

# CSE4213 Lecture Notes

## Data Refinement

Wordsworth

Computer Science and Software Engineering  
Monash University

20050523 / Lecture 23

- 1 Correctness of Loops
- 2 An Example of Loop Correctness

# Correctness of Loops

The following discussion is taken from Wordsworth, section 7.5

# Loop Format I

The loop construct we shall examine is the **while** loop. Other loops can be cast into this form.

```
WHILE P DO G VARIANT E INVARIANT Q END
```

This is the loop substitution, where  $P$  and  $Q$  are predicates,  $G$  is a substitution, and  $E$  is an expression.

The **loop variant** is an expression that denotes a natural number. Informally, it should decrease on each iteration, and the fact that it eventually reaches zero is the guarantee of loop termination.

The **loop invariant** is a predicate that identifies partial fulfilment of the goal of the loop. It is the key to asserting that when the loop terminates, the desired goal has been reached.

# Loop Format II

We extend the substitution above to identify both the initialisation of the loop, and the desired goal of the loop, expressed as a predicate. Then the substitution can be written:

$$[ H ; \text{WHILE } P \text{ DO } G \text{ VARIANT } E \text{ INVARIANT } Q \text{ END } ] R$$

where  $H$  is the substitution that establishes the initialisation, and  $R$  is the predicate identifying the goal of the loop

# Rules for Correctness

There are five rules for loop correctness, defining constraints on **initialisation**, **establishing the goal**, **termination** (2 rules), and **invariant maintenance**.

I-rule:  $[H] \ Q$

F-rule:  $! \ P \ \& \ Q \ \Rightarrow \ R$

T1-rule:  $Q \ \Rightarrow \ E \ : \ \text{NAT}$

T2-rule:  $P \ \& \ Q \ \Rightarrow \ [y \ := \ E] \ [G] \ E < y$

P-rule:  $P \ \& \ Q \ \Rightarrow \ [G] \ Q$

# The Initialisation I-rule

The loop invariant  $Q$  must be true before the start of the loop.

This is the role of the initialisation substitution  $H$ , and hence we can write the I-rule as

$$[H] \quad Q$$

read informally as “The loop initialisation must establish the loop invariant”.

# The Finalisation F-rule

When the loop ends, the goal predicate  $R$  must be true. For the loop to terminate, we have

- The loop invariant  $Q$  is true (since it must be true on every iteration of the loop);
- The while test is false (since otherwise we would have iterated again).

Hence we can state the F-rule as

$$! P \ \& \ Q \Rightarrow R$$

Note that this is the only rule which mentions  $R$ .

# The Termination T1-rule

This rule is really to ensure that the following rule is correctly typed. But it also ensures that the loop must terminate, since once the variant reaches zero, it cannot go any lower and still be a natural number.

$$Q \Rightarrow E : \text{NAT}$$

# The Termination T2-rule

The body of the loop  $G$  must decrease the loop variant. We take a snapshot of the variant before the loop body, execute it, and then check that the variant is less than it was.

$$P \ \& \ Q \Rightarrow [y := E] \ [G] \ E < y$$

# The Preserve Invariant P-rule

If the invariant  $Q$  is to truly be “invariant”, it must be true on each iteration of the loop. Since  $Q$  is established at the start, we use an induction rule to establish that it is true after any single iteration. We iterate when  $P$  is true, hence

$$P \ \& \ Q \Rightarrow [G] \ Q$$

which asserts that the loop body  $G$  re-establishes  $Q$ .

# An Example of Loop Correctness

The following trivial example shows how to use the five rules to establish loop correctness.

We wish to increment a variable  $xx$  by 5 using a loop.

We record the original value in  $yy$ , and check it at the end of the loop:

$$[ yy := xx ; H ; loop ] xx = yy+5$$

# Expanding the Loop I

We then start to expand the loop. We introduce a counter `ctr` which we initialise to 1, and loop until it is greater than 5:

```
[ yy := xx ; ctr := 1 ;  
  WHILE ctr <= 5 DO G VARIANT E INVARIANT Q END ]  
xx = yy+5
```

The substitution `G` is fairly obvious, since we have to increment both `xx` and `ctr`:

```
[ yy := xx ; ctr := 1 ;  
  WHILE ctr <= 5 DO  
    xx := xx + 1 ; ctr := ctr + 1  
    VARIANT E INVARIANT Q  
  END ]  
xx = yy+5
```

# Expanding the Loop II

But what do we put for  $E$  and  $Q$ ?  $E$  has to decrease on each cycle, so something like  $6 - ctr$  will suffice.

