CS4004: Assignment 2

Conor McCauley - 17323203

December 13, 2020

Question 1

(i) For each integer to appear at least twice there must exist j, k for each i such that $j \neq k$ and both $s'[j], s'[k]$ are equal to $s[i]$:

$$
\begin{aligned}
&\forall i.(\\
&\quad (0 \leq i < |s|) \rightarrow \exists j.\exists k.(\\
&\qquad (0 \leq j < |s'|) \wedge (0 \leq k < |s'|) \\
&\qquad \wedge (j \neq k) \wedge (s[i] = s'[j]) \wedge (s[i] = s'[k]) \\
&\quad) \\
&)
\end{aligned}
$$

(ii) For no integer to appear more than twice there cannot exist j, k, l for any i such that $j \neq k \neq l$ and all $s'[j], s'[k], s'[l]$ are equal to $s[i]$:

$$
\begin{aligned}
&\forall i.(\\
&\quad (0 \leq i < |s|) \rightarrow \neg\exists j.\exists k.\exists l.(\\
&\qquad (0 \leq j < |s'|) \wedge (0 \leq k < |s'|) \wedge (0 \leq l < |s'|) \\
&\qquad \wedge (j \neq k) \wedge (j \neq l) \\
&\qquad \wedge (s[i] = s'[j]) \wedge (s[i] = s'[k]) \wedge (s[i] = s'[l]) \\
&\quad) \\
&)
\end{aligned}
$$

(iii) For each integer to appear exactly twice then for all i our formula from (i) must be true and our formula from (ii) must not be true:

$$
\begin{aligned}
&\forall i.(\\
&\quad (0 \leq i < |s|) \rightarrow \exists j.\exists k.(\\
&\qquad (0 \leq j < |s'|) \wedge (0 \leq k < |s'|) \\
&\qquad \wedge (j \neq k) \wedge (s[i] = s'[j]) \wedge (s[i] = s'[k]) \\
&\qquad \wedge \neg\exists l.(\\
&\qquad\quad (0 \leq l < |s'|) \wedge (j \neq l) \wedge (k \neq l) \wedge (s[i] = s'[l]) \\
&\qquad) \\
&\quad) \\
&)
\end{aligned}
$$

(iv) We know from the theorems given in lectures that the following holds:

$$X \to Y \equiv \neg Y \to \neg X$$

Therefore, if we express an integer being in s' as X and an integer being in s as Y we can refer to the statement given in the question as $\neg Y \to \neg X$. From the above theorem we can then say that this is equivalent to the statement: $X \to Y$ or if an integer is in s' then it is also in s. We can express this new statement like so:

$$\begin{aligned}
&\forall i.(\\
&\quad (0 \leq i < |s'|) \to \exists j.(\\
&\qquad (0 \leq j \leq |s|) \land (s'[i] = s[j]) \\
&\quad) \\
&)
\end{aligned}$$

Question 2

(a) For this statement to be valid then lo must be within the bounds of s $(0 \leq lo < |s|)$ and all elements in s must be less than or equal to $s[lo]$. We can express these two conditions like so:

$$(0 \leq lo < |s|) \land \forall i.((0 \leq i < |s|) \to (s[i] \leq s[lo]))$$

(b) The while loop has an invariant of $0 \leq lo \leq hi < |s|$ and a variant of $hi - lo$:

$\{0 < \|s\|\}$	
`lo := 0`	
$\{0 \le 0\}$	impl
$\{0 \le lo\}$	asg
$\{lo < \|s\|\}$	asg
$\{lo \le \|s\| - 1\}$	impl
`hi := |s| - 1`	
$\{\|s\| - 1 \le \|s\| - 1\}$	impl
$\{hi \le \|s\| - 1\}$	asg
$\{0 \le lo \le hi < \|s\|\}$	asg
`while (lo < hi) {`	
$\quad\{(0 \le lo \le hi < \|s\|) \wedge (lo < hi) \wedge (0 \le hi - lo = E_0)\}$	while
`    if (s[lo] <= s[hi]) {`	
$\quad\quad\{0 < hi - lo = E_0\}$	impl
$\quad\quad\{0 \le hi - (lo + 1) < E_0\}$	impl
$\quad\quad\{lo + 1 \le hi\}$	impl
`        lo := lo + 1`	
$\quad\quad\{0 \le hi - lo < E_0\}$	asg
$\quad\quad\{lo \le hi\}$	asg
$\quad\quad\{0 \le lo \le hi < \|s\|\}$	impl
`    } else {`	
$\quad\quad\{0 < hi - lo = E_0\}$	impl
$\quad\quad\{0 \le (hi - 1) - lo < E_0\}$	impl
$\quad\quad\{lo \le hi - 1\}$	impl
`        hi := hi - 1`	
$\quad\quad\{0 \le hi - lo < E_0\}$	asg
$\quad\quad\{lo \le hi\}$	asg
$\quad\quad\{0 \le lo \le hi < \|s\|\}$	impl
`    }`	
$\quad\{(0 \le lo \le hi < \|s\|) \wedge (0 \le hi - lo < E_0)\}$	cond
`}`	
$\{(0 \le lo \le hi < \|s\|) \wedge \neg(lo < hi)\}$	while
$\{\top\}$	impl

