QuickSort Partition Code — Annotated

Jonathan Buss, Collin Roberts

Algorithm and Code

Overview of the Code

Array to partition: X, indexed from 1 to n.
Pivot: p. It may have any value.

First `while`-loop:
: Positions the a marker to the first element which is greater than p. If no such element exists, a becomes n, and the second while-loop will skip its body.

Second `while`-loop:
: Move b marker over any remaining elements, and swap an out-of-position one into place. After a swap, we move the a marker one position to the right (a large element moved out and a small one into its former place).

Postcondition:
: z is the cutoff position between the "small" elements on the left and the "large" elements on the right.

Code (unannotated)

$(\!|\, n \geq 1 \,|\!)$

```
a = 1 ;
while ( a < n && X[a] <= p ) {
   a = a + 1 ;
}
b = a + 1 ;
while ( b <= n ) {
  if ( X[b] <= p ) {
    t = X[b] ; X[b] = X[a] ; X[a] = t ;
    a = a + 1 ;
  }
  b = b + 1 ;
}
```

$(\!|\, \exists z \cdot (1 \leq z \leq n+1 \wedge X[1..z) \leq p \wedge (X[z..n] > p)) \,|\!)$

We shall annotate this code in two parts—one for each of the `while`-loops. To complete the annotation then requires additional implied conditions.

Annotations

Notes on Notation:

- "$X[a..b)$" refers to all $X[i]$, where $a \le i < b$. Similarly, $X[a..b]$ refers to all $X[i]$, where $a \le i \le b$. (This is analogous to the notation for open and closed intervals in calculus.)
- The various parts of formulas are coloured to show their origin and role.

  Greenish: the "lower" (or only) part of a loop invariant.

  Bluish: the "upper" part of a loop invariant, if any.

  Reddish: Conditions from loop guards and conditionals.

  Before an assignment, the assigned value has its colour reddened.

Preamble and first loop (invariant $1 \le a \wedge X[1..a) \le p$)

$\llparenthesis\, (1 \le 1 \le n) \wedge X[1..1) \le p \,\rrparenthesis$	⟨*precondition*⟩
`a = 1 ;`	
$\llparenthesis\, (1 \le a \le n) \wedge X[1..a) \le p \,\rrparenthesis$	assignment
`while ( a < n && X[a] <= p )` $\{$	
$\quad\llparenthesis\, (1 \le a \le n) \wedge X[1..a) \le p \wedge (a < n \wedge X[a] \le p) \,\rrparenthesis$	partial-**`while`**
$\quad\llparenthesis\, (1 \le a+1 \le n) \wedge X[1..a+1) \le p \,\rrparenthesis$	implied
`  a = a + 1 ;`	
$\quad\llparenthesis\, (1 \le a \le n) \wedge X[1..a) \le p \,\rrparenthesis$	assignment
$\}$	
$\llparenthesis\, (1 \le a \le n) \wedge X[1..a) \le p \wedge (a \ge n \vee X[a] > p) \,\rrparenthesis$	partial-**`while`**

Remarks:

- The implication involves a notational shift: $X[1..a+1) \le p$ is equivalent to $X[1..a) \le p \wedge X[a] \le p$.

Second loop (invariant $(1 \le a \le n) \wedge X[1..a) \le p \wedge X[a..\min(b,n)) > p$)

$(\!|\ X[1..a) \le p \wedge X[a..\min(a+1,n)) > p\ |\!)$	⟨*precondition*⟩
`b = a + 1 ;`	
$(\!|\ X[1..a) \le p \wedge X[a..\min(b,n)) > p\ |\!)$	assignment
`while ( b <= n )` $\{$	
$(\!|\ \big(X[1..a) \le p \wedge X[a..\min(b,n)) > p\big) \wedge b \le n\ |\!)$	partial-**`while`**
$(\!|\ X[1..a) \le p \wedge X[a..b) > p \wedge b \le n\ |\!)$	implied
`if ( X[b] <= p )` $\{$	
$(\!|\ X[1..a) \le p \wedge X[a..b) > p \wedge b \le n \wedge X[b] \le p\ |\!)$	**`if-then`**
$(\!|\ X[1..a) \le p \wedge X[a] > p \wedge X[a+1..b) > p \wedge X[b] \le p\ |\!)$	implied
`t = X[b] ; X[b] = X[a] ; X[a] = t ;`	
$(\!|\ X[1..a) \le p \wedge X[a] \le p \wedge X[a+1..b) > p \wedge X[b] > p\ |\!)$	swap
$(\!|\ X[1..a+1) \le p \wedge X[a+1..b) > p\ |\!)$	implied
`a = a + 1 ;`	
$(\!|\ X[1..a) \le p \wedge X[a..b+1) > p\ |\!)$	assignment
$\}$	
$(\!|\ X[1..a) \le p \wedge X[a..b+1) > p\ |\!)$	**`if-then`** $+$ **`implied`**
`b = b + 1`	
$(\!|\ X[1..a) \le p \wedge X[a..b) > p\ |\!)$	assignment
$(\!|\ X[1..a) \le p \wedge X[a..\min(b,n)) > p\ |\!)$	implied
$\}$	
$(\!|\ X[1..a) \le p \wedge X[a..\min(b,n)) > p \wedge b > n\ |\!)$	partial-**`while`**

Remarks:

- The the presence of "$\min(b,n)$" in the loop invariant accounts for the possibility that a advanced to n in the first loop. The relationship of $X[n]$ to p is unknown and immaterial. (We could instead include an additional test in the code, but such a test is not needed for correctness.)

