insieme

Tutte le modifiche e le operazioni insiemistiche devono garantire la proprietà del Set, di conseguenza tutte le espressioni vengono valutate nella correttezza sintattica e nel rispetto dei vincoli contrattuali (utente-operatore) stipulati.

Creazione di un insieme

Un insieme può essere creato mediante:

- **Empty(t)** -> dove "t" corrisponde al tipo dell'insieme
:=> restituisce un insieme vuoto di tipo "t"
- **Singleton(exp)** -> dove "exp" corrisponde al primo valore da inserire nell'insieme
:=> restituisce un insieme contenente il tipo exp valutato
- **Of(collection)** -> dove "collection" sono tutti valori di tipo exp che si vuole inserire nell'insieme da creare
:=> restituisce un insieme, contenente tutti i valori exp valutati

Operazioni base

- **Push(Set, Value)** -> inserisce il valore all'interno del Set, mantenendo tutte le proprietà stabilite.
- **RemoveFrom(Set, Value)** -> rimuove il valore Value dall'insieme Set. Se il valore non è presente, restituisce Set.

Operazioni insiemistiche

- **Union(Set1, Set2)** -> esegue l'unione insiemistica dei 2 insiemi Set1 e Set2, mantenendo le proprietà stabilite
- **Intersection(Set1, Set2)** -> esegue l'intersezione insiemistica dei 2 insiemi Set1 e Set2, mantenendo le proprietà stabilite
- **Difference(Set1, Set2)** -> esegue la differenza tra Set1 e Set2, mantenendo le proprietà stabilite
- **IsSubset(Set1, Set2)** -> restituisce true se l'insieme Set1 è sottoinsieme di Set2, false altrimenti

Altre operazioni

- **IsEmpty(Set)** -> restituisce true se l'insieme Set è vuoto, false altrimenti
- **Contains(Set, Value)** -> restituisce true se l'insieme Set contiene il valore Value, false altrimenti
- **MaxOf(Set)** e **MinOf(Set)** -> restituiscono rispettivamente il massimo e il minimo dell'insieme Set

Operatori di natura "funzionale"

- **For_all(Pred, Set)**, **Exists(Pred, Set)**, **Filter(Pred, Set)** -> viene controllato se il predicato, passato come parametro, è effettivamente un predicato. Questo lo si può dire in quanto, per definizione, un predicato è una funzione che restituisce sempre un valore booleano. Nel caso sia tutto corretto, viene applicato il predicato a tutti i valori all'interno del Set.
- **Map(Fun, Set)** -> Viene applicata la funzione Fun a tutti i valori del Set. Questo metodo modifica l'insieme, ergo viene utilizzata la Push(Set, Value) per il corretto rispetto delle proprietà stabilite