(c) The while loop has the following invariant:

$$(0 \le lo \le hi < |s|) \wedge (isMax(s[..(lo+1)] + s[hi..], lo) \vee isMax(s[..(lo+1)] + s[hi..], hi))$$

$\{0 < |s|\}$
`lo := 0`
$\{0 \leq 0\}$ impl
$\{0 \leq lo\}$ asg
$\{lo < |s|\}$ asg
$\{lo \leq |s| - 1\}$ impl
`hi := |s| - 1`
$\{|s| - 1 \leq |s| - 1\}$ impl
$\{hi \leq |s| - 1\}$ asg
$\{0 \leq lo \leq hi < |s|\}$ asg
$\{isMax(s[..1] + s[(|s|-1)..], 0) \lor isMax(s[..1] + s[(|s|-1)..], |s|-1)\}$ impl
$\{isMax(s[..(lo+1)] + s[hi..], lo) \lor isMax(s[..(lo+1)] + s[hi..], hi)\}$ asg
$\{(0 \leq lo \leq hi < |s|) \land isMax(s[..(lo+1)], lo) \land isMax(s[hi..], hi)\}$ impl
`while (lo < hi) {`
$\{(0 \leq lo \leq hi < |s|) \land (isMax(s[..(lo+1)] + s[hi..], lo) \lor isMax(s[..(lo+1)] + s[hi..], hi))$
$\land (lo < hi)\}$ while
`if (s[lo] <= s[hi]) {`
$\{s[lo] <= s[hi]\}$ cond
$\{isMax(s[..(lo+1)] + s[hi..], hi)\}$ impl
$\{isMax(s[..(lo+2)] + s[hi..], lo) \lor isMax(s[..(lo+2)] + s[hi..], hi)\}$ impl
$\{lo + 1 \leq hi\}$ impl
`lo := lo + 1`
$\{lo \leq hi\}$ asg
$\{0 \leq lo \leq hi < |s|\}$ impl
$\{isMax(s[..(lo+1)] + s[hi..], lo) \lor isMax(s[..(lo+1)] + s[hi..], hi)\}$ asg
$\{(isMax(s[..(lo+1)] + s[hi..], lo) \lor isMax(s[..(lo+1)] + s[hi..], hi))$
$\land (0 \leq lo \leq hi < |s|)\}$ impl
`} else {`
$\{\neg(s[lo] <= s[hi])\}$ cond
$\{isMax(s[..(lo+1)] + s[hi..], lo)\}$ impl
$\{isMax(s[..(lo+1)] + s[(hi-1)..], lo) \lor isMax(s[..(lo+1)] + s[(hi-1)..], hi)\}$ impl
$\{lo \leq hi - 1\}$ impl
`hi := hi - 1`
$\{lo \leq hi\}$ asg
$\{0 \leq lo \leq hi < |s|\}$ impl
$\{isMax(s[..(lo+1)] + s[hi..], lo) \lor isMax(s[..(lo+1)] + s[hi..], hi)\}$ asg
$\{(isMax(s[..(lo+1)] + s[hi..], lo) \lor isMax(s[..(lo+1)] + s[hi..], hi))$
$\land (0 \leq lo \leq hi < |s|)\}$ impl
`}`
$\{(0 \leq lo \leq hi < |s|) \land (isMax(s[..(lo+1)] + s[hi..], lo) \lor isMax(s[..(lo+1)] + s[hi..], hi))\}$ cond
`}`
$\{(0 \leq lo \leq hi < |s|) \land (isMax(s[..(lo+1)] + s[hi..], lo) \lor isMax(s[..(lo+1)] + s[hi..], hi))$
$\land \neg(lo < hi)\}$ while
$\{lo = hi\}$ impl
$\{isMax(s[..(lo+1)] + s[lo..], lo) \lor isMax(s[..(lo+1)] + s[lo..], lo)\}$ asg
$\{isMax(s, lo)\}$ impl

