MODULE *BinarySearch*

This module defines a binary search algorithm for finding an item in a sorted sequence, and contains a *TLAPS*-checked proof of its safety property. We assume a sorted sequence *seq* with elements in some set Values of integers and a number *val* in *Values*, it sets the value *result* to either a number *i* with $seq[i] = val$, or to 0 if there is no such *i*.

It is surprisingly difficult to get such a binary search algorithm correct without making errors that have to be caught by debugging. I suggest trying to write a correct *PlusCal* binary search algorithm yourself before looking at this one.

This algorithm is one of the examples in Section 7.3 of "Proving Safety Properties", which is at

http://*lamport.azurewebsites.net*/tla/proving-*safety.pdf*

EXTENDS *Integers*, *Sequences*, *TLAPS*

CONSTANT *Values*

ASSUME $ValAssump \triangleq Values \subseteq Int$

$$\begin{array}{l} SortedSeqs \triangleq \{ss \in Seq(Values) : \\ \qquad \forall i, j \in 1 \,..\, Len(ss) : (i < j) \Rightarrow (ss[i] \leq ss[j])\} \end{array}$$

```
**************************************************************************
--fair algorithm BinarySearch{
   variables $seq \in SortedSeqs$, $val \in Values$,
             $low = 1$, $high = Len(seq)$, $result = 0$ ;
  { a: while ( $low \leq high \wedge result = 0$ ) {
         with ( $mid = (low + high) \div 2$, $mval = seq[mid]$ ) {
           if ( $mval = val$ ) { $result := mid$ }
           else if ( $val < mval$ ) { $high := mid - 1$ }
           else { $low := mid + 1$ }                    } } } }
**************************************************************************
```

BEGIN TRANSLATION

VARIABLES *seq*, *val*, *low*, *high*, *result*, *pc*

$$vars \triangleq \langle seq, val, low, high, result, pc \rangle$$

$Init \triangleq$ Global variables

$$\begin{array}{l} \wedge\ seq \in SortedSeqs \\ \wedge\ val \in Values \\ \wedge\ low = 1 \\ \wedge\ high = Len(seq) \\ \wedge\ result = 0 \\ \wedge\ pc = \text{"a"} \end{array}$$

$$\begin{array}{l} a \triangleq\ \wedge\ pc = \text{"a"} \\ \qquad \wedge\ \text{IF } low \leq high \wedge result = 0 \\ \qquad\qquad \text{THEN } \wedge\ \text{LET } mid \triangleq (low + high) \div 2 \text{IN} \\ \qquad\qquad\qquad\quad \text{LET } mval \triangleq seq[mid] \text{IN} \end{array}$$

```
                    IF $mval = val$
                       THEN $\land result' = mid$
                            $\land$ UNCHANGED $\langle low, high \rangle$
                       ELSE $\land$ IF $val < mval$
                                     THEN $\land high' = mid - 1$
                                          $\land low' = low$
                                     ELSE $\land low' = mid + 1$
                                          $\land high' = high$
                            $\land$ UNCHANGED $result$
               $\land pc' =$ "a"
          ELSE $\land pc' =$ "Done"
               $\land$ UNCHANGED $\langle low, high, result \rangle$
     $\land$ UNCHANGED $\langle seq, val \rangle$

$Next \triangleq a$
          $\lor$ Disjunct to prevent deadlock on termination
            $(pc =$ "Done" $\land$ UNCHANGED $vars)$

$Spec \triangleq \land Init \land \Box[Next]_{vars}$
           $\land \mathrm{WF}_{vars}(Next)$

$Termination \triangleq \Diamond(pc =$ "Done"$)$
```

END TRANSLATION

Partial correctness of the algorithm is expressed by invariance of formula *resultCorrect*. To get *TLC* to check this property, we use a model that overrides the definition of *Seq* so *Seq*(S) is the set of sequences of elements of S having at most some small length. For example,

$$Seq(S) \triangleq \text{UNION } \{[1 \,..\, i \rightarrow S] : i \in 0 \,..\, 3\}$$

is the set of such sequences with length at most 3.

```
$resultCorrect \triangleq$
   $(pc =$ "Done"$) \Rightarrow$ IF $\exists\, i \in 1 \,..\, Len(seq) : seq[i] = val$
                            THEN $seq[result] = val$
                            ELSE $result = 0$
```

