CSI 2101 Discrete Structures | Winter 2012
Prof. Lucia Moura | University of Ottawa

Homework Assignment #4 (100 points, weight 5%)
Due: Thursday, April 5, at 1:00pm (in lecture)

Program verification, Recurrence Relations

1. Consider the following program that computes quotients and remainders:

   $r \leftarrow a$;
   $q \leftarrow 0$;
   while $r \geq d$ do
       begin
           $r \leftarrow r - d$;
           $q \leftarrow q + 1$;
   end

   Use the following steps in order to verify that the program is correct with respect to the initial assertion "a and d are positive integers" and final assertion "q and r are integers such that $a = dq + r$ and $0 \leq r < d$".

   (a) Find an appropriate loop invariant that is strong enough to give the final assertion, and prove that it is a loop invariant.

   (b) Using part (a) and other inference rules for program verification, prove the program is partially correct with respect to the initial and final assertions.

   (c) Complete a proof of correctness by formally proving the termination of the loop.

   (a) We claim that the loop invariant we need is the following proposition p:

   $$p = \text{“}a = qd + r \text{ and } r \geq 0\text{”}.$$

   To show that p is a loop invariant, we must show that:

   i. **p is true before the loop executes.** Since a is a positive integer and $r \leftarrow a$ before the loop executes, we have that $r \geq 0$. Since $q \leftarrow 0$ before the loop executes, then $qd + r = 0d + a = a$. Thus, p is true before the loop executes.

   ii. **If p is true before the loop is executed, then p is true after the loop executes.** Assume that p is true before the loop is executed. Then, after the loop executes, we have the new values $r_n = r - d$ and $q_n = q + 1$. We must show that p still holds with regards to these new values. Since, by the condition of the loop, $r \geq d$, we have that $r_n = r - d \geq d - d = 0$. Furthermore:

$$a = qd + r = qd + r - d + d = (qd + d) + (r - d) = (q + 1)d + (r - d) = q_n d + r_n.$$

Thus, p is still true after the loop executes.

Therefore, p is a loop invariant.

(b) Let S denote the entire program, S_1 denote the two statements before the while loop, and S_2 denote the statements in the while block. If q is the predicate "a and d are positive integers", and t is the predicate "q and r are positive integers such that $a = dq + r$ and $0 \leq r < d$", we show that $q\{S\}t$ holds. This is equivalent to showing $q\{S_1 \text{ while } r \geq d\{S_2\}\}t$ holds.

We must then show that $q\{S_1\}p$ and $(p \wedge r \geq d)\{S_2\}p$ holds: this is true from the first part, where we showed that p is a loop invariant. Thus, by the rules of inference for while loops, we have that $p\{\text{while } r \geq d\{S_2\}\}(p \wedge \neg(r \geq d))$. This implies that if the loop terminates, it does so with p true and $r \geq d$ false, i.e. $r < d$, and thus $a = qd + r$ and $0 \leq r < d$, which is precisely t. Thus, this is equivalent to $p\{\text{while } r \geq d\{S_2\}\}t$ holds. Since $q\{S_1\}p$ holds, we can combine these and have that $q\{S_1 \text{ while } r \geq d\{S_2\}\}t$, or $q\{S\}t$, as required.

(c) We show that the loop terminates eventually. Associate with each iteration of the loop the value of r. Since r is, by assumption, a positive integer, and in every iteration we decrease the value of r by d, the value of r forms a strictly decreasing sequence. Furthermore, since the loop terminates when $r < d$, we have that the value of r is bounded below by 0. Thus, by the well-ordering principle, the loop must terminate in a finite number of iterations.

2. (a) Find the characteristic roots of the linear homogeneous recurrence relation $a_n = 2a_{n-1} - 2a_{n-2}$. (Note these are complex numbers)

   (b) Find the solution of the recurrence relation in part (a) with $a_0 = 1$ and $a_1 = 2$.

The relation has characteristic equation:

$$r^2 - 2r + 2 = 0.$$

By using the quadratic equation, we have that:

$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{2 \pm \sqrt{4 - 8}}{2} = 1 \pm i.$$

Thus, the characteristic roots are $1 + i$ and $1 - i$.

