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. It supports seeking by prefix and by lower bound.

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 $setToSeq(S) \triangleq \text{CHOOSE } f \in [1 \,..\, Cardinality(S) \to S] : isInjective(f)$

Copied from *CommunityModules*.
LOCAL $mapThenFoldSet(op(\_, \_),\ base,\ f(\_),\ choose(\_),\ S) \triangleq$
  LET $iter[s \in \text{SUBSET } S] \triangleq$
        IF $s = \{\}$ THEN $base$
        ELSE LET $x \triangleq choose(s)$
             IN $op(f(x),\ iter[s \setminus \{x\}])$
  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 $foldLeft(op(\_, \_),\ base,\ seq) \triangleq$
  $mapThenFoldSet(\text{LAMBDA } x,\ y : op(y,\ x),\ base,$
                  $\text{LAMBDA } i : seq[i],$
                  $\text{LAMBDA } s : \text{CHOOSE } i \in s : \forall\, j \in s : i \geq j,$
                  $\text{DOMAIN } seq)$

Internal logic for *Iterate*.
RECURSIVE $iterate(\_, \_)$
$iterate(T,\ prefix) \triangleq$
  LET
    $current \triangleq \text{IF } T.Value \text{ THEN } \langle prefix \circ T.Prefix \rangle \text{ ELSE } \langle\rangle$

current value of node (if exists)

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

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

$flatChildren \triangleq foldLeft(\text{LAMBDA } x, y : x \circ y, \langle\rangle, children)$
children as a single tuple of values
IN $current \circ flatChildren$

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.
$Iterate(Stack) \triangleq iterate(Stack[1], \langle\rangle)$ *TODO* doesn't do the whole stack!

---

* Modification History
* Last modified *Wed Jun* 30 10:00:54 *PDT* 2021 by *mitchellh*
* Created *Tue Jun* 29 19:49:11 *PDT* 2021 by *mitchellh*