

Funcons-beta: Multisets *

The P_LanCompS Project

Multisets.cbs | PLAIN | PRETTY

Multisets (bags)

```
[ Type multisets
  Funcon multiset
  Funcon multiset-elements
  Funcon multiset-occurrences
  Funcon multiset-insert
  Funcon multiset-delete
  Funcon is-submultiset ]
```

Meta-variables $GT <: \text{ground-values}$

Built-in Type $\text{multisets}(GT)$

$\text{multisets}(GT)$ is the type of possibly-empty finite multisets of elements of GT .

Built-in Funcon $\text{multiset}(_ : (GT)^*) : \Rightarrow \text{multisets}(GT)$

Note that $\text{multiset}(\dots)$ is not a constructor operation. The order of argument values is ignored, but duplicates are significant, e.g., $\text{multiset}(1, 2, 2)$ is equivalent to $\text{multiset}(2, 1, 2)$, but not to $\text{multiset}(1, 2)$ or $\text{multiset}(2, 1)$.

Built-in Funcon $\text{multiset-elements}(_ : \text{multisets}(GT)) : \Rightarrow (GT)^*$

For each multiset MS , the sequence of values V^* returned by $\text{multiset-elements}(MS)$ contains each element of MS the same number of times as MS does. The order of the values in V^* is unspecified, and may vary between multisets.

Assert $\text{multiset}(\text{multiset-elements}(S)) == S$

Built-in Funcon $\text{multiset-occurrences}(_ : GT, _ : \text{multisets}(GT)) : \Rightarrow \text{natural-numbers}$

$\text{multiset-occurrences}(GV, MS)$ returns the number of occurrences of GV in MS .

Built-in Funcon $\text{multiset-insert}(_ : GT, _ : \text{natural-numbers}, _ : \text{multisets}(GT)) : \Rightarrow \text{multisets}(GT)$

*Suggestions for improvement: plancomps@gmail.com.
Reports of issues: <https://github.com/plancomps/CBS-beta/issues>.

`multiset-insert`(GV, N, MS) returns the multiset that differs from MS by containing N more copies of GV .

Built-in Funcon `multiset-delete`($_ : multisets(GT), _ : GT, _ : natural-numbers$) : $\Rightarrow multisets(GT)$

`multiset-delete`(MS, GV, N) removes N copies of V from the multiset MS , or all copies of GV if there are fewer than N in MS .

Built-in Funcon `is-submultiset`($_ : multisets(GT), _ : multisets(GT)$) : $\Rightarrow booleans$

`is-submultiset`(MS_1, MS_2) tests whether every element of MS_1 has equal or fewer occurrences in MS_1 than in MS_2 .