

This module specifies the iterator structure used in the `go-immutable-radix` project (<https://github.com/hashicorp/go-immutable-radix/>).

The iterator is meant to seek to some point in a radix tree and read all the subsequent values until it is over. Many algorithms make use of this “ordered read” property. This module specifies only the “read next” algorithm used by `go-immutable-radix`. From here, modules such as *RadixSeekLowerBound*, *RadixSeekPrefix*, etc. refine this module further to verify their own algorithms.

MODULE *RadixIterator*

LOCAL INSTANCE *RadixTrees*
 LOCAL INSTANCE *Sequences*
 LOCAL INSTANCE *FiniteSets*
 LOCAL INSTANCE *Integers*
 LOCAL INSTANCE *TLC*

CmpOp is the comparison operator for ordered iteration. This should be TRUE if the first value is less than the second value.

CONSTANT *CmpOp*(-, -)

TRUE iff the sequence *s* contains no duplicates. Copied from *CommunityModules*.

LOCAL *isInjective*(*s*) $\triangleq \forall i, j \in \text{DOMAIN } s : (s[i] = s[j]) \Rightarrow (i = j)$

Converts a set to a sequence that contains all the elements of *S* exactly once. Copied from *CommunityModules*.

LOCAL *setToSeq*(*S*) $\triangleq \text{CHOOSE } f \in [1 \dots \text{Cardinality}(S) \rightarrow S] : \text{isInjective}(f)$

Copied from *CommunityModules*.

LOCAL *mapThenFoldSet*(*op*(-, -), *base*, *f*(-), *choose*(-), *S*) \triangleq
 LET *iter*[*s* ∈ SUBSET *S*] \triangleq
 IF *s* = {} THEN *base*
 ELSE LET *x* \triangleq *choose*(*s*)
 IN *op*(*f*(*x*), *iter*[*s* \ {*x*}])
 IN *iter*[*S*]

foldLeft folds *op* on all elements of *seq* from left to right, starting with the first element and base. Copied from *CommunityModules*.

LOCAL *foldLeft*(*op*(-, -), *base*, *seq*) \triangleq
mapThenFoldSet(LAMBDA *x*, *y* : *op*(*y*, *x*), *base*,
 LAMBDA *i* : *seq*[*i*],
 LAMBDA *s* : CHOOSE *i* ∈ *s* : $\forall j \in s : i \geq j$,
 DOMAIN *seq*)

Internal logic for *Iterate*.

RECURSIVE *iterate*(-, -)
iterate(*T*, *prefix*) \triangleq

```

LET
  current  $\triangleq$  IF Len(T.Value) > 0 THEN  $\langle T.Value \rangle$  ELSE  $\langle \rangle$ 
  current value of node (if exists)

  orderedEdges  $\triangleq$  SortSeq(setToSeq(DOMAIN T.Edges), CmpOp)
  ordering that we'll visit edges

  children  $\triangleq$  [i ∈ 1 .. Len(orderedEdges) ↦
    iterate(T.Edges[orderedEdges[i]], prefix ∘ T.Prefix)]
  children values, this is a tuple of tuples

  flatChildren  $\triangleq$  foldLeft(LAMBDA x, y : x ∘ y,  $\langle \rangle$ , children)
  children as a single tuple of values

IN  current ∘ flatChildren

Iterate implements the core iteration algorithm. Given a sequence of nodes
this will return a sequence (not a set, since this is ordered) of keys that
are visited in the tree.
Iterate(Stack)  $\triangleq$ 
  foldLeft(LAMBDA x, y : x ∘ y,
     $\langle \rangle$ ,
    [i ∈ 1 .. Len(Stack) ↦ iterate(Stack[Len(Stack) - i + 1],  $\langle \rangle$ )]

```

```

\ * Modification History
\ * Last modified Fri Jul 02 08:12:17 PDT 2021 by mitchellh
\ * Created Tue Jun 29 19:49:11 PDT 2021 by mitchellh

```