Algorithm for Proving Hoare Triples?

- Have seen in Isabelle that much of proving a Hoare triple is routine.
- Why not automate the whole process?
  - Can’t (always) calculate needed loop invariants.
  - Can’t (always) prove implications (side-conditions) in Rule of Consequence application.
- Can we automate all but this?
  1. Annotate all while loops with needed invariants.
  2. Use routine to “roll back” post-condition to weakest precondition, gathering side-conditions as we go.
- 2 called verification condition generation.

Annotated Simple Imperative Language

- Give verification conditions for an annotated version of our simple imperative language.
- Add a presumed invariant to each while loop.

\[
\begin{align*}
(\text{command}) := & (\text{variable}) := (\text{term}) \\
& | (\text{command}) ; \ldots ; (\text{command}) \\
& | \text{if (statement) then (command) else (command)} \\
& | \text{while (statement) inv (statement) do (command)}
\end{align*}
\]

Example:

\[
\begin{align*}
\text{while } y < n \text{ inv } x &= y \ast y \\
\text{do } x &= (2 \ast y) + 1; \\
y &= y + 1 \\
\text{od}
\end{align*}
\]

HOL Type for Deep Part of Embedding

```
datatype 'data annotated_command =
  AnnAssignCom "var_name" "'data exp"
  (infix ":=" 110)
| AnnSeqCom "'data annotated_command"
  "'data annotated_command"
  (infixi ":;" 109)
| AnnCondCom "'data bool_exp"
  "'data annotated_command"
  ("If / Then / Else / Fi" [70,70,70]70)
| AnnWhileCom "'data bool_exp" "'data annotated_command"
  ("While / Inv / Do / Od" [70,70]70)
```

Hoare Logic for Annotated Programs

- Assignment Rule
- Rule of Consequence
- Sequencing Rule
- If Then Else Rule
- While Rule

\[
\begin{align*}
\langle P[e/x]\rangle x &= e \langle P\rangle \\
\langle P'\rangle C \langle Q\rangle &\Rightarrow Q' \Rightarrow Q \\
\langle P\rangle C_1 \langle Q\rangle &\Rightarrow \langle C_2 \langle R\rangle\rangle \\
\langle P\rangle C_1 \langle Q\rangle &\Rightarrow \langle P \land B\rangle C_1 \langle Q\rangle \langle P \land \neg B\rangle C_2 \langle Q\rangle \\
\langle P\rangle &\Rightarrow \langle P \land B\rangle C \langle P \land \neg B\rangle
\end{align*}
\]
Relation Between Two Hoare Logics

Theorem
For all pre- and post-conditions $P$ and $Q$, and annotated programs $C$, if $\{P\} C \{Q\}$, then $\{\text{strip}(C)\} \{Q\}$.

Proof.
(Sketch) Use rule induction on proof of $\{P\} C \{Q\}$; in case of While Rule, erase invariant.

Weakest Justification

Definition

$\forall C. P \implies Q \implies (wp C P \implies wp C Q)$

Assumes, without verifying, that $P$ is the correct invariant.
What About Precondition?

**Question:** Do we have \( \{ wp \ C \ Q \} \ C \{ Q \} \)?

**Answer:** Not always - need to check while-loop side-conditions – verification conditions

**Question:** How to calculate verification conditions?

### Definition

\[
\begin{align*}
\text{vcg } (x := e) & : Q \Rightarrow \text{true} \\
\text{vcg } (C_1 \ C_2) & : Q \Rightarrow (\text{vcg } C_1 \ Q) \land (\text{vcg } C_2 \ Q) \\
\text{vcg } (\text{if } B \text{ then } C_1 \text{ else } C_2 \text{ fi}) & : Q \Rightarrow (\text{vcg } C_1 \ Q) \land (\text{vcg } C_2 \ Q) \\
\text{vcg } (\text{while } B \text{ inv } P \text{ do } C \text{ od}) & : Q \Rightarrow ((P \land B) \Rightarrow (wp \ C \ P)) \land ((P \land \neg B) \Rightarrow Q)
\end{align*}
\]

Verification Condition Guarantees wp Precondition

**Theorem**

\( \text{vcg } C \ Q \Rightarrow \{ wp \ C \ Q \} \ C \{ Q \} \)

**Proof.**

(Sketch)

Induct on structure of \( C \)

- For each case, wind back as we did in specific examples:
  - Assignment: \( wp \ C \ Q \) exactly what is needed for Assignment Axiom
  - Sequence: Follows from inductive hypotheses, all elim, and modus ponens
  - If Then Else: Need to use Precondition Strengthening with each branch of conditional; \( wp \) and inductive hypotheses give the needed side conditions
  - While: Need to use Postcondition Weakening, While Rule and Precondition Strengthening

**Corollary**

\( ((P \Rightarrow wp \ C \ Q) \land (\text{vcg } C \ Q)) \Rightarrow \{ P \} \ C \{ Q \} \)

This amounts to a method for proving Hoare triple \( \{ P \} \ C \{ Q \} \):

1. Annotate program with loop invariants (reduces to showing \( \{ P \} \ C \{ Q \} \))
2. Calculate \( wp \ C \ Q \) and \( \text{vcg } C \ Q \) (automated)
3. Prove \( P \Rightarrow wp \ C \ Q \) and \( \text{vcg } C \ Q \)

Basic outline of interaction with Boogie: Human does 1, Boogie does 2, Z3 / Simplify / Isabelle + human / ... does 3

For more information