The invariant must capture the fact that both  $xx$  and  $ctr$  count up together as they head towards the goal of  $xx = yy + 5$ , so we put  $xx = yy + ctr - 1$  (subtract 1 since  $ctr$  starts at 1).

Note that by the T1-rule,  $Q$  must also imply that the variant remains a natural number, so we shall limit  $ctr$  in the set of values it can take on.

```
[ yy := xx ; ctr := 1 ;
  WHILE ctr <= 5 DO
    xx := xx + 1 ; ctr := ctr + 1
    VARIANT 6-ctr INVARIANT ctr : 1..6 & xx = yy + ctr - 1
  END ]
xx = yy+5
```

# Apply the I-rule

The I-rule is  $[H] Q$ , so substituting gives:

$$[yy:=xx;ctr:=1] \text{ ctr}:1..6 \ \& \ xx=yy+ctr-1$$

We apply the sequence rule and make the substitutions:

$$1:1..6 \ \& \ xx=xx+1-1$$

which simplifies to a tautology.

# Apply the F-rule

The F-rule is  $! P \ \& \ Q \Rightarrow R$ , so substituting gives:

$$!ctr \leq 5 \ \& \ ctr:1..6 \ \& \ xx=yy+ctr-1 \Rightarrow xx=yy+5$$

The first two conjuncts imply that  $ctr=6$ , so substituting gives:

$$6:1..6 \ \& \ xx=yy+6-1 \Rightarrow xx=yy+5$$

Since the second conjunct of the antecedent implies the consequent, this is clearly true.

# Apply the T1-rule

The T1-rule is  $Q \Rightarrow E : \text{NAT}$ , so substituting gives:

$$\text{ctr}:1..6 \ \& \ \text{xx}=\text{yy}+\text{ctr}-1 \Rightarrow 6-\text{ctr} : \text{NAT}$$

The value of the expression in the consequent must be in the range  $0..5$ , and hence the consequent is true.

# Apply the T2-rule

The T2-rule is  $P \ \& \ Q \Rightarrow [y := E] \ [G] \ E < y$ , so substituting gives:

$$\text{ctr} \leq 5 \ \& \ \text{ctr}:1..6 \ \& \ \text{xx}=\text{yy}+\text{ctr}-1 \Rightarrow \\ [y := 6-\text{ctr}] \ [\text{xx} := \text{xx}+1 \ ; \ \text{ctr} := \text{ctr}+1] \ 6-\text{ctr} < y$$

Applying the substitutions give:

$$\text{ctr} \leq 5 \ \& \ \text{ctr}:1..6 \ \& \ \text{xx}=\text{yy}+\text{ctr}-1 \Rightarrow \\ 6-(\text{ctr}+1) < 6-\text{ctr}$$

Simplifying the consequent, and taking into account the second conjunct of the antecedent establishes the predicate.

# Apply the P-rule

The P-rule is  $P \ \& \ Q \Rightarrow [G] \ Q$ , so substituting gives:

$$\begin{aligned} & \text{ctr} \leq 5 \ \& \ \text{ctr}:1..6 \ \& \ \text{xx}=\text{yy}+\text{ctr}-1 \Rightarrow \\ & \quad [\text{xx} := \text{xx} + 1 \ ; \ \text{ctr} := \text{ctr} + 1] \\ & \quad \text{ctr}:1..6 \ \& \ \text{xx}=\text{yy}+\text{ctr}-1 \end{aligned}$$

Perform the substitutions to get:

$$\begin{aligned} & \text{ctr} \leq 5 \ \& \ \text{ctr}:1..6 \ \& \ \text{xx}=\text{yy}+\text{ctr}-1 \Rightarrow \\ & \quad (\text{ctr}+1):1..6 \ \& \ \text{xx}+1=\text{yy}+(\text{ctr}+1)-1 \end{aligned}$$

The first two conjuncts of the antecedent establish the first conjunct of the consequent, and adding 1 to both sides of the last conjunct of the antecedent gives the same conjunct as the second of the consequent conjuncts.

# Exercise

Rework this example, using different strategies for counting. Start `ctr` at zero, for example, or count it downwards from 5.