

Class 24 – 4/22

Proving properties of imperative programs –
“Hoare logic”

- Judgments, a.k.a. “Hoare formulas”
- Axioms
- Rules of inference

Correctness of imperative programs

- “Hoare formula” says that if the variables in a program satisfy some properties, then after executing a given program, they satisfy some different properties.
- Examples:

$$x > 0 \{ \text{while} (x > 0) \\ \{ y := y * x; x := x - 1; \} \} y = y * x!$$
$$x = x_0 \ \& \ y = y_0 \{ t := x; x := y; y := t \} \\ x = y_0 \ \& \ y = x_0$$

true { if ($x < 0$) $x := -x$; } $x = |x|$

true { $n := \text{length}(a)$; $b := [\text{hd } a]$;

$a := \text{tl } a$;

while ($a \neq []$) {

$b = (\text{hd } a + \text{hd } b) :: b$;

$a = \text{tl } a$; }

} $b_i = \sum_{k=0}^{n-i-1} a_k$ (where $b_i = \text{hd } (\text{tl}^i b)$,
and similarly for a_k)

Inference rules of Hoare logic

Judgments: $P \{S\} Q$

P, Q assertions about variables in the program

S a statement in this language:

$\text{Stmt} \rightarrow \text{Var} := \text{Expr} \mid \text{Stmt}; \text{Stmt}$
| $\text{if} (\text{Expr}) \text{ then } \text{Stmt} \text{ else } \text{Stmt}$
| $\text{while} (\text{Expr}) \text{ Stmt}$

Inference rules of Hoare logic

$$\frac{}{P[e/x] \{x := e\} P}$$

$$\frac{P \ \& \ b \ \{S\} \ P}{P \ \{while \ (b) \ S\} \ P \ \& \ \neg b}$$

$$\frac{P \ \{S_1\} \ Q \quad Q \ \{S_2\} \ R}{P \ \{S_1; S_2\} \ R}$$

$$\frac{P \Rightarrow P' \quad P' \ \{S\} \ Q' \quad Q' \Rightarrow Q}{P \ \{S\} \ Q}$$

$$\frac{P \ \& \ b \ \{S_1\} \ Q \quad P \ \& \ \neg b \ \{S_2\} \ Q}{P \ \{if \ (b) \ then \ S_1 \ else \ S_2\} \ Q}$$

Rule of assignment:

$$P[e/x] \{x := e\} P$$

Examples

$$y=2 \{ x:=y \} x=2$$

$$y=2 \{ x:=2 \} y=x$$

$$x+1=n+1 \{ x:=x+1 \} x=n+1$$

$$x+1=n \{ x:=x+1 \} x=n$$

$$\text{true} \{ x:=2 \} x=2$$

Rule of consequence

$$\frac{P \Rightarrow P' \quad P' \{S\} Q' \quad Q' \Rightarrow Q}{P \{S\} Q}$$

Sequence rule

$$\frac{P \{S_1\} Q \quad Q \{S_2\} R}{P \{S_1; S_2\} R}$$

If rule

$$\frac{P \ \& \ b \ \{S_1\} \ Q \quad P \ \& \ \neg b \ \{S_2\} \ Q}{P \ \{\text{if } (b) \ \text{then } S_1 \ \text{else } S_2\} \ Q}$$

While rule

$$\frac{P \ \& \ b \ \{S\} \ P}{P \ \{\text{while } (b) \ S\} \ P \ \& \ \neg b}$$

Comments on Hoare logic

- Proofs in Hoare logic are *almost* syntax-directed, i.e. almost have the same shape as the program being proved. The only exceptions are the uses of the rule of consequence.
- Applying Hoare rules is largely mechanical – given A and Q, most of the proof (including P) can be generated automatically. Creativity is required mainly in determining the invariant in a while loop, because Q may not have the form “ $P \ \& \ \neg b$ ”, and so a formula of that form needs to be found (after which the rule of consequence can be used, proving $P \ \& \ \neg b \Rightarrow Q$).

Example: gcd algorithm

```
 $a > 0 \ \& \ b > 0 \ \& \ a=a_0 \ \& \ b=b_0 \{$   
  while ( $a \neq b$ )  
    if ( $a > b$ ) then  $a := a - b$ ;  
    else  $b := b - a$ ;  
  }  $a = \text{gcd}(a_0, b_0)$ 
```