MODULE *RadixIteratorValidation*

EXTENDS *FiniteSets*, *Integers*, *RadixTrees*, *Sequences*, *TLC*

Set of characters to use for the alphabet of generated strings.

CONSTANT *Alphabet*

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

CONSTANT $CmpOp(\_, \_)$

Length of input strings generated

CONSTANT *MinLength*, *MaxLength*

ASSUME

$\quad \land \{MinLength, MaxLength\} \subseteq Nat$

$\quad \land MinLength \leq MaxLength$

Number of unique elements to construct the radix tree with. This is a set of numbers so you can test with inputs of multiple sizes.

CONSTANT *ElementCounts*

ASSUME $ElementCounts \subseteq Nat$

Inputs is the set of input strings valid for the tree.

$Inputs \triangleq \text{UNION } \{[1 \,..\, n \rightarrow Alphabet] : n \in MinLength \,..\, MaxLength\}$

InputSets is the full set of possible inputs we can send to the radix tree.

$InputSets \triangleq \{T \in \text{SUBSET } Inputs : Cardinality(T) \in ElementCounts\}$

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

INSTANCE *RadixIterator*

The iteration of a tree should be just its sorted inputs.

$IterateIsSortedInput \triangleq$

$\quad \forall\, input \in InputSets :$

$\qquad \text{LET}$

$\qquad\quad actual \triangleq Iterate(\langle RadixTree(input)\rangle)$

CmpOp operates on individual elements so we have to write a LAMBDA here that performs per-element. We expect *CmpOp* to be a LESS THAN operation. The logic below does not work for GREATER THAN operations (∀ would have

```
      to be ∃).
      expected ≜ SortSeq(setToSeq(input),
        LAMBDA x, y :
            ∨ Len(x) < Len(y)
            ∨ ∧ Len(x) = Len(y)
              ∧ ∀ i ∈ DOMAIN x : CmpOp(x[i], y[i]))
    IN
      IF actual ≠ expected
       THEN Print(⟨"actual: ", actual, "expected: ", expected, "input: ", input⟩, FALSE)
       ELSE  TRUE

  The expression that should be checked for validity in the model.
  Valid ≜ IterateIsSortedInput
```

* Modification History
* Last modified *Thu Jul* 01 10:40:26 *PDT* 2021 by *mitchellh*
* Created *Thu Jul* 01 09:57:41 *PDT* 2021 by *mitchellh*