PSOFT hw3 problem 3

Brian Wang-Chen

February 2024

- Write a pseudocode algorithm for polynomial division. Write your answer in the file answers/problem3.pdf.

```
FUNCTION div(p: RatPoly) -> RatPoly:
// Check for NaN or zero divisor
if this.isNaN() OR p.isNaN() THEN
    RETURN NaN
if p.equals(ZERO) THEN
    RETURN NaN
// Check if the divisor's degree is greater than this polynomial's degree
if p.degree > this.degree THEN
    RETURN ZERO
// Initialize a new array to store the coefficients of the quotient polynomial
quotient = new array of RatNum with length (this.degree - p.degree + 1)
// Initialize a copy of this polynomial's coefficients for the remainder polynomial
remainderCoeffs = copy(this.coeffs)
// Perform polynomial long division
for i from (quotient.length - 1) down to 0:
    if i + p.degree >= remainderCoeffs.length:
        quotient[i] = 0
    // Calculate the current coefficient of the quotient polynomial
    divided = remainderCoeffs[i + p.degree] / p.coeffs[p.degree]
    quotient[i] = divided
    // Update the remainder polynomial coefficients
    for j FROM p.degree down to 0:
        remainderCoeffs[i + j] = remainderCoeffs[i + j] - (divided * p.coeffs[j])
// Return the quotient polynomial
return new RatPoly(quotient)
```

- State the loop invariant for the main loop and prove partial correctness. Write your answer in the file answers/problem3.pdf. For the proof question, you d

loop Invariants:

quotient.length = this.degree - p.degree + 1

i > -1

u = v * quotient(q) + remainder(r)

Partial correctness proof:

Base Case: before loop

quotient.length = this.degree - p.degree + 1, ✓holds because degrees of the ratpoly object do not change

i > -1, ✓holds because i is set to be quotient.length - 1, so it will always be 0 or greater at base since quotient.length = this.degree - p.degree + 1

u = v * quotient + remainder, ✓holds because quotient is u / v = quotient + remainder, so u = v * quotient + remainder holds at base

---

Inductive step: k + 1

quotient.length = this.degree - p.degree + 1, ✓holds because degrees of the ratpoly object do not change

i - 1 > -1, ✓holds because i is set to be quotient.length - 1, p.degree is greater than zero, this.degree is greater than p, quotient has to be greater than zero. decrementing i sets it closer to zero so still holds because it will never be negative u = v * quotient + remainder, ✓holds because quotient is u / v = quotient + remainder, so u = v * quotient + remainder holds.

---

at loop termination: i == -1

loop terminates when i == -1 because it breaks the loop condition of i >= 0, and at the end, invariant still holds.

quotient.length = this.degree - p.degree + 1, ✓holds because degrees of the ratpoly object do not change

u = v * quotient + remainder, ✓holds because quotient is u / v = quotient + remainder, so u = v * quotient + remainder at at the end of loop, quotient, remainder should be calculated correctly.