Funcons-beta: Trees

The PLanCompS Project

`Funcons-beta/Values/Composite/Trees/Trees.cbs`*

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]

Meta-variables T <: values

Datatype trees(T) ::= tree($\_ : T, \_ : (\text{trees}(T))^*$)

trees(T) consists of finitely-branching trees with elements of type T. When $V : T$, tree(V) is a leaf, and tree($V, B_1, \cdots, B_n$) is a tree with branches B_1, ..., B_n.

Funcon tree-root-value($\_ : \text{trees}(T)$) : $\Rightarrow (T)$?
  Rule tree-root-value tree($V : T, \_^* : (\text{trees}(T))^*$) $\rightsquigarrow V$

Funcon tree-branch-sequence($\_ : \text{trees}(T)$) : $\Rightarrow (\text{trees}(T))^*$
  Rule tree-branch-sequence tree($\_ : T, B^* : (\text{trees}(T))^*$) $\rightsquigarrow B^*$

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

Funcon single-branching-sequence(_ : trees(T)) : ⇒ T^+

single-branching-sequence B extracts the values in B starting from the root, provided that B is at most single-branching; otherwise it fails.

Rule single-branching-sequence tree(V : T) ⇝ V
Rule single-branching-sequence tree(V : T, B : trees(T)) ⇝ left-to-right(V, single-branching-sequence B)
Rule single-branching-sequence tree(_ : T, _ : trees(T), _$^+$: (trees(T))$^+$) ⇝ fail

A sequence of trees corresponds to a forest, and the selector funcons on trees B extend to forests B^*:

Funcon forest-root-value-sequence(_ : (trees(T))*) : ⇒ T^*
Rule forest-root-value-sequence(B : trees(T), B^* : (trees(T))*) ⇝ (tree-root-value B, forest-root-value-se
Rule forest-root-value-sequence( ) ⇝ ( )

Funcon forest-branch-sequence(_ : (trees(T))*) : ⇒ T^*
Rule forest-branch-sequence(B : trees(T), B^* : (trees(T))*) ⇝ (tree-branch-sequence B, forest-branch-seq
Rule forest-branch-sequence( ) ⇝ ( )

Funcon forest-value-sequence(_ : (trees(T))*) : ⇒ T^*

forest-value-sequence B^* provides the values from a left-to-right pre-order depth-first traversal.

Rule forest-value-sequence(tree(V : T, $B_1{}^*$: (trees(T))*), $B_2{}^*$: (trees(T))*) ⇝ (V, forest-value-sequence B_1
Rule forest-value-sequence( ) ⇝ ( )

Other linearizations of trees can be added: breadth-first, right-to-left, C3, etc.