

Sintassi Concreta

In questo documento sono elencati tutti i costrutti del linguaggio nella loro sintassi concreta.

Blocchi

Let var = exp **in** body

- Viene valutato il body nell'ambiente contenente il binding tra var e il valore ottenuto dalla valutazione di exp

LetRec fun = bodyFun **in** bodyLet

- Viene valutato il bodyLet nell'ambiente contenente il binding tra fun e la sua chiusura (codFun, puntatore all'ambiente statico).

Operazioni su Float/Interi

- Somma: $a1 + a2 = a3$
 - a1 deve essere un intero/float
 - a2 deve essere un intero/float
 - a3 è un intero solo se a1 e a2 sono interi. Altrimenti è un float.

Le stesse regole valgono per tutte le altre operazioni binarie.

- Differenza: $a1 - a2 = a3$.
- Divisione: $a1 / a2 = a3$.
- Moltiplicazione: $a1 * a2 = a3$.
- Modulo: $a1 \% a2 = a3$
- GreaterThan/LowerThan: $a1 >/< a2 = a3$
 - a3 è un booleano
- Floor/Ceil: **Floor/Ceil** $a1 = a2$
 - a1 deve essere un intero/float.
 - a2 è un float.

Operazioni su Booleani

- And: $a1 \text{ and } a2 = a3$.
- Or: $a1 \text{ or } a2 = a3$.
- Not: **not** $a1 = a3$.
- Implicazione: $a1 \rightarrow a2 = a3$.
 - a1 è un booleano.
 - a2 è un booleano.
 - a3 è il risultato dell'operazione (un booleano).

Operazioni su Stringhe/Booleani/Interi/Float/Array

- Concatenamento: $s1 \wedge s2 = s3$
 - s1 deve essere una stringa/intero/float/booleano/Array.
 - s2 deve essere una stringa/intero/float/booleano/Array.
 - s3 è una stringa ottenuta dal concatenamento di s1 e s2.

Operazioni su Array

Let a = Type [len]

- Crea un array di tipo Type di lunghezza len.

Let a2 = Type [len] **initialized to** val

- Crea un array di tipo Type di dimensione len in cui ogni elemento è inizializzato a val.

Let a3 = [val1; val2; ...; valN]

- Crea un array di tipo Type di dimensione N tale che a3 == [val1; val2; ...; valN] vale true.

Let a4 = **Set** array index1 index2 ... val

- Dato un array di tipo Type, restituisce un array a2 tale che a2 index1 index2 ... = val e per ogni i j ... ≠ index1 index2, a2 i j ... = array i j.

Let a5 = **Get** array index1 index2 ...

- Restituisce l'elemento index1 index2 ... dell'array.

Operazioni su Dizionario (Oltre a quelle elencate nel documento del progetto)

Let v1 = **Select** dictionary index

- Restituisce l'index-esimo elemento del dizionario.

Ler d2 = **Remove** dictionary index

- Restituisce un dizionario uguale a dictionary senza l'index-esimo elemento.

Operazioni su Dizionari/Array

- Lunghezza: **length** v = len
 - v è un dizionario/array
 - len è un intero che indica la lunghezza dell'array/dizionario.

Operazioni su Stringhe/Booleani/Interi/Float/Array/Dizionari

- Uguaglianza: (v1 = v2) <-> v3
 - v1 è una stringa, Booleano, Intero, Float, Array, Dizionario.
 - v2 è una stringa, Booleano, Intero, Float, Array, Dizionario, Tipo.
 - v3 = true se e solo se v1 = v2.

IfThenElse

- **If** espressione booleana b **then** body1 **else** body2
 - Valuta b, se è vera valuta body1 altrimenti valuta body2.

Switch

- **Switch** val1 val2 ... **match with**
 - | sv1 sv12 ... -> body1
 - | sv2 sv22 ... -> body2

- Se val1 = sv1 e val2 = sv12 e ... allora valuta body1. Altrimenti passa al secondo caso ecc.
- Se sv1 = **“Default”** allora viene valutato il body1. Lo stesso negli altri casi dello switch.

Exception

- Sollevare eccezione: **Exception**(Stringa di errore)

Gli esempi sono nella batteria di test.