Resume : Formal Methods

January 25, 2018

Chapter 1

Hoare Logic

1.1 Hoare Triple

Logical formulas can be used to express information about program states. We called $\{P\}S\{Q\}$ a "Hoare Triple" :

- A precondition P what can be assumed to be true before executing a sequence of statements S.
- A postcondition Q states what will be true after the execution of S.

We write $\{P\}S\{Q\}$ to indicate : if P is true, then executing S will make Q true.
We also use the notation x' to denote the variable x after the execution of S. For example

$$\{true\}\ x = x + 1;\ \{x' = x + 1\}$$

is a valid Hoare Triple in which x' is the value of x after the execution of S. We can write any predicate in between any two lines of code (an assertion), we assume that such a predicate is the postcondition of the previous line and the precondition of the following line.

1.2 Correctness of Hoare Triple

We translate our program S into a formula ϕ_S and then, we check

$$P \land \phi_S \to Q$$

or, equivalently

$$\phi_S \to (P \to Q)$$

So, for example, we turn the following Hoare Triple

$$\{x \neq 0\}\ x = 1/x;\ x = 1/x;\ \{x' = x\}$$

into the following formula (also using the primed notation) :

$$\underbrace{x \neq 0}_{P} \land \underbrace{x'' = 1/x \land x' = 1/x''}_{S} \to \underbrace{x' = x}_{Q}$$

Which is true by elementary arithmetic.

1.2.1 If Clauses

We turn

$$\{P\}\ if(condition)\ \{prog1\}\ else\ \{prog2\};\ \{Q\}$$

into

$$\{P \wedge condition\}\ prog1\ \{Q\}$$

and

$$\{P \wedge \neg condition\}\ prog2\ \{Q\}$$

Both of those Hoare triple must be true for the if clause to be correct.

1.2.2 Loops Clauses

In order to check the total correctness of a program with loops, we have to check the partial correctness and the termination of the program. Such a program is in the following form :

$$\{P\}\ initialisation;\ while\ (condition)\ \{loop\ body\};\ \{Q\}$$

Partial correctness

A loop invariant is a logical formula that is true

- before the loop,
- before each execution of the loop body,
- after each execution of the loop body,
- after the loop.

Then, from

$$\{P\}\ initialisation;\ while\ (condition)\ \{loop\ body\};\ \{Q\}$$

we get

$$\{P\}\ initialisation;\ \{inv\}$$
$$\{inv \wedge condition\}\ loop\ body;\ \{inv\}$$
$$\{inv \wedge \neg condition\}\ skip;\ \{Q\}$$

Termination

A loop variant is an integer-valued expression that

- is decreased at least by 1 in each execution of the loop body,
- cannot go below 0.

Then, from

$$\{P\}\ initialisation;\ while\ (condition)\ \{loop\ body\};\ \{Q\}$$

we get

$$\{int\ var \wedge var > 0\}\ loop\ body;\ \{var > var' \geq 0\}$$

Example

$$\begin{aligned} \{n > 0 \wedge x = 1\}\ & sum = 1; \\ & while\ (x < n)\ \{ \\ & x = x + 1; \\ & sum = sum + x; \\ & \}; \{sum = n(n+1)/2\} \end{aligned}$$

For the invariant, we try $sum = x(x+1)/2$ and we get the following Hoare Triples :

$$\{n > 0 \wedge x = 1\}\ sum = 1;\ \{sum = x(x+1)/2\}$$
$$\{sum = x(x+1)/2 \wedge x < n\}\ x = x + 1;\ sum = sum + x\ \{sum = x(x+1)/2\}$$
$$\{sum = x(x+1)/2 \wedge \neg(x < n)\}\ skip;\ \{sum = n(n+1)/2\}$$

(1) : We obtain the following formula to prove :

$$n > 0 \wedge x = 1 \wedge sum = 1 \rightarrow sum = \frac{x(x+1)}{2}$$

$1 = \frac{1(1+1)}{2} = \frac{2}{2} = 1$

(2) : We obtain the following formula to prove :

$$sum = x(x+1)/2 \wedge x < n \wedge x' = x + 1 \wedge sum' = sum + x' \rightarrow sum' = x'(x'+1)/2$$

$$\begin{aligned} & sum = \frac{x(x+1)}{2} \\ & x < n \\ & x' = x + 1 \\ & sum' = sum + x' = sum + x + 1 = \frac{x(x+1)}{2} + x + 1 \\ & sum' = \frac{x'(x'+1)}{2} \\ & \frac{x(x+1)}{2} + x + 1 = \frac{(x+1)(x+1+1)}{2} \\ & \frac{x(x+1)}{2} + x + 1 = \frac{x^2+x}{2} + x + 1 = 0.5x^2 + 1.5x^2 + 1 \\ & \frac{(x+1)(x+1+1)}{2} = \frac{x^2+2+2x+x}{2} = \frac{x^2+3x+2}{2} = 0.5x^2 + 1.5x + 1 \end{aligned}$$