This gives that the solution to the relation has form:

$$a_n = \alpha(1 + i)^n + \beta(1 - i)^n$$

for some numbers α, β. We use the initial values to determine α and β:

$$
\begin{aligned}
a_0 &= 1 = \alpha + \beta \\
a_1 &= 2 = \alpha(1+i) + \beta(1-i)
\end{aligned}
$$

By substituting $\beta = 1 - \alpha$ into the second equation, we derive:

$$
\begin{aligned}
\alpha(1+i) + (1-\alpha)(1-i) &= 2 \\
2i\alpha &= 1+i \\
\alpha &= \frac{1+i}{2i} = \frac{1+i}{2i} \times \frac{i}{i} = \frac{1-i}{2}.
\end{aligned}
$$

Thus:

$$\beta = 1 - \alpha = 1 - \frac{1-i}{2} = \frac{1+i}{2}.$$

Hence, the solution to the recurrence relation is:

$$a_n = \left(\frac{1-i}{2}\right)(1+i)^n + \left(\frac{1+i}{2}\right)(1-i)^n.$$

3. Find all solutions of the recurrence relation $a_n = 7a_{n-1} - 16a_{n-2} + 12a_{n-3} + n4^n$ with $a_0 = -2$, $a_1 = 0$ and $a_2 = 5$.

   This is a nonhomogeneous recurrence relation, so we need to find the solution to the associated homogeneous recurrence relation and a particular solution to the original relation.

   The associated homogeneous recurrence relation is:

   $$a_n^{(h)} = 7a_{n-1}^{(h)} - 16a_{n-2}^{(h)} + 12a_{n-3}^{(h)}.$$

   This has characteristic equation:

   $$
   \begin{aligned}
   r^3 - 7r^2 + 16r - 12 &= 0 \\
   (r-2)^2(r-3) &= 0
   \end{aligned}
   $$

   Thus, the solution to the homogeneous relation is:

   $$a_n^{(h)} = \alpha 2^n + \beta n 2^n + \gamma 3^n$$

   for some real numbers α, β, γ, which we will find later via the initial values after we have the general solution to the full recurrence.

   We now need the particular solution. We have that:

   $$F(n) = n4^n$$

This has polynomial part n, so the degree of the polynomial part is $t = 1$. It has exponential part 4^n, so $s = 4$. By S7.2 Theorem 6, the particular solution thus has form:

$$a_n^{(p)} = (qn + p)4^n$$

for some real numbers p, q. We find the values of p and q by substituting the particular solution $a^{(p)}$ into the original recurrence relation:

$$\begin{aligned} a_n^{(p)} &= 7a_{n-1}^{(p)} - 16a_{n-2}^{(p)} + 12a_{n-3}^{(p)} + n4^n \\ (qn + p)4^n &= 7(q(n-1) + p)4^{n-1} - 16(q(n-2) + p)4^{n-2} + 12(q(n-3) + p)4^{n-3} + n4^n \end{aligned}$$

We now divide the equation by 4^{n-3} to get:

$$(qn + p)4^3 = 7(q(n-1) + p)4^2 - 16(q(n-2) + p)4^1 + 12(q(n-3) + p) + n4^3.$$

Multiplying out and simplifying gives:

$$(4q - 64)n + (4p - 20q) = 0 = 0n + 0.$$

This can be separated into two equations by setting the coefficients of the polynomials to be equal:

$$\begin{aligned} 4q - 64 &= 0 \\ 4p - 20q &= 0 \end{aligned}$$

This has solution $p = -80$, $q = 16$, so the particular solution is:

$$a_n^{(p)} = (16n - 80)4^n.$$

Thus, the format of the general solution to the recurrence relation is:

$$a_n = a_n^{(h)} + a_n^{(p)} = \alpha 2^n + \beta n 2^n + \gamma 3^n + (16n - 80)4^n.$$

Using the initial values, we have:

$$\begin{aligned} a_0 &= -2 = \alpha + \gamma - 80 \\ a_1 &= 0 = 2\alpha + 2\beta + 3\gamma + (-64) \cdot 4 \\ a_2 &= 5 = 4\alpha + 8\beta + 9\gamma + (-48) \cdot 16 \end{aligned}$$

This gives a system of three linear equations in three unknowns, which has solution $\alpha = 17$, $\beta = \frac{39}{2}$, $\gamma = 61$. Hence, the recurrence relation has solution:

$$\begin{aligned} a_n &= 17 \cdot 2^n + \frac{39}{2} n 2^n + 61 \cdot 3^n + (16n - 80)4^n \\ &= 17 \cdot 2^n + 39n2^{n-1} + 61 \cdot 3^n + (16n - 80)4^n. \end{aligned}$$

4. Consider the following recursive procedure to compute the fibonacci numbers:

procedure FIB(n: non-negative integer)
  if $n = 0$ then return 0
  else if $n = 1$ then return 1
  else return FIB($n-1$)+FIB($n-2$)

(a) Set up a recurrence relation that counts the number of times the sum (+) is executed considering all the recursive calls used for input n. (Don't forget to provide initial conditions as well)

(b) Solve the recurrence relation of part (a).

