Funcons-beta: Funcons-Index [*]

The PLanCompS Project

`Funcons-Index.cbs` | PLAIN | PRETTY

OUTLINE

Computations

- Types of computation
- Normal computation
    - Flowing
    - Giving
    - Binding
    - Generating
    - Storing
    - Linking
    - Interacting
        - Input
        - Output
- Abnormal computation
    - Terminating abruptly
    - Failing
    - Throwing
    - Returning
    - Breaking
    - Continuing
    - Controlling

Values

- Value Types
- Primitive values
    - Booleans
    - Integers
    - Floats
    - Characters
    - The null value
- Composite values
    - Sequences of values
    - Datatypes
    - Tuples
    - Lists
    - Strings
    - Vectors
    - Bits and bit vectors
    - Sets
    - Maps
    - Multisets (bags)
    - Trees
    - Graphs
    - References and pointers
    - Records

[*] Suggestions for improvement: `plancomps@gmail.com`.
Reports of issues: `https://github.com/plancomps/CBS-beta/issues`.

  - Variants
  - Classes
  - Objects
- Abstraction values
  - Generic abstractions
  - Thunks
  - Functions
  - Patterns

Computations

Types of computation

```
[ Funcon   computation-types ]
```

Normal computation

Flowing

```
[ Funcon    left-to-right
    Alias   l-to-r
   Funcon   right-to-left
    Alias   r-to-l
   Funcon   sequential
    Alias   seq
   Funcon   effect
   Funcon   choice
   Funcon   if-true-else
    Alias   if-else
   Funcon   while-true
    Alias   while
   Funcon   do-while-true
    Alias   do-while
   Funcon   interleave
 Datatype   yielding
   Funcon   signal
   Funcon   yielded
   Funcon   yield
   Funcon   yield-on-value
   Funcon   yield-on-abrupt
   Funcon   atomic ]
```

Giving

```
[ Entity   given-value
  Funcon   initialise-giving
  Funcon   give
  Funcon   given
  Funcon   no-given
  Funcon   left-to-right-map
  Funcon   interleave-map
  Funcon   left-to-right-repeat
  Funcon   interleave-repeat
  Funcon   left-to-right-filter
  Funcon   interleave-filter
  Funcon   fold-left
  Funcon   fold-right ]
```

Binding

```
    [ Type   environments
      Alias  envs
   Datatype  identifiers
      Alias  ids
     Funcon  identifier-tagged
      Alias  id-tagged
     Funcon  fresh-identifier
     Entity  environment
      Alias  env
     Funcon  initialise-binding
     Funcon  bind-value
      Alias  bind
     Funcon  unbind
     Funcon  bound-directly
     Funcon  bound-value
      Alias  bound
     Funcon  closed
     Funcon  scope
     Funcon  accumulate
     Funcon  collateral
     Funcon  bind-recursively
     Funcon  recursive ]
```

Generating

```
  [ Type   atoms
  Entity   used-atom-set
  Funcon   initialise-generating
  Funcon   fresh-atom
  Funcon   use-atom-not-in ]
```

Storing

[*Datatype*	locations
Alias	locs
Type	stores
Entity	store
Funcon	initialise-storing
Funcon	store-clear
Datatype	variables
Alias	vars
Funcon	variable
Alias	var
Funcon	allocate-variable
Alias	alloc
Funcon	recycle-variables
Alias	recycle
Funcon	initialise-variable
Alias	init
Funcon	allocate-initialised-variable
Alias	alloc-init
Funcon	assign
Funcon	assigned
Funcon	current-value
Funcon	un-assign
Funcon	structural-assign
Funcon	structural-assigned]

Linking

[*Datatype*	links
Funcon	initialise-linking
Funcon	link
Funcon	fresh-link
Funcon	fresh-initialised-link
Alias	fresh-init-link
Funcon	set-link
Funcon	follow-if-link]

Interacting

Input

[*Entity*	standard-in
Funcon	read]

Output

[*Entity*	standard-out
Funcon	print]

Abnormal computation

Terminating abruptly

```
[ Funcon  stuck
   Entity  abrupted
  Funcon  finalise-abrupting
  Funcon  abrupt
  Funcon  handle-abrupt
  Funcon  finally ]
```

Failing

```
[ Datatype  failing
    Funcon  failed
    Funcon  finalise-failing
    Funcon  fail
    Funcon  else
    Funcon  else-choice
    Funcon  checked
    Funcon  check-true ]
```

Throwing

```
[ Datatype  throwing
    Funcon  thrown
    Funcon  finalise-throwing
    Funcon  throw
    Funcon  handle-thrown
    Funcon  handle-recursively
    Funcon  catch-else-throw ]
```