Question 3

(a) For this statement to be valid then each integer at index i should be equal to the integer at index $|s| - i - 1$. We can express this condition like so:

$$\forall i.((0 \leq i < |s|) \rightarrow (s[i] = s[|s| - i - 1))$$

(b) The while loop has an invariant of $0 \leq i \land j < |s|$ and a variant of $j - i + 1$:

$\{0 \leq |s|\}$

`res := 1`

`i := 0`

$\{0 \leq 0\}$ impl

$\{0 \leq i\}$ asg

`j := |s| - 1`

$\{|s| - 1 < |s|\}$ impl

$\{j < |s|\}$ asg

$\{0 \leq i \wedge j < |s|\}$ impl

`while (i < j & res = 1) {`

$\{(0 \leq i \wedge j < |s|) \wedge i < j \wedge res = 1\}$ while

$\{0 < j - i\}$ impl

$\{0 < j - i + 1\}$ impl

$\{0 \leq j - i + 1 = E_0\}$ impl

`if (s[i] != s[j]) {`

$\{0 \leq j - i + 1 = E_0 \wedge s[i] \neq s[j]\}$ cond

`res := 0`

$\{0 \leq j - i + 1 = E_0\}$ impl

`} else {`

$\{0 \leq j - i + 1 = E_0 \wedge \neg(s[i] \neq s[j])\}$ cond

`skip`

$\{0 \leq j - i + 1 = E_0\}$ impl

`}`

$\{0 < j - i + 1 = E_0\}$ impl

$\{0 < j - i < E_0\}$ impl

$\{0 \leq (j - 1) - (i + 1) + 1 < E_0\}$ impl

$\{0 \leq i + 1\}$ impl

`i := i + 1`

$\{0 \leq i\}$ asg

$\{0 \leq (j - 1) - i + 1 < E_0\}$ asg

$\{j - 1 < |s|\}$ impl

`j := j - 1`

$\{j < |s|\}$ asg

$\{0 \leq j - i + 1 < E_0\}$ asg

$\{(0 \leq i \wedge j < |s|) \wedge 0 \leq j - i + 1 < E_0\}$ impl

`}`

$\{(0 \leq i \wedge j < |s|) \wedge \neg(i < j \wedge res = 1)\}$ while

$\{\top\}$ impl

(c) The while loop has the following invariant (note: this means that when $i = 0, j = |s| - 1$ the checked sequence will be empty - which is also a palindrome):

$$res = 1 \Leftrightarrow isPal(s[..i] + s[(j+1)..])$$

$\{0 \leq |s|\}$
`res := 1`
`i := 0`
`j := |s| - 1`
$\{isPal([])\}$ impl
$\{s[..0] = [] \land s[|s|..] = [] \land [] + [] = []\}$ impl
$\{isPal(s[..0] + s[|s|..])\}$ asg
$\{res = 1 \land isPal(s[..i] + s[(j+1)..])\}$ impl
$\{res = 1 \Leftrightarrow isPal(s[..i] + s[(j+1)..])\}$ impl
`while (i < j & res = 1) {`
    $\{i < j \land res = 1 \land isPal(s[..i] + s[(j+1)..])\}$ while
    $\{res = 1 \Leftrightarrow isPal(s[..i] + s[(j+1)..])\}$ impl
    `if (s[i] != s[j]) {`
        $\{s[i] \neq s[j]\}$ cond
        $\{\neg isPal(s[..(i+1)] + s[(j-1+1)..])\}$ impl
        `res := 0`
        $\{res \neq 1 \land \neg isPal(s[..(i+1)] + s[(j-1+1)..])\}$ impl
        $\{res = 1 \Leftrightarrow isPal(s[..(i+1)] + s[(j-1+1)..])\}$ impl
    `} else {`
        $\{\neg(s[i] \neq s[j])\}$ cond
        $\{isPal(s[..(i+1)] + s[(j-1+1)..])\}$ impl
        `skip`
        $\{res = 1 \land isPal(s[..(i+1)] + s[(j-1+1)..])\}$ impl
        $\{res = 1 \Leftrightarrow isPal(s[..(i+1)] + s[(j-1+1)..])\}$ impl
    `}`
    $\{res = 1 \Leftrightarrow isPal(s[..(i+1)] + s[(j-1+1)..])\}$ impl
    `i := i + 1`
    `j := j - 1`
    $\{res = 1 \Leftrightarrow isPal(s[..i] + s[(j+1)..])\}$ asg
`}`
$\{(res = 1 \Leftrightarrow isPal(s[..i] + s[(j+1)..])) \land \neg(i < j \land res = 1)\}$ while
$\{res = 1 \Leftrightarrow isPal(s)\}$ impl

Time

6 hours.