Funcons-beta: Sequences [*]

The PLanCompS Project

Sequences.cbs | PLAIN | PRETTY

OUTLINE

Sequences of values
  Sequence indexing
  Homogeneous sequences

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]

Sequences of two or more values are not themselves values, nor is the empty sequence a value. However, sequences can be provided to funcons as arguments, and returned as results. Many operations on composite values can be expressed by extracting their components as sequences, operating on the sequences, then forming the required composite values from the resulting sequences.

A sequence with elements X_1, . . . , X_n is written $X_1, \cdots, X_n$. A sequence with a single element X is identified with (and written) X. An empty sequence is indicated by the absence of a term. Any sequence X^* can be enclosed in parentheses (X^*), e.g.: ( ), (1), (1, 2, 3). Superfluous commas are ignored.

The elements of a type sequence $T_1, \cdots, T_n$ are the value sequences $V_1, \cdots, V_n$ where $V_1 : T_1$, . . . , $V_n : T_n$. The only element of the empty type sequence ( ) is the empty value sequence ( ).

$(T)^N$ is equivalent to $T, \cdots, T$ with N occurrences of T. $(T)^*$ is equivalent to the union of all $(T)^N$ for N>=0, $(T)^+$ is equivalent to the union of all $(T)^N$ for N>=1, and $(T)^?$ is equivalent to $T \mid ( )$. The parentheses around T above can be omitted when they are not needed for disambiguation.

(Non-trivial) sequence types are not values, so not included in types.

Meta-variables T, T' <: values

Funcon length(_ : values*) : ⇒ natural-numbers

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

length(V^*) gives the number of elements in V^*.

Rule length( ) $\leadsto$ 0
Rule length(V : values, V^* : values*) $\leadsto$ natural-successor(length(V^*))

Funcon is-in(_ : values, _ : values*) : $\Rightarrow$ booleans
Rule is-in(V : values, V' : values, V^* : values*) $\leadsto$ or(is-equal(V, V'), is-in(V, V^*))
Rule is-in(V : values, ( )) $\leadsto$ false

Sequence indexing

Funcon index(_ : natural-numbers, _ : values*) : $\Rightarrow$ values$^?$

index(N, V^*) gives the Nth element of V^*, if it exists, otherwise ( ).

Rule index(1, V : values, V^* : values*) $\leadsto$ V

Rule
$$\frac{\text{natural-predecessor}(N) \leadsto N'}{\text{index}(N : \text{positive-integers}, \_ : \text{values}, V^* : \text{values}^*) \leadsto \text{index}(N', V^*)}$$

Rule index(0, V^* : values*) $\leadsto$ ( )
Rule index(_ : positive-integers, ( )) $\leadsto$ ( )

Total indexing funcons:

Funcon first(_ : T, _ : values*) : $\Rightarrow$ T
Rule first(V : T, V^* : values*) $\leadsto$ V

Funcon second(_ : values, _ : T, _ : values*) : $\Rightarrow$ T
Rule second(_ : values, V : T, V^* : values*) $\leadsto$ V

Funcon third(_ : values, _ : values, _ : T, _ : values*) : $\Rightarrow$ T
Rule third(_ : values, _ : values, V : T, V^* : values*) $\leadsto$ V

Homogeneous sequences

Funcon first-n(_ : natural-numbers, _ : $(T)^*$) : $\Rightarrow$$(T)^*$
Rule first-n(0, V^* : $(T)^*$) $\leadsto$ ( )

Rule
$$\frac{\text{natural-predecessor}(N) \leadsto N'}{\text{first-n}(N : \text{positive-integers}, V : T, V^* : (T)^*) \leadsto (V, \text{first-n}(N', V^*))}$$

Rule first-n(N : positive-integers, ( )) $\leadsto$ ( )

Funcon drop-first-n(_ : natural-numbers, _ : $(T)^*$) : $\Rightarrow$$(T)^*$
Rule drop-first-n(0, V^* : $(T)^*$) $\leadsto$ V^*

Rule
$$\frac{\text{natural-predecessor}(N) \leadsto N'}{\text{drop-first-n}(N : \text{positive-integers}, \_ : T, V^* : (T)^*) \leadsto \text{drop-first-n}(N', V^*)}$$

Rule drop-first-n(N : positive-integers, ( )) $\leadsto$ ( )

Funcon reverse(_ : $(T)^*$) : $\Rightarrow$$(T)^*$
Rule reverse( ) $\leadsto$ ( )
Rule reverse(V : T, V^* : $(T)^*$) $\leadsto$ (reverse(V^*), V)

Funcon $\text{n-of}(N : \text{natural-numbers}, V : T) : \Rightarrow (T)^*$

Rule $\text{n-of}(0, \_ : T) \rightsquigarrow (\,)$

Rule $$\frac{\text{natural-predecessor}(N) \rightsquigarrow N'}{\text{n-of}(N : \text{positive-integers}, V : T) \rightsquigarrow (V, \text{n-of}(N', V))}$$

Funcon $\text{intersperse}(\_ : T', \_ : (T)^*) : \Rightarrow (T, (T', T)^*)^?$

Rule $\text{intersperse}(\_ : T', (\,)) \rightsquigarrow (\,)$

Rule $\text{intersperse}(\_ : T', V) \rightsquigarrow V$

Rule $\text{intersperse}(V' : T', V_1 : T, V_2 : T, V^* : (T)^*) \rightsquigarrow (V_1, V', \text{intersperse}(V', V_2, V^*))$