(3) : We obtain the following formula to prove :

$$sum = \frac{x(x+1)}{2} \wedge x \geq n \rightarrow sum = \frac{n(n+1)}{2}$$

Which is true because if $x = n$ then both side of the equation are equivalent.

termination : we try $n - x$ for the variant and we got the following formula :

$$\begin{aligned}
& int\ var \wedge \\
& var > 0 \wedge \\
& x = 1 \wedge \\
& n > 0 \wedge \\
& var = n - x \wedge \\
& x' = x + 1 \wedge \\
& sum' = sum + x' \wedge \\
& var' = n - x' \ \rightarrow \\
& var > var' \geq 0
\end{aligned}$$

$$var' = n - x' = n - x + 1$$

$$n - x > n - x - 1 \geq 0$$

Termination is proved.

1.3 Weakness and Strength of Predicates

P is weaker than $Q \leftrightarrow Q \rightarrow P$ ($\leftrightarrow$ stand for if and only if). true is the weakest predicate and false is the strongest one.
If P is weaker than P' ($P' \rightarrow P$), then proving $\{P\}\ S\ \{Q\}$ guarantees the truth of $\{P'\}\ S\ \{Q\}$.
If Q is stronger than Q' ($Q \rightarrow Q'$), then proving $\{P\}\ S\ \{Q\}$ guarantees the truth of $\{P\}\ S\ \{Q'\}$.

Example : assume that we have proved

$$\begin{aligned}
& \{x > 0\} \\
& x = 1/x; \\
& \{x' > 0\} \\
& \{x \neq 0\} \\
& \text{if } (x < 0)\text{then } x = -x \text{ else } x = x; \\
& \{x' > 0\}
\end{aligned}$$

How to prove

$$\begin{aligned}
& \{x > 0\} \\
& x = 1/x; \\
& \text{if } (x < 0)\text{then } x = -x \text{ else } x = x; \\
& \{x' > 0\}
\end{aligned}$$

We can assume that both of the assertion present in the first one are just replaced by the assertion $\{true\}$. Then, we would have $\{x > 0\}\ x = 1/x\ \{true\}$ where $Q' = \{true\}$. From the first one, we can extract the Hoare Triple $\{x > 0\}\ x = 1/x\ \{x' > 0 \wedge x \neq 0\}$, where $Q = \{x' > 0 \wedge x \neq 0\}$. Q is stronger that Q' and $\{P\}\ S\ \{Q\}$ is proved, then $\{P\}\ S\ \{Q'\}$ is proved too.

Chapter 2

Propositional Logic

A formula in propositional logic can be constructed as follows :

- $\top$ (true), $\bot$ (false) and propositional variables ($q, p, m, m, \dots$) are formulas of propositional logic.
- if F and G are formulas of propositional logic, then so are :
  - (F)
  - $\neg F$
  - $F \wedge G$
  - $F \vee G$
  - $F \rightarrow G$
  - $F \leftrightarrow G$

Definition. *If P is a variable, then P and $\neg P$ are called **literals**.*

Definition. *An **interpretation** I is a truth-value assignment to propositional variables $P, Q, \dots$*

$$I : \{P \mapsto \top, Q \mapsto \bot, \dots\}$$

Definition. *The truth value of a variable P under an interpretation I is denoted by $I[P]$. For example, we would have*

$$I[P] = \top, I[Q] = \bot, \dots$$

Definition. *For an interpretation I, we write*

$$I \models F$$

if and only if propositional formula F is true under the interpretation I.

Theorem 1. *We have the following for F and G being a propositional variables :*

- $I \models \top$
- $I \not\models \bot$
- $I \models F$ *iff* $I[F] = \top$
- $I \not\models F$ *iff* $I[F] = \bot$
- $I \models (F)$ *iff* $I \models F$
- $I \models \neg F$ *iff* $I \not\models F$
-

Chapter 3

Computability

3.1 Undecidability of First-order Logic

Chapter 4

Complexity

4.1 Cook's Theorem

4.2 NP-completeness

4.3 PSPACE-Completeness

Chapter 5

Polynomial Time Reductions