Proving the invariance of *resultCorrect* requires finding an inductive invariant that implies it. A suitable inductive invariant *Inv* is defined here. You can use *TLC* to check that *Inv* is an inductive invariant.

```
$TypeOK \triangleq \land seq \in SortedSeqs$
             $\land val \in Values$
             $\land low \in 1 \,..\, (Len(seq) + 1)$
             $\land high \in 0 \,..\, Len(seq)$
             $\land result \in 0 \,..\, Len(seq)$
             $\land pc \in \{$"a", "Done"$\}$

$Inv \triangleq \land TypeOK$
        $\land (result \neq 0) \Rightarrow (Len(seq) > 0) \land (seq[result] = val)$
```

```
$\land (pc = \text{“a”}) \Rightarrow$
    IF $\exists\, i \in 1\,..\,Len(seq) : seq[i] = val$
      THEN $\exists\, i \in low\,..\,high : seq[i] = val$
      ELSE $result = 0$
$\land (pc = \text{“Done”}) \Rightarrow (result \neq 0) \lor (\forall\, i \in 1\,..\,Len(seq) : seq[i] \neq val)$
```

Here is the invariance proof.

```
THEOREM $Spec \Rightarrow \Box resultCorrect$
$\langle 1\rangle 1.$ $Init \Rightarrow Inv$
  BY DEF $Init$, $Inv$, $TypeOK$
$\langle 1\rangle 2.$ $Inv \land [Next]_{vars} \Rightarrow Inv'$
  $\langle 2\rangle$ SUFFICES ASSUME $Inv$,
                          $[Next]_{vars}$
                   PROVE  $Inv'$
    OBVIOUS
  $\langle 2\rangle 1.$CASE $a$
    $\langle 3\rangle 1.$CASE $low \leq high \land result = 0$
      $\langle 4\rangle$ DEFINE $mid \triangleq (low + high) \div 2$
                $mval \triangleq seq[mid]$
      $\langle 4\rangle$ $(low \leq mid) \land (mid \leq high) \land (mid \in 1\,..\,Len(seq))$
        BY $\langle 3\rangle 1$, $Z3$ DEF $Inv$, $TypeOK$
      $\langle 4\rangle 1.$ $TypeOK'$
        $\langle 5\rangle 1.$ $seq' \in SortedSeqs$
          BY $\langle 2\rangle 1$ DEF $a$, $Inv$, $TypeOK$
        $\langle 5\rangle 2.$ $val' \in Values$
          BY $\langle 2\rangle 1$ DEF $a$, $Inv$, $TypeOK$
        $\langle 5\rangle 3.$ $(low \in 1\,..\,(Len(seq) + 1))'$
          $\langle 6\rangle 1.$CASE $seq[mid] = val$
            BY $\langle 6\rangle 1$, $\langle 2\rangle 1$, $\langle 3\rangle 1$, $Z3$ DEF $Inv$, $TypeOK$, $a$
          $\langle 6\rangle 2.$CASE $seq[mid] \neq val$
            BY $\langle 6\rangle 2$, $\langle 2\rangle 1$, $\langle 3\rangle 1$, $Z3$ DEF $Inv$, $TypeOK$, $a$
          $\langle 6\rangle 3.$ QED
            BY $\langle 6\rangle 1$, $\langle 6\rangle 2$
        $\langle 5\rangle 4.$ $(high \in 0\,..\,Len(seq))'$
          $\langle 6\rangle 1.$CASE $seq[mid] = val$
            BY $\langle 6\rangle 1$, $\langle 2\rangle 1$, $\langle 3\rangle 1$, $Z3$ DEF $Inv$, $TypeOK$, $a$
          $\langle 6\rangle 2.$CASE $seq[mid] \neq val$
            BY $\langle 6\rangle 2$, $\langle 2\rangle 1$, $\langle 3\rangle 1$, $Z3$ DEF $Inv$, $TypeOK$, $a$