Let a_n be the number of sum operations that are performed in calculating the nth fibonacci number using the recursive procedure. If $n = 0$ or $n = 1$, no sum operations are performed, which gives the initial conditions $a_0 = a_1 = 0$. For $n > 1$, we have that the recursive procedure calculates the $(n-1)$th and $(n-2)$th number and adds them together. Calculating the $(n-1)$th number requires a_{n-1} sum operations, and calculating the $(n-2)$th number requires a_{n-2} of them. We then have one more sum operation to add the two numbers together, giving that:

$$a_n = a_{n-1} + a_{n-2} + 1.$$

This is a nonhomogeneous recurrence relation. The associated homogeneous recurrence relation is:

$$a_n^{(h)} = a_{n-1}^{(h)} + a_{n-2}^{(h)}$$

which has characteristic equation:

$$r^2 - r - 1 = 0.$$

This equation has roots:

$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{1 \pm \sqrt{5}}{2}.$$

Thus, the homogeneous relation has solution:

$$a_n^{(h)} = \alpha \left(\frac{1+\sqrt{5}}{2} \right)^n + \beta \left(\frac{1-\sqrt{5}}{2} \right)^n$$

for values α, β that we will later derive from the initial values.

We now need to find a particular solution to the original recurrence. Since $F(n) = 1$, we have that the polynomial part is 1, so $t = 0$, and the exponential part is $1 = 1^n$, so $s = 1$. Thus, the particular solution has form:

$$a_n^{(p)} = (p)1^n = p$$

for some value p. To find p, we substitute the particular solution into the original relation:

$$\begin{aligned} a_n &= a_{n-1} + a_{n-2} + 1 \\ p &= p + p + 1 \\ p &= -1 \end{aligned}$$

Thus, the general solution is:

$$a_n = a^{(h)} + a^{(p)} = \alpha \left(\frac{1+\sqrt{5}}{2} \right)^n + \beta \left(\frac{1-\sqrt{5}}{2} \right)^n - 1$$

Using the initial values, we have that:

$$\begin{aligned} a_0 &= 0 = \alpha + \beta - 1 \\ a_1 &= 0 = \alpha \left(\frac{1+\sqrt{5}}{2} \right) + \beta \left(\frac{1-\sqrt{5}}{2} \right) - 1 \end{aligned}$$

The solution to this system of equations is:

$$\alpha = \frac{5+\sqrt{5}}{10}, \qquad \beta = \frac{5-\sqrt{5}}{10}.$$

Thus, the solution to the recurrence relation is:

$$a_n = \frac{5+\sqrt{5}}{10} \times \left(\frac{1+\sqrt{5}}{2} \right)^n + \frac{5-\sqrt{5}}{10} \times \left(\frac{1-\sqrt{5}}{2} \right)^n - 1.$$

5. Consider the method by Karatsuba for multiplication of large integers given below:

procedure KMULT(A, B, n: A and B are integers with n bits)
1. If $n = 1$ then return $A \cdot B$;
2. else Write $A = A_h 2^{n/2} + A_l$ and $B = B_h 2^{n/2} + B_l$
3. Compute $A' = A_h + A_l$ and $B' = B_h + B_l$
4. $C =$ KMULT$(A', B', n/2)$
5. $D_h =$ KMULT$(A_h, B_h, n/2)$
6. $D_l =$ KMULT$(A_l, B_l, n/2)$
7. return $X = D_h \cdot 2^n + [C - D_h - D_l] \cdot 2^{n/2} + D_l$

(a) Based on the program we can see that the number of basic operations for line 1 is 1 and the total number of basic operations for lines 2, 3 and 7 is at most $C \cdot n$ for some constant C (since the operations are on numbers of at most n bits). Write a recurrence relation for $T(n)$, the number of basic operations used in all recursive calls for the cases in which n is a power of 2 (i.e. $n = 2^k$ for some k).

(b) Use the master theorem (page 479) to find a big-Oh estimate for $T(n)$.

We have that there are three recursive calls to KMULT with sequences of about half the number of the original number of bits, thus giving that the recurrence relation is:

$$T(n) = 3T\left(\frac{n}{2}\right) + C \cdot n.$$

Additionally, $T(1) = 1$ since when $n = 1$, we perform one operation (line 1). This, however, is not necessary to apply the master theorem. Using the master theorem, we have that $a = 3$, $b = 2$, and $d = 1$. Thus, $b^d = 2^1 = 2$, and we have that $a > b^d$. Hence, we are in the third case of the master theorem, which says that $T(n)$ is $O(n^{\log_b a}) = O(n^{\log_2 3}) = O(n^{1.585})$.