Unstable-Languages-beta: LD-Funcons-Index *

The PLanCompS Project

`LD-Funcons-Index.cbs` | PLAIN | PRETTY

OUTLINE

Computations

Values

Computations

Normal computation

Flowing

[*Funcon*   left-to-right
     Alias   l-to-r
  Funcon   sequential
     Alias   seq
  Funcon   effect
  Funcon   if-true-else
     Alias   if-else
  Funcon   while-true
     Alias   while]

Giving

[*Funcon*   give
  Funcon   given]

Binding

[*Funcon*   initialise-binding
  Funcon   bind-value
     Alias   bind
  Funcon   bound-value
     Alias   bound
  Funcon   scope]

Generating

[*Type*   atoms
  Funcon   fresh-atom]

Storing

[*Funcon*   initialise-storing
  Datatype   variables
     Alias   vars
  Funcon   allocate-initialised-variable
     Alias   alloc-init
  Funcon   assign
  Funcon   assigned]

Abnormal computation

Terminating abruptly

[*Funcon*   handle-abrupt]

Failing

[*Funcon*   finalise-failing
  Funcon   fail
  Funcon   checked]

Values

Value Types

[*Type*   values
   Alias   vals
   Funcon   is-value
   Alias   is-val
   Type   ground-values
   Alias   ground-vals]

Primitive values

Booleans

[*Datatype*   booleans
   Alias   bools
   Funcon   true
   Funcon   false]

Integers

[*Type*   integers
   Alias   ints
   Type   natural-numbers
   Alias   nats
   Funcon   integer-add
   Alias   int-add
   Funcon   integer-multiply
   Alias   int-mul
   Funcon   integer-divide
   Alias   int-div
   Funcon   integer-is-less-or-equal
   Alias   is-less-or-equal
   Funcon   decimal-natural
   Alias   decimal]

The null value

[*Datatype*   null-type
   Funcon   null-value
   Alias   null]

Composite values

Sequences of values

[*Funcon*   length
   Funcon   index]

Lists

[*Datatype*   lists]

Sets

[*Type*   sets
Funcon   is-in-set
Funcon   is-subset
Funcon   set-unite
Funcon   set-intersect
Funcon   set-difference
Funcon   some-element]

Maps

[*Type*   maps
Funcon   map
Funcon   map-lookup
Alias   lookup
Funcon   map-domain
Alias   dom
Funcon   map-override
Funcon   map-unite
Funcon   map-delete]

Abstraction values

Generic abstractions

[*Type*   abstractions
Funcon   abstraction
Funcon   closure]

Thunks

[*Datatype*   thunks
Funcon   thunk]

Functions

[*Datatype*   functions
Funcon   function
Funcon   apply
Funcon   supply]