Problem Sheet 12

Henri Sota
h.sota@jacobs-university.de
Computer Science 2022

December 6, 2019

Problem 12.1

```
1  K := n
2  P := x
3  Y := 1
4  while K > 0 do
5      if K mod 2 = 0 then
6          P := P x P
7          K := K/2
8      else
9          Y := Y x P
10         K := K - 1
11     fi
12 od
```

a) To find a proper precondition, we look into the algorithm and we see that in order for it work:

- $K > 0$ in order for our loop to execute
- $K = n$ (assignment)
- $P = x$ (assignment)
- $Y = 1$ (assignment)

where n and x are auxiliary variables we will use to connect the precondition P with postcondition Q. Our precondition is:

$$P = \{(n > 0), (n = n), (x = x), (1 = 1)\}$$

This is because K, P, Y are set by assignment later on, which form the annotation after the sequence of assignment commands and before the WHILE loop:

$$\{(K > 0), (K = n), (P = x), (Y = 1)\}$$

To find a proper postcondition, we look into the algorithm and we see that given different input for n and x, the value of Y becomes x^n in the end and K equals to 0.

$$Q = \{Y = x^n\}$$

This can be verified by inputting different values for n and x. One example is given on the table below:

Iteration	**n**	**x**	**Y**	**K**	**P**	**K>0**
0	5	3	1	5	3	true
1	5	3	3	4	3	true
2	5	3	3	2	9	true
3	5	3	3	1	81	true
4	5	3	243	0	81	false

Table 1: State of the algorithm during each iteration given $n = 5$ and $x = 3$

b) Annotations are required:

i) before each command C_i (with $i > 1$) in a sequence $C_1; C_2; \ldots; C_n$, where C_i is not an assignment command

ii) after the keyword DO in a WHILE command (loop invariant)

Since there is a while loop inside of our algorithm, we need to find an invariant P such that:

- $\{P \wedge K > 0\} C_1; C_2; \ldots; C_n \{P\}$ (Invariant is maintained after execution of commands)
- $Y = 1 \wedge P = x \wedge K = n \rightarrow P$ (Invariant is initially true)
- $P \wedge \neg\{K > 0\} \rightarrow Y = x^n$ (Invariant and exit condition imply postcondition)

From the trace and the post-condition, also logically verified from table 1, a candidate loop invariant is $Y * p^K = x^n$, which holds after each iteration. We can strengthen it by adding another term which is necessary for the loop to be executed correctly: $\{(K \geq 0), (Y * P^K = x^n)\}$. Substituting and using laws of logic, we can prove:

- $\{\{(K \geq 0), (Y * P^K = x^n)\} \wedge K > 0\}$
  IF K mod 2 = 0 THEN P := P * P; K := K/2 ELSE Y := Y * P; K := K - 1 FI
  $\{(K \geq 0), (Y * P^K = x^n)\}$
- $\{Y = 1 \wedge P = x \wedge K = n\} \rightarrow \{(K \geq 0), (Y * P^K = x^n)\}$ — since $P = x$,$K = n$ and $Y = 1$, then $1 * x^n = x^n$
- $\{\{(K \geq 0), (Y * P^K = x^n)\} \wedge \neg(K > 0)\} \rightarrow \{Y = x^n\}$ — since on exit $K \geq 0 \wedge K \leq 0$, which means $K = 0$, then $Y * P^0 = \{Y = x^n\}$

Since the command sequence inside of the WHILE loop is a conditional, we need annotation for the command sequences inside of the branches of the conditional:

- When K mod $2 = 0$ is TRUE:
  $\{(K \geq 0), (Y * P^K = x^n)\} \wedge \{K > 0\} \wedge \{K \text{ mod } 2 = 0\} \rightarrow \{(\frac{K}{2} \geq 0), (Y * (P * P)^{\frac{K}{2}} = x^n)\}$
- When K mod $2 = 0$ is FALSE:
  $\{(K \geq 0), (Y * P^K = x^n)\} \wedge \{K > 0\} \wedge \{K \text{ mod } 2 = 1\} \rightarrow \{(K - 1 \geq 0), ((Y * P) * P^{K-1} = x^n)\}$

Annotation for partial correctness:

Precondition: $\{(n > 0) \wedge (n = n) \wedge (x = x) \wedge (1 = 1)\}$

```
1     K := n
2     P := x
3     Y := 1
4     {(K > 0) ∧ (K = n) ∧ (P = x) ∧ (Y = 1)}
5     while K > 0 do
6         {(K ≥ 0) ∧ (Y * P^K = x^n)}
7         if K mod 2 = 0 then
8             P := P x P
9             K := K/2
10            {(K/2 ≥ 0) ∧ (Y * (P * P)^(K/2) = x^n)}
11        else
12            Y := Y x P
13            K := K - 1
14            {(K − 1 ≥ 0) ∧ ((Y * P) * P^(K−1) = x^n)}
15        fi
16    od
```

Postcondition: $\{Y = x^n\}$

c) According to the sequence rules, verification rules for the sequence consisting of the initial assignments, the while loop and the conditional and its assignments.

