CSCI 2600 - Homework 3

Problem 3: Polynomial Division

Pseudocode Algorithm for Polynomial Division

The algorithm implements polynomial long division: for polynomials u and v with $v \neq 0$, it computes q and r such that $u = q \cdot v + r$, where the degree of r is strictly less than the degree of v.

Algorithm 1 Polynomial Division

```
procedure Divide(u, v)
    if v = 0 or u.isNaN() or v.isNaN() then
        return NaN
    end if
    if u = 0 or degree(u) < degree(v) then
        return 0
    end if
    remainder ← u
    quotient ← 0
    while degree(remainder) ≥ degree(v) and remainder ≠ 0 do
        newTermExp ← degree(remainder) − degree(v)
        newTermCoeff ← remainder[degree(remainder)]/v[degree(v)]
        if newTermCoeff.isNaN() then
            return NaN
        end if
        term ← newTermCoeff · x^newTermExp
        quotient ← quotient + term
        remainder ← remainder − term · v
    end while
    return quotient
end procedure
```

Loop Invariant and Partial Correctness Proof

Loop Invariant: At the start of each iteration of the while loop (line 9), the following holds:

$$u = quotient \cdot v + remainder \tag{1}$$

Where:

- u is the dividend polynomial (input)
- v is the divisor polynomial (input)
- $quotient$ is the partial quotient polynomial computed so far
- $remainder$ is the current remainder polynomial

Proof of Partial Correctness:
Initialization: Before the first iteration of the loop:

$$quotient = 0 \tag{2}$$
$$remainder = u \tag{3}$$

Therefore:

$$\begin{aligned} quotient \cdot v + remainder &= 0 \cdot v + u && (4) \\ &= 0 + u && (5) \\ &= u && (6) \end{aligned}$$

So the invariant holds initially.

Maintenance: Assume the invariant holds at the beginning of an arbitrary iteration of the loop:

$$u = quotient \cdot v + remainder \tag{7}$$

During this iteration, we:

1. Compute $term = newTermCoeff \cdot x^{newTermExp}$ where:

$$newTermExp = degree(remainder) - degree(v) \tag{8}$$
$$newTermCoeff = \frac{remainder[degree(remainder)]}{v[degree(v)]} \tag{9}$$

2. Update $quotient' = quotient + term$
3. Update $remainder' = remainder - term \cdot v$

show that the invariant still holds after these updates:

$$
\begin{aligned}
\mathit{quotient}' \cdot v + \mathit{remainder}' &= (\mathit{quotient} + \mathit{term}) \cdot v + (\mathit{remainder} - \mathit{term} \cdot v) && (10)\\
&= \mathit{quotient} \cdot v + \mathit{term} \cdot v + \mathit{remainder} - \mathit{term} \cdot v && (11)\\
&= \mathit{quotient} \cdot v + \mathit{remainder} && (12)\\
&= u \quad \text{(by our assumption of the invariant)} && (13)
\end{aligned}
$$

Thus, the invariant is maintained through each iteration.

Termination: The loop terminates when either:

1. $\mathit{degree}(\mathit{remainder}) < \mathit{degree}(v)$, or
2. $\mathit{remainder} = 0$

In both cases, the invariant still holds:

$$u = \mathit{quotient} \cdot v + \mathit{remainder} \tag{14}$$

In the first case, we have a $\mathit{remainder}$ with degree less than $\mathit{degree}(v)$, which satisfies the condition for polynomial division.

In the second case, $\mathit{remainder} = 0$, which means $u = \mathit{quotient} \cdot v$, and v divides u exactly.

Therefore, when the algorithm terminates, we have computed $\mathit{quotient}$ such that $u = \mathit{quotient} \cdot v + \mathit{remainder}$ where either $\mathit{remainder} = 0$ or $\mathit{degree}(\mathit{remainder}) < \mathit{degree}(v)$.

This completes the proof of partial correctness.