          $\langle 6\rangle 3.$ QED
            BY $\langle 6\rangle 1$, $\langle 6\rangle 2$
        $\langle 5\rangle 5.$ $(result \in 0\,..\,Len(seq))'$
          $\langle 6\rangle 1.$CASE $seq[mid] = val$
            BY $\langle 6\rangle 1$, $\langle 2\rangle 1$, $\langle 3\rangle 1$, $Z3$ DEF $Inv$, $TypeOK$, $a$
          $\langle 6\rangle 2.$CASE $seq[mid] \neq val$
            BY $\langle 6\rangle 2$, $\langle 2\rangle 1$, $\langle 3\rangle 1$, $Z3$ DEF $Inv$, $TypeOK$, $a$
    $\langle 6\rangle 3.$ QED
      BY $\langle 6\rangle 1, \langle 6\rangle 2$
  $\langle 5\rangle 6.$ $(pc \in \{\text{"a"}, \text{"Done"}\})'$
    BY $\langle 2\rangle 1, \langle 3\rangle 1$  DEF $Inv, TypeOK, a$
  $\langle 5\rangle 7.$ QED
    BY $\langle 5\rangle 1, \langle 5\rangle 2, \langle 5\rangle 3, \langle 5\rangle 4, \langle 5\rangle 5, \langle 5\rangle 6$  DEF $TypeOK$
$\langle 4\rangle 2.$ $((result \neq 0) \Rightarrow (Len(seq) > 0) \land (seq[result] = val))'$
  $\langle 5\rangle 1.$CASE $seq[mid] = val$
    BY $\langle 5\rangle 1, \langle 2\rangle 1, \langle 3\rangle 1$  DEF $Inv, TypeOK, a$
  $\langle 5\rangle 2.$CASE $seq[mid] \neq val$
    BY $\langle 5\rangle 2, \langle 2\rangle 1, \langle 3\rangle 1$  DEF $Inv, TypeOK, a$
  $\langle 5\rangle 3.$ QED
    BY $\langle 5\rangle 1, \langle 5\rangle 2$
$\langle 4\rangle 3.$ $((pc = \text{"a"}) \Rightarrow$
          IF $\exists\, i \in 1 \,..\, Len(seq) : seq[i] = val$
             THEN $\exists\, i \in low \,..\, high : seq[i] = val$
             ELSE  $result = 0)'$
  $\langle 5\rangle 1.$CASE $seq[mid] = val$
    BY $\langle 5\rangle 1, \langle 2\rangle 1, \langle 3\rangle 1$  DEF $Inv, TypeOK, a$
  $\langle 5\rangle 2.$CASE $seq[mid] \neq val$
    $\langle 6\rangle 1.$ $\land\, (low \leq mid) \,\land\, (mid \leq high) \land (mid \in 1 \,..\, Len(seq))$
          $\land\, Len(seq) > 0 \land Len(seq) \in Nat$
          $\land\, low \in 1 \,..\, Len(seq)$
          $\land\, high \in 1 \,..\, Len(seq)$
      BY $ValAssump$  DEF $Inv, TypeOK$
    $\langle 6\rangle 2.$CASE $\exists\, i \in 1 \,..\, Len(seq) : seq[i] = val$
      $\langle 7\rangle 1.$ PICK $i \in low \,..\, high : seq[i] = val$
        BY $\langle 6\rangle 2, \langle 2\rangle 1$  DEF $a, Inv$
      $\langle 7\rangle 2.$ $\land\, (low \leq mid) \,\land\, (mid \leq high) \land (mid \in 1 \,..\, Len(seq))$
            $\land\, Len(seq) > 0 \land Len(seq) \in Nat$
            $\land\, low \in 1 \,..\, Len(seq)$
            $\land\, high \in 1 \,..\, Len(seq)$
            $\land\, seq[i] = val$
        BY $ValAssump, \langle 6\rangle 2, \langle 7\rangle 1$  DEF $Inv, TypeOK$
      $\langle 7\rangle 3.$ $\forall\, j \in 1 \,..\, Len(seq) : seq[j] \in Int$
        $\langle 8\rangle 1.$ $seq \in Seq(Values)$
