Weakest Precondition Calculus

COMP2600 — Formal Methods for Software Engineering

Rajeev Goré

Australian National University

Semester 2, 2016

(Most lecture slides due to Ranald Clouston)

Weakest Preconditions for Conditionals (Rule 3a/4)

$$wp(\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2, Q) \equiv (b \Rightarrow wp(S_1, Q)) \wedge (\neg b \Rightarrow wp(S_2, Q))$$

Proof:

By cases on condition b,

- b is *true*: $RHS \equiv (True \Rightarrow wp(S_1, Q)) \wedge (False \Rightarrow wp(S_2, Q))$
  wp for the conditional is the weakest precondition for S_1 guaranteeing postcondition Q – that is, LHS is $wp(S_1, Q)$.
  The right hand side reduces to the same thing if we replace b with $True$.

- b is *false*:
  Similarly, both left hand and right hand sides reduce to $wp(S_2, Q)$

Conditional Example:

$$wp(\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2, Q) \equiv (b \Rightarrow wp(S_1, Q)) \wedge (\neg b \Rightarrow wp(S_2, Q))$$

$$wp(\texttt{if x>2 then y:=1 else y:=-1}, (y > 0))$$

$$\equiv ((x > 2) \Rightarrow wp(\texttt{y:=1}, (y > 0)) \wedge (\neg(x > 2) \Rightarrow wp(\texttt{y:=-1}, (y > 0))$$

$$\equiv ((x > 2) \Rightarrow (1 > 0)) \wedge (\neg(x > 2) \Rightarrow (-1 > 0))$$

$$\equiv ((x > 2) \Rightarrow True) \wedge ((x \leq 2) \Rightarrow False)$$

$$\equiv x > 2$$

(If you are unhappy with the last step, draw a truth table.)

Alternative Rule for Conditionals (Rule 3b/4)

The conditional rule tends to produce complicated logical expressions which we then have to simplify.

It is often easier to deal with disjunctions and conjunctions than implications, so the following ***equivalent*** rule for conditionals is usually more convenient.

$$wp(\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2, Q) \equiv (b \wedge wp(S_1, Q)) \vee (\neg b \wedge wp(S_2, Q))$$

Conditional Example Again:

$$wp(\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2, Q) \equiv (b \wedge wp(S_1, Q)) \vee (\neg b \wedge wp(S_2, Q))$$

$$wp(\texttt{if x>2 then y:=1 else y:=-1}, (y > 0))$$

$$\equiv ((x > 2) \wedge wp(\texttt{y:=1}, (y > 0)) \vee (\neg(x > 2) \wedge wp(\texttt{y:=-1}, (y > 0))$$

$$\equiv ((x > 2) \wedge (1 > 0)) \vee (\neg(x > 2) \wedge (-1 > 0))$$

$$\equiv ((x > 2) \wedge True) \vee (\neg(x > 2) \wedge False)$$

$$\equiv (x > 2) \vee False$$

$$\equiv x > 2$$

(Again, any step you are unhappy with can be confirmed via truth table.)

Why The Rules are Equivalent

All that has changed is the form of the proposition. Rather than

$$(b \Rightarrow p) \wedge (\neg b \Rightarrow q)$$

we have

$$(b \wedge p) \vee (\neg b \wedge q) \;:$$

b	p	q	$(b \Rightarrow p)$	$\wedge$	$(\neg b \Rightarrow q)$	$(b \wedge p)$	$\vee$	$(\neg b \wedge q)$
T	T	T	T	**T**	T	T	**T**	F
T	T	F	T	**T**	T	T	**T**	F
T	F	T	F	**F**	T	F	**F**	F
T	F	F	F	**F**	T	F	**F**	F
F	T	T	T	**T**	T	F	**T**	T
F	T	F	T	**F**	F	F	**F**	F
F	F	T	T	**T**	T	F	**T**	T
F	F	F	T	**F**	F	F	**F**	F

Conditionals Without 'Else'

It is sometimes convenient to have conditionals without `else`, i.e.

```
if b then S
```

recalling that this is just a compact way of writing

```
if b then S else x := x
```

We can derive wp rules for this case:

$$\begin{aligned} wp(\textbf{if } b \textbf{ then } S, Q) &\equiv (b \Rightarrow wp(S_1, Q)) \ \wedge \ (\neg b \Rightarrow Q) \\ &\equiv (b \wedge wp(S_1, Q)) \ \vee \ (\neg b \wedge Q) \end{aligned}$$