

Problem 5(a)

Proof of correctness for MAXODD(L):

For a natural $n \geq 0$, we define predicate $Q(n)$ as follows:

$Q(n)$: If L is a list of integers, $n = \text{len}(L)$, then MAXODD(L) terminates and returns the largest integer in $O(L)$ if $O(L)$ is non-empty. Otherwise, return 0.

By complete induction, we will prove $Q(n)$ holds for all $n \geq 0$ as follows:

BASIS:

Let $n = 0$

Then $L = \emptyset$ which means $O(L) = \emptyset$

$\implies \text{MAXODD}(L) = 0$ by Postcondition.

Line 1 is run since $\text{len}(L) \leq 1$, and line 2 is run which terminates and returns 0 as wanted.

Let $n = 1$

then L is a 1-element list which has no odd indices. By the definition of $O(L)$, it is empty.

$\implies \text{MAXODD}(L) = 0$ by Postcondition.

Line 1 is run since $\text{len}(L) \leq 1$, and line 2 is run which terminates and returns 0 as wanted.

INDUCTION STEP: let $n > 1$

Let $\max(L)$ return 0 if a list of integers L is empty.

Suppose $Q(j)$ holds whenever $0 \leq j < n$. [IH]

WTP: $Q(n)$ holds.

Since $\text{len}(L) = n > 1$, lines 4-8 is run.

There are 3 cases to consider:

Case 1: $L[1]$ is even

$O(L[2:]) = \emptyset$ by definition of $O(L)$

$\text{result} = m$ by line 5.

By line 8, result is returned and program terminates

$\implies \text{MAXODD}(L) = m$

WTS: $\text{MAXODD}(L) = \max(O(L))$

$\text{MAXODD}(L) = m$

$= \text{MAXODD}(L[2:])$

by line 4

$= \max(O(L[2:]))$

by [IH]

$= \max(\emptyset \cup O(L[2:]))$

by union properties

$= \max(O(L[2:]) \cup O(L[2:]))$

since $O(L[2:]) = \emptyset$

$= \max(O(L))$

by union properties

$\implies Q(n)$ holds as wanted.

Case 2: $L[1]$ is odd, $m = 0$

$O(L[2:]) = \emptyset$ by Postcondition of $\text{MAXODD}(L[2:])$ from line 4 which terminates by [IH].

$O(L[2:])$ is a 1-element list containing $L[1]$.

$\text{result} = L[1]$ by line 6.

By line 8, result is returned and program terminates

$\implies \text{MAXODD}(L) = L[1]$

WTS: $\text{MAXODD}(L) = \max(O(L))$

$\text{MAXODD}(L) = L[1]$

$= \max(L[1])$

by definition of $\max(L)$

$= \max(O(L[2:]))$

since $O(L[2:]) = L[1]$

$= \max(O(L[2:]) \cup \emptyset)$

by union properties

$= \max(O(L[2:]) \cup O(L[2:]))$

since $O(L[2:]) = \emptyset$

$= \max(O(L))$

by union properties

$\implies Q(n)$ holds as wanted.

Case 3: $L[1]$ is odd, $m \neq 0$

$O(L[2 :])$ is non-empty.

$O(L[: 2])$ is a 1-element list containing $L[1]$

$result = \max(L[1], m)$ by line 7

By line 8, $result$ is returned and program terminates

$\implies \text{MAXODD}(L) = \max(L[1], m)$

WTS: $\text{MAXODD}(L) = \max(O(L))$

$$\begin{aligned}
 \text{MAXODD}(L) &= \max(L[1], m) \\
 &= \max(L[1], \text{MAXODD}(L[2 :])) && \text{by line 4} \\
 &= \max(L[1], \max(O(L[2 :]))) && \text{by [IH]} \\
 &= \max(O(L[: 2]), \max(O(L[2 :]))) && \text{since } O(L[: 2]) = L[1] \\
 &= \max(O(L[: 2]) \cup O(L[2 :])) && \text{by definition of } \max(L) \\
 &= \max(O(L)) && \text{by union properties}
 \end{aligned}$$

$\implies Q(n)$ holds as wanted.

\therefore by Complete Induction, $Q(n)$ holds for all $n \in \mathbb{N}$ ■