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* $\triangleq$ $Values \subseteq Int$

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

LEMMA *SortedLess* $\triangleq$
ASSUME NEW $s \in SortedSeqs$, NEW $i \in 1 \,..\, Len(s)$, NEW $j \in 1 \,..\, Len(s)$,
$s[i] < s[j]$
PROVE $i < j$
$\langle 1 \rangle$.SUFFICES ASSUME $j \leq i$PROVE FALSE
OBVIOUS
$\langle 1 \rangle$.QED BY *ValAssump* DEF *SortedSeqs*

****

--fair algorithm *BinarySearch* **{**
variables $seq \in SortedSeqs$, $val \in Values$,
$low = 1$, $high = Len(seq)$, $result = 0$ **;**
{ a: **while (** $low \leq high \land result = 0$ **) {**
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
$\land\ seq \in SortedSeqs$
$\land\ val \in Values$
$\land\ low = 1$

$$
\begin{array}{l}
\land\ high = Len(seq) \\
\land\ result = 0 \\
\land\ pc = \text{"a"}
\end{array}
$$

$$
\begin{array}{rl}
a \triangleq & \land\ pc = \text{"a"} \\
& \land\ \textsc{if}\ low \leq high \land result = 0 \\
& \quad \textsc{then}\ \land\ \textsc{let}\ mid \triangleq (low + high) \div 2\ \textsc{in} \\
& \qquad\qquad \textsc{let}\ mval \triangleq seq[mid]\ \textsc{in} \\
& \qquad\qquad\ \textsc{if}\ mval = val \\
& \qquad\qquad\quad \textsc{then}\ \land\ result' = mid \\
& \qquad\qquad\qquad\quad \land\ \textsc{unchanged}\ \langle low, high \rangle \\
& \qquad\qquad\quad \textsc{else}\ \land\ \textsc{if}\ val < mval \\
& \qquad\qquad\qquad\qquad \textsc{then}\ \land\ high' = mid - 1 \\
& \qquad\qquad\qquad\qquad\qquad \land\ low' = low \\
& \qquad\qquad\qquad\qquad \textsc{else}\ \land\ low' = mid + 1 \\
& \qquad\qquad\qquad\qquad\qquad \land\ high' = high \\
& \qquad\qquad\qquad\quad \land\ \textsc{unchanged}\ result \\
& \qquad\quad \land\ pc' = \text{"a"} \\
& \quad \textsc{else}\ \land\ pc' = \text{"Done"} \\
& \qquad\quad \land\ \textsc{unchanged}\ \langle low, high, result \rangle \\
& \land\ \textsc{unchanged}\ \langle seq, val \rangle
\end{array}
$$

Allow infinite stuttering to prevent deadlock on termination.

$Terminating \triangleq pc = \text{"Done"} \land \textsc{unchanged}\ vars$

$$
\begin{array}{rl}
Next \triangleq & a \\
& \lor\ Terminating
\end{array}
$$

$$
\begin{array}{rl}
Spec \triangleq & \land\ Init \land \Box[Next]_{vars} \\
& \land\ \mathrm{WF}_{vars}(Next)
\end{array}
$$

$Termination \triangleq \Diamond(pc = \text{"Done"})$

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,

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

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

$$
\begin{array}{l}
resultCorrect \triangleq \\
\quad (pc = \text{"Done"}) \Rightarrow \textsc{if}\ \exists\, i \in 1\,..\,Len(seq) : seq[i] = val \\
\qquad\qquad\qquad\qquad \textsc{then}\ seq[result] = val \\
\qquad\qquad\qquad\qquad \textsc{else}\ result = 0
\end{array}
$$

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.

$$
\begin{aligned}
TypeOK \triangleq\ & \land\ seq \in SortedSeqs \\
& \land\ val \in Values \\
& \land\ low \in 1 \,..\, (Len(seq)+1) \\
& \land\ high \in 0 \,..\, Len(seq) \\
& \land\ result \in 0 \,..\, Len(seq) \\
& \land\ pc \in \{\text{“a”}, \text{“Done”}\}
\end{aligned}
$$

$$
\begin{aligned}
Inv \triangleq\ & \land\ TypeOK \\
& \land\ (result \neq 0) \Rightarrow (Len(seq) > 0) \land (seq[result] = val) \\
& \land\ (pc = \text{“a”}) \Rightarrow \\
& \qquad \text{IF } \exists\, i \in 1 \,..\, Len(seq) : seq[i] = val \\
& \qquad\quad \text{THEN } \exists\, i \in low \,..\, high : seq[i] = val \\
& \qquad\quad \text{ELSE } result = 0 \\
& \land\ (pc = \text{“Done”}) \Rightarrow (result \neq 0) \lor (\forall\, i \in 1 \,..\, Len(seq) : seq[i] \neq val)
\end{aligned}
$$

Here is the invariance proof.

THEOREM $Spec \Rightarrow \Box resultCorrect$

$\langle 1\rangle 1.\ Init \Rightarrow Inv$

  BY DEF $Init, Inv, TypeOK, SortedSeqs$

$\langle 1\rangle 2.\ Inv \land [Next]_{vars} \Rightarrow Inv'$

  $\langle 2\rangle$ SUFFICES ASSUME Inv, $[Next]_{vars}$ PROVE Inv'

    OBVIOUS

  $\langle 2\rangle 1.$CASE a

    $\langle 3\rangle.$UNCHANGED $\langle seq, val\rangle$

      BY $\langle 2\rangle 1$ DEF 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 $Inv, TypeOK, SortedSeqs$

      $\langle 4\rangle 1.\ TypeOK'$

        $\langle 5\rangle 1.\ seq' \in SortedSeqs$

          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 $Inv, TypeOK, a, SortedSeqs$