  The adjustment is not needed, and hence disappears, inside of the loop:

  – At the start of the while-loop, we have $b \le n$, and the minimum is always b.

  – At the end of the loop, $X[a..b) > p$ implies $X[a..\min(b,n)) > p$, whatever value b has.

- The condition labelled "swap" follows from the proof in the course notes: the three assignments exchange the values of $X[a]$ and $X[b]$, leaving the rest of the array unchanged.

  The implieds before and after the swap simply re-write the notation, to focus on the changing locations.

Inter-loop implications

After first `while`-loop:

$(\!|\ X[1..a) \le p \wedge (a \ge n \vee X[a] > p)\ |\!)$	partial-**`while`**
$(\!|\ X[1..a) \le p \wedge X[a..\min(a+1,n)) > p\ |\!)$	implied
	(*pre-condition for assignment to* **`b`**)

Algebra and use of notation.

At end:

$(\!|\ X[1..a) \le p \wedge X[a..\min(b,n)) > p \wedge b > n\ |\!)$	partial-**`while`**
$(\!|\ \exists z \cdot (1 \le z \le n+1 \wedge X[1..z) \le p \wedge (X[z..n] > p))\ |\!)$	postcondition

The value for z is a if $a \le n$ and $X[n] > p$, or $n+1$ otherwise.

Full annotated code

$\llparenthesis\, n \geq 1 \,\rrparenthesis$	
$\llparenthesis\, X[1..1) \leq p \,\rrparenthesis$	implied
`a = 1 ;`	
$\llparenthesis\, X[1..a) \leq p \,\rrparenthesis$	assignment
`while ( a < n && X[a] <= p ) {`	
$\quad\llparenthesis\, X[1..a) \leq p \wedge (a < n \wedge X[a] \leq p) \,\rrparenthesis$	partial-**`while`**
$\quad\llparenthesis\, X[1..a+1) \leq p \,\rrparenthesis$	implied
`    a = a + 1 ;`	
$\quad\llparenthesis\, X[1..a) \leq p \,\rrparenthesis$	assignment
`}`	
$\llparenthesis\, X[1..a) \leq p \wedge (a \geq n \vee X[a] > p) \,\rrparenthesis$	partial-**`while`**
$\llparenthesis\, X[1..a) \leq p \wedge X[a..\min(a+1, n)) > p \,\rrparenthesis$	implied
`b = a + 1 ;`	
$\llparenthesis\, X[1..a) \leq p \wedge X[a..\min(b, n)) > p \,\rrparenthesis$	assignment
`while ( b <= n ) {`	
$\quad\llparenthesis\, \big(X[1..a) \leq p \wedge X[a..\min(b, n)) > p\big) \wedge b \leq n \,\rrparenthesis$	partial-**`while`**
$\quad\llparenthesis\, X[1..a) \leq p \wedge X[a..b) > p \wedge b \leq n \,\rrparenthesis$	implied
`  if ( X[b] <= p ) {`	
$\qquad\llparenthesis\, X[1..a) \leq p \wedge X[a..b) > p \wedge b \leq n \wedge X[b] \leq p \,\rrparenthesis$	**`if-then`**
$\qquad\llparenthesis\, X[1..a) \leq p \wedge X[a] > p \wedge X[a+1..b) > p \wedge X[b] \leq p \,\rrparenthesis$	implied
`    t = X[b] ; X[b] = X[a] ; X[a] = t ;`	
$\qquad\llparenthesis\, X[1..a) \leq p \wedge X[a] \leq p \wedge X[a+1..b) > p \wedge X[b] > p \,\rrparenthesis$	swap
$\qquad\llparenthesis\, X[1..a+1) \leq p \wedge X[a+1..b) > p \,\rrparenthesis$	implied
`    a = a + 1 ;`	
$\qquad\llparenthesis\, X[1..a) \leq p \wedge X[a..b+1) > p \,\rrparenthesis$	assignment
`  }`	
$\quad\llparenthesis\, X[1..a) \leq p \wedge X[a..b+1) > p \,\rrparenthesis$	**`if-then`** + **`implied`**
`  b = b + 1`	
$\quad\llparenthesis\, X[1..a) \leq p \wedge X[a..b) > p \,\rrparenthesis$	assignment
$\quad\llparenthesis\, X[1..a) \leq p \wedge X[a..\min(b, n)) > p \,\rrparenthesis$	implied
`}`	
$\llparenthesis\, X[1..a) \leq p \wedge X[a..\min(b, n)) > p \wedge b > n \,\rrparenthesis$	partial-**`while`**
$\llparenthesis\, \exists z \cdot (1 \leq z \leq n+1 \wedge X[1..z) \leq p \wedge (X[z..n] > p)) \,\rrparenthesis$	implied

Remarks:

- The proof of termination (i.e. the last ingredient required for total correctness) is left as an exercise.