Returning

```
[ Datatype  returning
    Funcon  returned
    Funcon  finalise-returning
    Funcon  return
    Funcon  handle-return ]
```

Breaking

```
[ Datatype  breaking
    Funcon  broken
    Funcon  finalise-breaking
    Funcon  break
    Funcon  handle-break ]
```

Continuing

```
[ Datatype  continuing
    Funcon  continued
    Funcon  finalise-continuing
    Funcon  continue
    Funcon  handle-continue ]
```

Controlling

```
[ Datatype  continuations
    Funcon  continuation
    Entity  plug-signal
    Funcon  hole
    Funcon  resume-continuation
    Entity  control-signal
    Funcon  control
    Funcon  delimit-current-continuation
     Alias  delimit-cc ]
```

Values

Value Types

```
[ Type  values
  Alias  vals
   Type  value-types
  Alias  types
   Type  empty-type
 Funcon  is-in-type
  Alias  is
 Funcon  is-value
  Alias  is-val
 Funcon  when-true
  Alias  when
   Type  cast-to-type
  Alias  cast
   Type  ground-values
  Alias  ground-vals
 Funcon  is-equal
  Alias  is-eq ]
```

Primitive values

Booleans

```
[ Datatype   booleans
     Alias   bools
    Funcon   true
    Funcon   false
    Funcon   not
    Funcon   implies
    Funcon   and
    Funcon   or
    Funcon   exclusive-or
     Alias   xor ]
```

Integers

[*Type*	integers
Alias	ints
Type	integers-from
Alias	from
Type	integers-up-to
Alias	up-to
Type	bounded-integers
Alias	bounded-ints
Type	positive-integers
Alias	pos-ints
Type	negative-integers
Alias	neg-ints
Type	natural-numbers
Alias	nats
Funcon	natural-successor
Alias	nat-succ
Funcon	natural-predecessor
Alias	nat-pred
Funcon	integer-add
Alias	int-add
Funcon	integer-subtract
Alias	int-sub
Funcon	integer-multiply
Alias	int-mul
Funcon	integer-divide
Alias	int-div
Funcon	integer-modulo
Alias	int-mod
Funcon	integer-power
Alias	int-pow
Funcon	integer-absolute-value
Alias	int-abs
Funcon	integer-negate
Alias	int-neg
Funcon	integer-is-less
Alias	is-less
Funcon	integer-is-less-or-equal
Alias	is-less-or-equal
Funcon	integer-is-greater
Alias	is-greater
Funcon	integer-is-greater-or-equal
Alias	is-greater-or-equal
Funcon	binary-natural
Alias	binary
Funcon	octal-natural
Alias	octal
Funcon	decimal-natural
Alias	decimal
Funcon	hexadecimal-natural
Alias	hexadecimal
Funcon	integer-sequence]

Floats

[Datatype	float-formats
Funcon	binary32
Funcon	binary64
Funcon	binary128
Funcon	decimal64
Funcon	decimal128
Type	floats
Funcon	float
Funcon	quiet-not-a-number
Alias	qNaN
Funcon	signaling-not-a-number
Alias	sNaN
Funcon	positive-infinity
Alias	pos-inf
Funcon	negative-infinity
Alias	neg-inf
Funcon	float-convert
Funcon	float-equal
Funcon	float-is-less
Funcon	float-is-less-or-equal
Funcon	float-is-greater
Funcon	float-is-greater-or-equal
Funcon	float-negate
Funcon	float-absolute-value
Funcon	float-add
Funcon	float-subtract
Funcon	float-multiply
Funcon	float-multiply-add
Funcon	float-divide
Funcon	float-remainder
Funcon	float-sqrt
Funcon	float-integer-power
Funcon	float-float-power
Funcon	float-round-ties-to-even
Funcon	float-round-ties-to-infinity
Funcon	float-floor
Funcon	float-ceiling
Funcon	float-truncate
Funcon	float-pi
Funcon	float-e
Funcon	float-log
Funcon	float-log10
Funcon	float-exp
Funcon	float-sin
Funcon	float-cos
Funcon	float-tan
Funcon	float-asin
Funcon	float-acos
Funcon	float-atan
Funcon	float-sinh
Funcon	float-cosh

Characters

[*Type*	characters
Alias	chars
Datatype	unicode-characters
Alias	unicode-chars
Type	unicode-points
Funcon	unicode-character
Alias	unicode-char
Funcon	unicode-point
Alias	unicode
Type	basic-multilingual-plane-characters
Alias	bmp-chars
Type	basic-multilingual-plane-points
Type	iso-latin-1-characters
Alias	latin-1-chars
Type	iso-latin-1-points
Type	ascii-characters
Alias	ascii-chars
Type	ascii-points
Funcon	ascii-character
Alias	ascii-char
Funcon	utf-8
Funcon	utf-16
Funcon	utf-32
Funcon	backspace
Funcon	horizontal-tab
Funcon	line-feed
Funcon	form-feed
Funcon	carriage-return
Funcon	double-quote
Funcon	single-quote
Funcon	backslash]

