Funcons-beta: Lists *

The PLanCompS Project

`Lists.cbs` | PLAIN | PRETTY

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]

Meta-variables $T <:$ values

Datatype lists(T) ::= list($\_ : (T)^*$)

lists(T) is the type of possibly-empty finite lists $[V_1, \cdots, V_n]$ where $V_1 : T, \ldots, V_n : T$.
N.B. $[T]$ is always a single list value, and *not* interpreted as the type lists(T).
The notation $[V_1, \cdots, V_n]$ for list($V_1, \cdots, V_n$) is built-in.

Assert $[V^* : \text{values}^*] == \text{list}(V^*)$

Funcon list-elements($\_ :$ lists(T)) : $\Rightarrow (T)^*$
Rule list-elements(list($V^* :$ values*)) $\rightsquigarrow V^*$

Funcon list-nil : $\Rightarrow$ lists($\_$)
$\rightsquigarrow$ [ ]
Alias nil = list-nil

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

Funcon list-cons(_ : T, _ : lists(T)) : ⇒ lists(T)

Alias cons = list-cons

Rule list-cons(V : values, [V^* : values*]) ⇝ [V, V^*]

Funcon list-head(_ : lists(T)) : ⇒(T)?

Alias head = list-head

Rule list-head [V : values, _* : values*] ⇝ V

Rule list-head [ ] ⇝ ( )

Funcon list-tail(_ : lists(T)) : ⇒(lists(T))?

Alias tail = list-tail

Rule list-tail [_ : values, V^* : values*] ⇝ [V^*]

Rule list-tail [ ] ⇝ ( )

Funcon list-length(_ : lists(T)) : ⇒ natural-numbers

Rule list-length [V^* : values*] ⇝ length(V^*)

Funcon list-append(_ : (lists(T))*) : ⇒ lists(T)

Rule list-append([${V_1}^*$: values*], [${V_2}^*$: values*]) ⇝ [${V_1}^*$, ${V_2}^*$]

Rule list-append(L_1 : lists(_), L_2 : lists(_), L_3 : lists(_), L^* : (lists(_))*) ⇝
list-append(L_1, list-append(L_2, L_3, L^*))

Rule list-append( ) ⇝ [ ]

Rule list-append(L : lists(_)) ⇝ L

Datatypes of infinite and possibly-infinite lists can be specified as algebraic datatypes using abstractions.