1. Initial assignments:
   $((n > 0) \wedge (n = n) \wedge (x = x) \wedge (1 = 1)) \rightarrow ((K > 0) \wedge (K = n) \wedge (P = x) \wedge (Y = 1))$
2. While loop entry:
   $((Y = 1) \wedge (P = x) \wedge (K = n)) \rightarrow ((K \geq 0) \wedge (Y * P^K = x^n))$
3. While loop exit:
   $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge \neg(K > 0)) \rightarrow (Y = x^n)$
4. While loop and true conditional:
   $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K \bmod 2 = 0)) \rightarrow ((\frac{K}{2} \geq 0) \wedge (Y * (P * P)^{\frac{K}{2}} = x^n))$
5. While loop and false conditional:
   $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge \neg(K \bmod 2 = 0))$
   $\rightarrow ((K - 1 \geq 0) \wedge ((Y * P) * P^{K-1} = x^n))$

d) *Proof:*
To prove partial correctness, each VC should be proved:

1. Initial assignments:
   $((n > 0) \wedge (n = n) \wedge (x = x) \wedge (1 = 1)) \rightarrow ((K > 0) \wedge (K = n) \wedge (P = x) \wedge (Y = 1))$
   $\{(n > 0) \wedge (n = n) \wedge (x = x) \wedge (1 = 1)\}$
   $K := n; P := x; Y := 1$
   $\{(K > 0) \wedge (K = n) \wedge (P = x) \wedge (Y = 1)\}$
   $\{(n > 0) \wedge (n = n) \wedge (x = x) \wedge (1 = 1)\}$
   $K := n; P := x;$
   $\{(K > 0) \wedge (K = n) \wedge (P = x) \wedge (1 = 1)\}$
   $\{(n > 0) \wedge (n = n) \wedge (x = x) \wedge (1 = 1)\}$
   $K := n;$
   $\{(K > 0) \wedge (K = n) \wedge (x = x) \wedge (1 = 1)\}$
   $((n > 0) \wedge (n = n) \wedge (x = x) \wedge (1 = 1)) \rightarrow ((n > 0) \wedge (n = n) \wedge (x = x) \wedge (1 = 1))$

2. While loop entry:
   $((Y = 1) \wedge (P = x) \wedge (K = n)) \rightarrow ((K \geq 0) \wedge (Y * P^K = x^n))$
   Already proved while finding the invariant up above. Writing the same result here:
   In the formula $Y * P^K = x^n$, substituting Y for 1, P for x and K for n on while loop entry, implies $1 * x^n = x^n$
3. While loop exit:
   $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge \neg(K > 0)) \rightarrow (Y = x^n)$
   Already proved while finding the invariant up above. Writing the same result here:
   Since we have two conditions, $K \geq 0$ and $\neg(K > 0)$ while exiting the loop, this implies that $K = 0$. Formula $Y * P^K = x^n$ becomes $Y * P^0 = Y * 1$ which implies $Y = x^n$.
4. While loop and true conditional:
   $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K \bmod 2 = 0)) \rightarrow ((\frac{K}{2} \geq 0) \wedge (Y * (P * P)^{\frac{K}{2}} = x^n))$
   $\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K \bmod 2 = 0)\}$
   $P := P * P; K := K/2$
   $\{(\frac{K}{2} \geq 0) \wedge (Y * (P * P)^{\frac{K}{2}} = x^n)\}$
   $\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K \bmod 2 = 0)\}$
   $P := P * P$
   $\{(K \geq 0) \wedge (Y * (P * P)^K = x^n)\}$
   $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K \bmod 2 = 0)) \rightarrow ((K \geq 0) \wedge (Y * P^K = x^n))$
5. While loop and false conditional:
   $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge \neg(K \bmod 2 = 0))$
   $\rightarrow ((K - 1 \geq 0) \wedge ((Y * P) * P^{K-1} = x^n))$
   $\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge \neg(K \bmod 2 = 0)\}$
   $Y := Y * P; K := K - 1$
   $\{(K - 1 \geq 0) \wedge ((Y * P) * P^{K-1} = x^n)\}$
   $\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge \neg(K \bmod 2 = 0)\}$
   $Y := Y * P$
   $\{(K \geq 0) \wedge ((Y * P) * P^K = x^n)\}$
   $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge \neg(K \bmod 2 = 0)) \rightarrow ((K \geq 0) \wedge (Y * P^K = x^n))$

■

(NOTE) Expanding each of these using sequence rule and assignment axiom leads to a lengthier proof which proves partial correctness the same way. By combining the steps, proof is more compact.