The null value

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

Composite values

Sequences of values

```
[ Funcon   length
  Funcon   index
  Funcon   is-in
  Funcon   first
  Funcon   second
  Funcon   third
  Funcon   first-n
  Funcon   drop-first-n
  Funcon   reverse
  Funcon   n-of
  Funcon   intersperse ]
```

Datatypes

```
[ Funcon   datatype-value
  Funcon   datatype-value-id
  Funcon   datatype-value-elements ]
```

Tuples

```
[ Datatype   tuples
    Funcon   tuple-elements
    Funcon   tuple-zip ]
```

Lists

```
[ Datatype   lists
    Funcon   list
    Funcon   list-elements
    Funcon   list-nil
     Alias   nil
    Funcon   list-cons
     Alias   cons
    Funcon   list-head
     Alias   head
    Funcon   list-tail
     Alias   tail
    Funcon   list-length
    Funcon   list-append ]
```

Strings

```
 [ Type   strings
Funcon   string
Funcon   string-append
Funcon   to-string ]
```

Vectors

```
[ Datatype   vectors
    Funcon   vector
    Funcon   vector-elements ]
```

Bits and bit vectors

```
    [ Type   bits
  Datatype   bit-vectors
    Funcon   bit-vector
      Type   bytes
     Alias   octets
    Funcon   bit-vector-not
    Funcon   bit-vector-and
    Funcon   bit-vector-or
    Funcon   bit-vector-xor
    Funcon   bit-vector-shift-left
    Funcon   bit-vector-logical-shift-right
    Funcon   bit-vector-arithmetic-shift-right
    Funcon   integer-to-bit-vector
    Funcon   bit-vector-to-integer
    Funcon   bit-vector-to-natural
    Funcon   unsigned-bit-vector-maximum
    Funcon   signed-bit-vector-maximum
    Funcon   signed-bit-vector-minimum
    Funcon   is-in-signed-bit-vector
    Funcon   is-in-unsigned-bit-vector ]
```

Sets

```
 [ Type   sets
Funcon   set
Funcon   set-elements
Funcon   is-in-set
Funcon   is-subset
Funcon   set-insert
Funcon   set-unite
Funcon   set-intersect
Funcon   set-difference
Funcon   set-size
Funcon   some-element
Funcon   element-not-in ]
```

Maps

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

Multisets (bags)

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

Trees

[*Datatype*	trees
Funcon	tree
Funcon	tree-root-value
Funcon	tree-branch-sequence
Funcon	single-branching-sequence
Funcon	forest-root-value-sequence
Funcon	forest-branch-sequence
Funcon	forest-value-sequence]

Graphs

[*Type*	directed-graphs
Funcon	is-cyclic
Funcon	topological-sort]

References and pointers

[*Datatype*	references
Funcon	reference
Type	pointers
Funcon	dereference]

Records

[*Datatype*	records
Funcon	record
Funcon	record-map
Funcon	record-select]

Variants

[Datatype	variants
Funcon	variant
Funcon	variant-id
Funcon	variant-value]

Classes

[Datatype	classes
Funcon	class
Funcon	class-instantiator
Funcon	class-feature-map
Funcon	class-superclass-name-sequence
Funcon	class-name-tree
Funcon	is-subclass-name
Funcon	class-name-single-inheritance-feature-map]

Objects

[Datatype	objects
Funcon	object
Funcon	object-identity
Funcon	object-class-name
Funcon	object-feature-map
Funcon	object-subobject-sequence
Funcon	object-tree
Funcon	object-single-inheritance-feature-map]

Abstraction values

Generic abstractions

[Type	abstractions
Funcon	abstraction
Funcon	closure
Funcon	enact]

Thunks

[Datatype	thunks
Funcon	thunk
Funcon	force]

Functions

[Datatype	functions
Funcon	function
Funcon	apply
Funcon	supply
Funcon	compose
Funcon	uncurry
Funcon	curry
Funcon	partial-apply]

Patterns

```
[ Datatype  patterns
   Funcon   pattern
   Funcon   pattern-any
   Funcon   pattern-bind
   Funcon   pattern-type
   Funcon   pattern-else
   Funcon   pattern-unite
   Funcon   match
   Funcon   match-loosely
   Funcon   case-match
   Funcon   case-match-loosely
   Funcon   case-variant-value ]
```