COS 4892 Assignment 3

Adriaan Louw (53031377)

September 21, 2019

1 The Integer Remainder Computation Algorithm

The concept of a remainder is a very important concept in computing. It is used in fields like encryption and error checking. This essay will discuss how the pinciples developed in "Progam Construction" by Roland Backhouse was used to develop an algorithm for Remainder Computation in Chapter 15 of said book.

For integers, the remainder can be defined as what is left over after one number is divided by another. For instance, when we devide 10 by 3 we get a quotient of 3 and a remainder of 1. Or when we devide 4 by 2 we get a quotient of 2 and a remainder of 0.

Given an integer P divided by a positive natural number Q gives remainder r and quotient d. This can be expressed in the following way:

$$0 \leq r < Q \wedge P = Q.d + r \tag{1}$$

Where r will always be positive and less than Q by definition.

We can define the 2nd half of Formula 1 more precisely as:

$$0 \leq r < Q \wedge \langle \exists d :: P = Q.d + r \rangle \tag{2}$$

This uses the Quantifier Notation from Chapter 11. This notation has the general form

$$\langle qv : r : t \rangle \tag{3}$$

Where q is the quatifier that apllies to the dummy variable v. This could be the existential quatifier $\exists v$, the summation quantifier $\sum v$ or even the universal quantifier $\forall v$. r defines the range over which the quatifier operates. This is a boolean expression that determines the set of values the dummy variable can take. Then t is the term to which the dummy variable applies. In Formula 2 this means that there exists a value for d such that the expression $P = Q.d + r$ is valid. Here the range is not specified. Therefor d could be any integer.

The question remains whether in Formula 1 the values of r and d are unique for some value of P and Q. To prove this we need the concept of substitution of equals for equals or Liebniz's rule (Formula 4) for short which was discussed in Chapter 7.

$$(e = f) \equiv (e = f) \wedge (E[x := e] = E[x := f]) \tag{4}$$

This formula states that the "..application of a function to equal values gives equal results..."(From Chapter 7).

Now to prove the uniqueness of r and d we assume that 2 pairs (r,d) and (r',d') satisfy Formula 1 then we have

$$(0 \leq r < Q \wedge P = Q.d + r) \wedge (0 \leq r' < Q \wedge P = Q.d' + r') \tag{5}$$

which is equivalent to

$$(0 \leq r < Q \wedge P = Q.d + r) \wedge (Q > r' \geq 0 \wedge P = Q.d' + r') \tag{6}$$

Then to introduce r-r' and d-d' we see that by 'subtraction' the 2 terms, the previous statement will imply

$$-Q < r - r' < Q \wedge 0 = Q.(d - d') + (r - r') \tag{7}$$

Now using Liebniz and some some arithmatic, the above statement is equivalent to

$$-Q < -(Q.(d - d')) < Q \wedge -(Q.(d - d')) = r - r' \tag{8}$$

which becomes for Q ¿ 0

(9)

which with some arithmatic becomes

$$d = d' \wedge r = r' \tag{10}$$

Now that we know that r and d is unique, we can start to develop an algoritm to compute the remainder. Backhouse's book suggests splitting Formula 1 in to 2 giving:

$$P = Q.d + r \tag{11}$$

and

$$0 \leq r < Q \tag{12}$$

where Formula 11 can be established via the following initialization

$$r, d := P, 0 \tag{13}$$

This implies that this algorithm should consist of a loop with Formula 11 as the loop invariant and Formula 12 as the termination condition.

We have seen in chapter 3 of Bader - The spine of Software that there are 5 steps in proving the correctness of a loop.

1. Determine the loop invariant.
2. Prove that the initialization establishes the truth of the loop invariant.
3. Prove that the body of the loop preserves the truth of the loop invariant
4. Prove that the truth of the loop invariant and the termination condition together implies the correctness of the final result.
5. Prove that the termination condition will terminate in a finite number of executions of the loop.

For the case where $p \geq 0$, Backhouse suggests the following algorithm.

$$\begin{aligned} &\{0 < Q\} \\ &r, d := P, 0; \\ &\{Invariant : 0 \leq r \wedge P = Q.d + rBoundfunctionr\} \\ &do \\ &r \geq Q \rightarrow r, d := r - m, n \\ &od \\ &\{0 \leq r < Q \wedge P = Q.d + r\} \end{aligned} \tag{14}$$

This loop body is executed while $r \geq Q$ which means m and n need to satisfy the following conditions:

$$0 < m \tag{15}$$

and

$$(0 \leq r \wedge P = Q.d + r)[r, d := r - m, n] \tag{16}$$

Now we need to calculate the values for m and n that satisfy the invariant

$$(0 \leq r \wedge P = Q.d + r)[r, d := r - m, n] \tag{17}$$

which when doing the substitution becomes

$$0 \leq r - m \wedge P = Q.n + (r - m) \tag{18}$$

Now we assuming $r \geq Q$ which suggests that $m = Q$. Then the above equation implies

$$m = Q \wedge r \wedge P = Q.n + (r - Q) \tag{19}$$

Now by choosing n=d+1 the above statement implies

$$n = d + 1 \wedge m = Q \wedge r \geq Q \wedge O = Q(d + 1) + (r - Q) \tag{20}$$

which can be simpliefied into

$$n = d + 1 \wedge m = Q \wedge r \geq Q \wedge O = Q.d + r \tag{21}$$