e) Only WHILE can cause loops that potentially do not terminate. Other command rules can simply be extended to cover total correctness. Our program contains one and therefore we need to add annotation for the while rule for total correctness. We will use E, an integer-valued expression and an auxiliary variable n, which doesn't occur in the precondition, while loop's command, condition or E itself. We have to show that this non-negative integer, the variant, decreases on each iteration of the loop command.

$$P \text{ WHILE } S \text{ DO } R \text{ [E] C OD } Q$$
$$P \rightarrow R$$
$$R \wedge \neg S \rightarrow Q$$
$$R \wedge S \rightarrow E \geq 0$$
$$\{R \wedge S \wedge (E = n)\} C \{R \wedge (E < n)\}$$

Use [K] for [E] as the only annotation needed for total correctness of a while loop.

Annotation for total correctness:

Precondition: $\{(n > 0) \wedge (n = n) \wedge (x = x) \wedge (1 = 1)\}$

```
1     K := n
2     P := x
3     Y := 1~
4     {(K > 0) ∧ (K = n) ∧ (P = x) ∧ (Y = 1)}
5     while K > 0 do
6         {(K ≥ 0) ∧ (Y * P^K = x^n)}
7         [K]
8         if K mod 2 = 0 then
9             P := P x P
10            K := K/2
11            {(K/2 ≥ 0) ∧ (Y * (P * P)^(K/2) = x^n)}
12        else
13            Y := Y x P
14            K := K - 1
15            {(K − 1 ≥ 0) ∧ ((Y * P) * P^(K−1) = x^n)}
16        fi
17    od
```

Postcondition: $\{Y = x^n\}$

f) The while loop rule gives use the following termination verification conditions:

1. $((Y = 1) \wedge (P = x) \wedge (K = n)) \rightarrow ((K \geq 0) \wedge (Y * P^K = x^n))$
2. $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge \neg(K > 0)) \rightarrow (Y = x^n)$
3. $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0)) \rightarrow (K \geq 0)$
4. $\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K = n)\}$
   IF K mod 2 = 0 THEN P := P * P; K := K/2 ELSE Y := Y * P; K := K - 1 FI
   $\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K < n)\}$

Since there is a conditional in the command sequence inside of the while loop, last termination verification condition can be divided into:

- True Conditional:
  $\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K = n)\} \wedge \{K \bmod 2 = 0\}$
  $P := P * P; K := K/2$
  $\{((\frac{K}{2} \geq 0) \wedge (Y * (P * P)^{\frac{K}{2}} = x^n)) \wedge (\frac{K}{2} < n)\}$
- False Conditional:
  $\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K = n)\} \wedge \{K \bmod 2 = 1\}$
  $Y := Y * P; K := K - 1$
  $\{((K - 1 \geq 0) \wedge ((Y * P) * P^{K-1} = x^n)) \wedge (K - 1 < n)\}$

g) *Proof:*
To prove total correctness, each VC should be proved:

1. Already proved as verification condition of partial correctness
2. Already proved as verification condition of partial correctness
3. $(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0)) \rightarrow (K \geq 0)$ since $(K \geq 0) \wedge (K > 0)$ then $(K > 0)$
4. $\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K = n)\} \wedge \{K \bmod 2 = 0\}$
   $P := P * P; K := K/2$

$\{((\frac{K}{2} \geq 0) \wedge (Y * (P * P)^{\frac{K}{2}} = x^n)) \wedge (\frac{K}{2} < n)\}$
$\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K = n)\} \wedge \{K \bmod 2 = 0\}$
$P := P * P$
$\{((K \geq 0) \wedge (Y * (P * P)^K = x^n)) \wedge (K < n)\}$
$(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K = n)\} \wedge \{K \bmod 2 = 0) \rightarrow (((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K < n))$
Since during each execution of the true branch of the conditional, K will become half of its initial value when starting the iteration, K will be smaller than the value of the auxiliary variable it was assigned to, and, K will be positive as division by 2 still yields a positive value between initial K and 0.

5. $\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K = n)\} \wedge \{K \bmod 2 = 1\}$
$Y := Y * P; K := K - 1$
$\{((K - 1 \geq 0) \wedge ((Y * P) * P^{K-1} = x^n)) \wedge (K - 1 < n)\}$
$\{((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K = n)\} \wedge \{K \bmod 2 = 1\}$
$Y := Y * P;$
$\{((K \geq 0) \wedge ((Y * P) * P^K = x^n)) \wedge (K < n)\}$
$(((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K > 0) \wedge (K = n)\} \wedge \{K \bmod 2 = 1) \rightarrow (((K \geq 0) \wedge (Y * P^K = x^n)) \wedge (K < n))$
Since during each execution of the false branch of the conditional, K will decrement by 1 of its initial value when starting the iteration, K will be smaller than the value of the auxiliary variable it was assigned to, and, K will be positive as subtracting by 1 still yields a positive value between initial K and 0.

■