$\langle 6\rangle 3.$ 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 $Inv, TypeOK, a, SortedSeqs$
$\langle 6\rangle 3.$ 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 $seq[mid] \neq val$
$\langle 6\rangle 1.$ $\land\, Len(seq) > 0$ $\land\, Len(seq) \in Nat$
$\land\, low \in 1 \,..\, Len(seq)$
$\land\, high \in 1 \,..\, Len(seq)$
BY $ValAssump$ DEF $Inv, TypeOK, SortedSeqs$
$\langle 6\rangle 2.$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
$\langle 7\rangle 2.$ $\land\, Len(seq) > 0 \land Len(seq) \in Nat$
$\land\, low \in 1 \,..\, Len(seq)$
$\land\, high \in 1 \,..\, Len(seq)$
$\land\, seq[i] = val$

BY $ValAssump$, $\langle 6\rangle 2$, $\langle 7\rangle 1$ DEF Inv, $TypeOK$, $SortedSeqs$

$\langle 7\rangle 3.$ $\forall j \in 1 \, .. \, Len(seq) : seq[j] \in Int$

BY $ValAssump$ DEF Inv, $TypeOK$, $SortedSeqs$

$\langle 7\rangle 4.$CASE $val < seq[mid]$

$\langle 8\rangle 1.$ $seq[i] \ < seq[mid]$

BY $\langle 7\rangle 2$, $\langle 7\rangle 4$

$\langle 8\rangle 2.$ $i < mid$

BY $\langle 7\rangle 2$, $\langle 8\rangle 1$, $SortedLess$ DEF Inv, $TypeOK$

$\langle 8\rangle 3.$ $i \in low \, .. \, mid - 1$

BY ONLY $\langle 7\rangle 2$, $\langle 8\rangle 1$, $\langle 8\rangle 2$, $Z3$

$\langle 8\rangle 4.$ $\land (pc' = \text{“a”}) \land (low' = low) \land (high' = mid - 1)$
$\land \exists j \ \in 1 \, .. \, Len(seq) : seq[j] = val$

BY $\langle 2\rangle 1$, $\langle 3\rangle 1$, $\langle 5\rangle 2$, $\langle 6\rangle 2$, $\langle 7\rangle 4$ DEF a, mid

$\langle 8\rangle$.QED

BY $\langle 7\rangle 2$, $\langle 8\rangle 4$, $\langle 8\rangle 3$

$\langle 7\rangle 5.$CASE $\neg(val < seq[mid])$

$\langle 8\rangle$ HIDE DEF mid

$\langle 8\rangle 1.$ $seq[mid] < seq[i]$

BY $ValAssump$, $\langle 7\rangle 2$, $\langle 7\rangle 5$, $\langle 5\rangle 2$, $\langle 7\rangle 3$, $Z3$

$\langle 8\rangle 2.$ $mid < i$

BY $\langle 7\rangle 2$, $\langle 8\rangle 1$, $SortedLess$ DEF Inv, $TypeOK$

$\langle 8\rangle 3.$ $i \in mid + 1 \, .. \, high$

BY $\langle 7\rangle 2$, $\langle 8\rangle 1$, $\langle 8\rangle 2$, $Z3$

$\langle 8\rangle 4.$ $\land (pc' = \text{“a”}) \land (low' = mid + 1) \land (high' = high)$
$\land \exists j \ \in 1 \, .. \, Len(seq) : seq[j] = val$

BY $\langle 2\rangle 1$, $\langle 3\rangle 1$, $\langle 5\rangle 2$, $\langle 6\rangle 2$, $\langle 7\rangle 5$ DEF a, mid

$\langle 8\rangle 5.$ QED

BY $\langle 7\rangle 2$, $\langle 8\rangle 4$, $\langle 8\rangle 3$, $\langle 8\rangle 5$

$\langle 7\rangle 7.$ QED

BY $\langle 7\rangle 4$, $\langle 7\rangle 5$

$\langle 6\rangle 3.$CASE $\neg \exists i \in 1 \, .. \, Len(seq) : seq[i] = val$

BY $\langle 6\rangle 3$, $\langle 5\rangle 2$, $\langle 2\rangle 1$, $\langle 3\rangle 1$ DEF Inv, $TypeOK$, a

$\langle 6\rangle 4.$ QED

BY $\langle 6\rangle 2$, $\langle 6\rangle 3$

$\langle 5\rangle 3.$ QED

BY $\langle 5\rangle 1$, $\langle 5\rangle 2$

$\langle 4\rangle 4.$ $((pc = \text{“Done”}) \Rightarrow (result \neq 0) \lor (\forall i \in 1 \, .. \, Len(seq) : seq[i] \neq val))'$

BY $\langle 3\rangle 1$, $\langle 2\rangle 1$ DEF Inv, $TypeOK$, a

$\langle 4\rangle 5.$ QED

BY $\langle 4\rangle 1$, $\langle 4\rangle 2$, $\langle 4\rangle 3$, $\langle 4\rangle 4$ DEF Inv

$\langle 3\rangle 2.$CASE $\neg(low \leq high \land result = 0)$

BY $\langle 3\rangle 2$, $\langle 2\rangle 1$ DEF Inv, $TypeOK$, a

$\langle 3\rangle 3.$ QED

BY $\langle 3\rangle 1$, $\langle 3\rangle 2$

$\langle 2\rangle 2.$CASE UNCHANGED $vars$

    BY $\langle 2\rangle 2$ DEF Inv, $TypeOK$, $vars$
  $\langle 2\rangle 3.$ QED
    BY $\langle 2\rangle 1$, $\langle 2\rangle 2$ DEF $Next$, $Terminating$
$\langle 1\rangle 3.$ $Inv \Rightarrow resultCorrect$
  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 $Spec$