Correctness Example: Horners Rule

Horner's Rule is an efficient algorithm for evaluating a polynomial at a point. The polynomial is represented by an ordered list of coefficients.

Algorithm:

1. $y \leftarrow 0$;
2. $i \leftarrow n$;
3. **while** $i \geq 0$ **do**
4. $\quad y \leftarrow a_i + x \cdot y$;
5. $\quad i \leftarrow i - 1$;

To show correctness, we want to show that

$$P(x) = y = \sum_{k=0}^{n} a_k x^k \tag{1}$$

Since the algorithm contains a loop, the first order of business is creation of a [useful] loop invariant.

Proposed Loop Invariant:

$$y = \sum_{k=0}^{n-(i+1)} a_{k+i+1} x^k$$

The idea is that the loop invariant captures the state of the computation at the start of each loop iteration.

Following the course text, we break the proof into three parts: initialization, maintenance, and termination.

Initialization: We show the loop invariant holds before the first execution of the loop.

Before the first execution of the loop, $i = n$, so the sum runs from $k = 0$ to $k = n - (n+1) = -1$. Hence the sum is 0. Since $y = 0$ before the first loop execution, the loop invariant is true.

Maintenance: Supposing the loop invariant is true at the start of execution of the j-th loop iteration, we want to show it is true at the start of the (j+1)-st loop iteration.

Let y_j and i_j be the values of y and i at the start of the j-th loop iteration and let y_{j+1} and i_{j+} be the values of y and i at the start of the (j+1)-st loop iteration.

At the start of execution of the j-th loop iteration,

$$y_j = \sum_{k=0}^{n-(i_j+1)} a_{k+i_j+1} x^k$$

We are assuming that the loop test is true and the loop body executes, so $i_j \geq 0$.

$$
\begin{aligned}
y_{j+1} &= a_{i_j} + x \cdot y_j \\
&= a_{i_j} + x \cdot \sum_{k=0}^{n-(i_j+1)} a_{k+i_j+1} \cdot x^k \\
&= a_{i_j} + \sum_{k=0}^{n-(i_j+1)} a_{k+i_j+1} \cdot x^{k+1} \\
&= a_{i_j} \cdot x^0 + a_{i_j+1} \cdot x^1 + a_{i_j+2} \cdot x^2 + \ldots + a_{i_j+n} \cdot x^{n-i_j} \qquad (2) \\
&= \sum_{k=0}^{n-i_j} a_{k+i_j} \cdot x^k
\end{aligned}
$$

For line (2), note that when $k = n-(i_j+1)$, $k+i_j+1 = n-(i_j+1)+(i_j+1) = n$ and $k+1 = n-(i_j+1)+1 = n - i_j$.

Finally, noting that $i_j = i_{j+1} + 1$, we have

$$y_{j+1} = \sum_{k=0}^{n-(i_{j+1}+1)} a_{k+i_{j+1}+1} x^k$$

which is the loop invariant at the start of the (j+1)-st loop iteration.

Termination: We show that the algorithm is correct (i.e., that equation (1) holds) at the end of the algorithm (using the loop invariant).

Since the loop invariant was true just before execution of the last loop test, and since no variables of the loop invariant changed during the loop test, the loop invariant is true. At termination, $i = -1$. The sum runs from $k = 0$ to $k = n - (-1 + 1) = n$; and the general term is $a_k x^k$. Hence the loop invariant reduces to

$$y = \sum_{k=0}^{n} a_k x^k$$

completing the proof.