A very brief introduction to program verification

Robert L. Baber

for 2F04

2001 November 30
Overview

1. Precondition, postcondition
2. Selected lemmata (“proof rules”)
   2.1 assignment statement
   2.2 while loop
3. Correctness proof for a simple program (example)
4. Conclusions
If the truth of a condition $V$ before executing a program statement $S$ guarantees that a condition $P$ is true afterward, then we say that $V$ is a *precondition* of the *postcondition* $P$ with respect to the program statement $S$, written $\{V\} S \{P\}$.

(ambiguous but can be reformulated mathematically)
Proof rules: assignment statement

A1: \( \{P^x_{(E)}\} \ x := E \ \{P\} \)
(the value of x after = the value of E before, 
so P after = \(P^x_{(E)}\) before)

A2: If V \(\Rightarrow\) \(P^x_{(E)}\)
then \(\{V\} \ x := E \ \{P\}\)

Use A1 to derive a precondition, 
A2, to verify a precondition.
Proof rule: while loop

If

\{V\} \text{ init} \ {I}\ and
\{I \land B\} \ S \ {I}\ and
I \land \neg B \implies P

then

\{V\} \text{ init; while } B \text{ do } S \text{ endwhile} \ {P}\n
“I” is the loop invariant.
Example of a correctness proposition

\{n \in \mathbb{Z} \land 0 \leq n\} \quad [V]

i := 1

while i \leq n and A(i) \neq key do \quad [B]

\quad i := i + 1

invariant \{n \in \mathbb{Z} \land i \in \mathbb{Z} \land 1 \leq i \leq n+1 \land \sum_{j=1}^{i-1} A(j) \neq key\} \quad [I]

endwhile

\{n \in \mathbb{Z} \land i \in \mathbb{Z} \land 1 \leq i \leq n+1 \land \sum_{j=1}^{i-1} A(j) \neq key
\land (i = n+1 \lor A(i) = key)\} \quad [P]
By the applicable proof rules, the program will be correct (satisfy its specification V, P) if:

1. \{V\} \texttt{i:=1} \{I\}, i.e. if \( V \Rightarrow I^i_1 \)

2. \{I \land B\} \texttt{i:=i+1} \{I\}, i.e. if \( I \land B \Rightarrow I^i_{(i+1)} \)

3. \( I \land \neg B \Rightarrow P \)
1. $V \Rightarrow I_{1}^{i}$

$I_{1}^{i}$

$= n \in \mathbb{Z} \land 1 \in \mathbb{Z} \land 1 \leq l \leq n+1 \land _{j=1}^{1-1} A(j) \neq \text{key}$

$= n \in \mathbb{Z} \land 0 \leq n$

$= V$
2. $I \land B \Rightarrow I^i_{(i+1)}$ ?

\[
I^i_{(i+1)} = \begin{align*}
&n \in \mathbb{Z} \land i+1 \in \mathbb{Z} \land 1 \leq i+1 \leq n+1 \land \bigwedge_{j=1}^{i+1-1} A(j) \neq \text{key} \\
= &\quad n \in \mathbb{Z} \land i \in \mathbb{Z} \land 0 \leq i \leq n \land \bigwedge_{j=1}^{i} A(j) \neq \text{key} \\
\iff &\quad n \in \mathbb{Z} \land i \in \mathbb{Z} \land 1 \leq i \leq n \land \bigwedge_{j=1}^{i-1} A(j) \neq \text{key} \land A(i) \neq \text{key} \\
= &\quad I \land B
\end{align*}
\]
3. $I \land \neg B \Rightarrow P$?

$I \land \neg B$

= 

$n \in \mathbb{Z} \land i \in \mathbb{Z} \land 1 \leq i \leq n+1 \land \sum_{j=1}^{i-1} A(j) \neq \text{key} \land (i > n \lor A(i) = \text{key})$

= 

$n \in \mathbb{Z} \land i \in \mathbb{Z} \land 1 \leq i \leq n+1 \land \sum_{j=1}^{i-1} A(j) \neq \text{key} \land (i = n+1 \lor A(i) = \text{key})$

= 

$P$
Conclusions

- Mathematical verification eliminates guesswork (+1?, 0?, -1?, n+1?, n?, n-1?)
- Use proof rules to decompose correctness proposition to be proved (to Boolean ⇒s).
- Proving is a mechanistic process; creativity is in design, not verification.

- But most importantly: Use proof rules as design guidelines → program correct by design
Further information

- 4/6L03 (MRSD) lecture notes at http://www.cas.mcmaster.ca/~baber/Courses/46L03/MRSDLect.pdf
- Other literature in 4/6L03 course outline http://www.cas.mcmaster.ca/~baber/Courses/46L03/COut46L03.html