

Unstable-Funcons-beta: Memos

The P_{Plan}CompS Project

Unstable-Funcons-beta/Computations/Normal/Memos/Memos.cbs*

Memos

```
[ Entity memo-map
  Funcon initialise-memos
  Funcon memo-value
  Funcon initialise-memo-value
  Funcon memo-value-recall ]
```

A memo is like a mutable variable, except that the memo is updated and accessed by a specified key, rather than by an allocated location. The collection of memos is represented by a mutable entity that maps keys to values.

```
Entity ⟨-, memo-map(- : maps(ground-values,
  values)))⟩ → ⟨-, memo-map(- : maps(ground-values,
  values)))⟩
```

```
Funcon initialise-memos(- : ⇒ values) : ⇒ values
Rule ⟨initialise-memos(X), memo-map(.)⟩ → ⟨X, memo-map(map( ))⟩
```

When key K is associated with value V , the funcon `memo-value(K , X)` simply gives V , without evaluating X . When K is not currently associated with any value, it associates K with the value computed by X .

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

Funcon $\text{memo-value}(K : \text{ground-values}, X : \Rightarrow \text{values}) : \Rightarrow \text{values}$
 $\rightsquigarrow \text{else}(\text{memo-value-recall}(K),$
 $\quad \text{give}(X,$
 $\quad \quad \text{sequential}(\text{else}(\text{initialise-memo-value}(K,$
 $\quad \quad \quad \text{given}),$
 $\quad \quad \quad \text{null-value}),$
 $\quad \quad \quad \text{memo-value-recall}(K))))$

The initialisation could fail due to memoisation of a (potentially different) value for K during the computation X . In that case, the value computed by X is simply discarded; a resource-safe funcon would take an extra argument to roll back the effects of X .

Funcon $\text{initialise-memo-value}(_ : \text{ground-values}, _ : \text{values}) : \Rightarrow \text{null-type}$

Rule
$$\frac{\text{map-unite}(M, \{K \mapsto V\}) \rightsquigarrow M'}{\langle \text{initialise-memo-value}(K : \text{ground-values}, V : \text{values}), \text{memo-map}(M) \rangle \longrightarrow \langle \text{null-value}, \text{memo-map}(M) \rangle}$$

Rule
$$\frac{\text{map-unite}(M, \{K \mapsto V\}) \rightsquigarrow ()}{\langle \text{initialise-memo-value}(K : \text{ground-values}, V : \text{values}), \text{memo-map}(M) \rangle \longrightarrow \langle \text{fail}, \text{memo-map}(M) \rangle}$$

Funcon $\text{memo-value-recall}(_ : \text{ground-values}) : \Rightarrow \text{values}$

Rule
$$\frac{\text{lookup}(M, K) \rightsquigarrow V}{\langle \text{memo-value-recall}(K : \text{ground-values}), \text{memo-map}(M) \rangle \longrightarrow \langle V, \text{memo-map}(M) \rangle}$$

Rule
$$\frac{\text{lookup}(M, K) \rightsquigarrow ()}{\langle \text{memo-value-recall}(K : \text{ground-values}), \text{memo-map}(M) \rangle \longrightarrow \langle \text{fail}, \text{memo-map}(M) \rangle}$$