          BY   DEF $Inv, TypeOK, SortedSeqs$
        $\langle 8\rangle 2.$ $seq \in Seq(Int)$
          BY $\langle 8\rangle 1, ValAssump$
        $\langle 8\rangle 3.$ QED
          BY $\langle 8\rangle 2$  DEF $Inv, TypeOK, SortedSeqs$
      $\langle 7\rangle 4.$ $\forall\, j, k \in 1 \,..\, Len(seq) : j < k \Rightarrow seq[j] \leq seq[k]$
          BY  DEF $Inv, TypeOK, SortedSeqs$
      $\langle 7\rangle 5.$CASE $val < seq[mid]$
        $\langle 8\rangle 1.$ $seq[i] \; < seq[mid]$
```

```
               BY ⟨7⟩2, ⟨7⟩5 , ⟨8⟩5
              ⟨8⟩2. $i < mid$
                BY $ValAssump$, ⟨7⟩2, ⟨8⟩1, ⟨7⟩4, ⟨7⟩3, $Z3$
              ⟨8⟩3. $i \in low \,..\, mid - 1$
                BY ONLY ⟨7⟩2, ⟨8⟩1, ⟨8⟩2, $Z3$
              ⟨8⟩4. $\wedge (pc' = \text{“a”}) \wedge (low' = low) \wedge (high' = mid - 1)$
                    $\wedge \exists j \in 1 \,..\, Len(seq) : seq[j] = val$
                BY ⟨2⟩1, ⟨3⟩1, ⟨5⟩2, ⟨6⟩2, ⟨7⟩5 DEF $a$, $mid$
              ⟨8⟩5. QED
               BY ONLY ⟨7⟩2, ⟨8⟩4, ⟨8⟩3 , ⟨8⟩5
            ⟨7⟩6.CASE $\neg(val < seq[mid])$
              ⟨8⟩ HIDE DEF $mid$
              ⟨8⟩1. $seq[mid] < seq[i]$
                  BY $ValAssump$, ⟨7⟩2, ⟨7⟩6, ⟨5⟩2, ⟨7⟩3, $Z3$
              ⟨8⟩2. $mid < i$
                BY $ValAssump$, ⟨7⟩2, ⟨8⟩1, ⟨8⟩a, ⟨9⟩1,⟨7⟩3, ⟨7⟩4, $Z3$
              ⟨8⟩3. $i \in mid + 1 \,..\, high$
                BY ⟨7⟩2, ⟨8⟩1, ⟨8⟩2, $Z3$
              ⟨8⟩4. $\wedge (pc' = \text{“a”}) \wedge (low' = mid + 1) \wedge (high' = high)$
                    $\wedge \exists j \in 1 \,..\, Len(seq) : seq[j] = val$
                BY ⟨2⟩1, ⟨3⟩1, ⟨5⟩2, ⟨6⟩2, ⟨7⟩6 DEF $a$, $mid$
              ⟨8⟩5. QED
               BY ONLY ⟨7⟩2, ⟨8⟩4, ⟨8⟩3 , ⟨8⟩5
            ⟨7⟩7. QED
              BY ⟨7⟩5, ⟨7⟩6
          ⟨6⟩3.CASE $\neg\exists i \in 1 \,..\, Len(seq) : seq[i] = val$
            BY ⟨6⟩3, ⟨5⟩2, ⟨2⟩1, ⟨3⟩1 DEF $Inv$, $TypeOK$, $a$
          ⟨6⟩4. QED
            BY ⟨6⟩2, ⟨6⟩3
        ⟨5⟩3. QED
          BY ⟨5⟩1, ⟨5⟩2
      ⟨4⟩4. $((pc = \text{“Done”}) \Rightarrow (result \neq 0) \vee (\forall i \in 1 \,..\, Len(seq) : seq[i] \neq val))'$
        BY ⟨3⟩1, ⟨2⟩1 DEF $Inv$, $TypeOK$, $a$
      ⟨4⟩5. QED
        BY ⟨4⟩1, ⟨4⟩2, ⟨4⟩3, ⟨4⟩4 DEF $Inv$
    ⟨3⟩2.CASE $\neg(low \leq high \wedge result = 0)$
      BY ⟨3⟩2, ⟨2⟩1 DEF $Inv$, $TypeOK$, $a$
    ⟨3⟩3. QED
      BY ⟨3⟩1, ⟨3⟩2
  ⟨2⟩2.CASE UNCHANGED $vars$
    BY ⟨2⟩2 DEF $Inv$, $TypeOK$, $vars$
  ⟨2⟩3. QED
    BY ⟨2⟩1, ⟨2⟩2 DEF $Next$
⟨1⟩3. $Inv \Rightarrow resultCorrect$
  BY DEF $resultCorrect$, $Inv$, $TypeOK$
```

$\langle 1 \rangle 4.$ QED
  BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, $\langle 1 \rangle 3$, PTL DEF $Spec$

* Modification History
* Last modified *Fri* May 03 16:28:58 *PDT* 2019 by *lamport*
* Created *Wed Apr* 17 15:15:12 *PDT* 2019 by *lamport*