Proof Rules For Partial Correctness

Overview

- **The proof rules that follow constitute an *axiomatic semantics* of our programming language (more on "axiomatic semantics" later)**

E ::= N | V | $E_1 + E_2$ | $E_1 - E_2$ | $E_1 \times E_2$ | ...

B ::= `T` | `F` | E_1=E_2 | $E_1 \leq E_2$ | ...

C ::= `SKIP`
| V `:=` E
| V`(`E_1`)` `:=` E_2
| C_1 `;` C_2
| `IF` B `THEN` C **(redundant)**
| `IF` B `THEN` C_1 `ELSE` C_2
| `BEGIN VAR` V_1 `; .. VAR` V_1 `;` C `END`
| `WHILE` B `DO` C
| `FOR` V `:=` E_1 `UNTIL` E_2 `DO` C

- **Arrays and `FOR`-commands are tricky and discussed in a later lecture**

- **One-armed conditionals will be used to illustrate derived semantics**

Judgements

- **Three kinds of things that could be true or false:**
  - **statements of mathematics, e.g.** $(\mathtt{X}+1)^2 = \mathtt{X}^2 + 2\times\mathtt{X}+1$
  - **partial correctness specifications** $\{P\}\ C\ \{Q\}$
  - **total correctness specifications** $[P]\ C\ [Q]$
- **These three kinds of things are examples of** *judgements*
  - **a logical system gives rules for proving judgements**
  - **Floyd-Hoare logic provides rules for proving partial correctness specifications**
  - **the laws of arithmetic provide ways of proving statements about integers**
- $\vdash S$ **means statement** S **can be proved**
  - **how to prove predicate calculus statements assumed known**
  - **this course covers axioms and rules for proving**
    program correctness statements

Syntactic Conventions

- **Symbols V, V_1, ... , V_n stand for arbitrary variables**
  - **examples of particular variables are $\mathtt{X}$, $\mathtt{R}$, $\mathtt{Q}$ etc**
- **Symbols E, E_1, ... , E_n stand for arbitrary expressions (or terms)**
  - **these are things like $\mathtt{X}+\mathtt{1}$, $\sqrt{2}$ etc. which denote values (usually numbers)**
- **Symbols S, S_1, ... , S_n stand for arbitrary statements**
  - **these are conditions like $\mathtt{X} < \mathtt{Y}$, $\mathtt{X}^2 = \mathtt{1}$ etc. which are either true or false**
  - **will also use P, Q, R to range over pre and postconditions**
- **Symbols C, C_1, ... , C_n stand for arbitrary commands**

Skip

- **Syntax:** `SKIP`
- **Semantics: the state is unchanged**

The Skip Axiom

$$\vdash\ \{P\}\ \texttt{SKIP}\ \{P\}$$

- **It is an axiom schema**
  - P **can be instantiated with arbitrary predicate calculus formulae (statements)**
- **Instances of the skip axiom are:**
  - $\vdash\ \{\texttt{Y} = 2\}\ \texttt{SKIP}\ \{\texttt{Y} = 2\}$
  - $\vdash\ \{\texttt{T}\}\ \texttt{SKIP}\ \{\texttt{T}\}$
  - $\vdash\ \{\texttt{R=X+(Y} \times \texttt{Q)}\}\ \texttt{SKIP}\ \{\texttt{R=X+(Y} \times \texttt{Q)}\}$

Substitution Notation

- Q[E/V] **is the result of replacing all occurrences of** V **in** Q **by** E
  - **read** Q[E/V] **as '**Q **with** E **for** V**'**
  - **for example:** $(\texttt{X}{+}1 > \texttt{X})\texttt{[Y+Z/X]} \;=\; ((\texttt{Y+Z}){+}1 > \texttt{Y+Z})$
  - **ignoring issues with bound variables for now (e.g. variable capture)**
- **Same notation for substituting into terms, e.g.** E_1[E_2/V]
- **Think of this notation as the 'cancellation law'**

$$V\texttt{[}E/V\texttt{]} \;=\; E$$

which is analogous to the cancellation property of fractions

$$v \times (e/v) \;=\; e$$

- **Note that** Q[x/V] **doesn't contain** V **(if** $V \neq x$**)**

The Assignment Axiom (Hoare)

- **Syntax:** V `:=` E
- **Semantics: value of V in final state is value of E in initial state**
- **Example:** `X:=X+1` **(adds one to the value of the variable** `X`**)**

The Assignment Axiom

$$\vdash \ \{Q\texttt{[}E/V\texttt{]}\}\ V\texttt{:=}E\ \{Q\}$$

Where V is any variable, E is any expression, Q is any statement.

- **Instances of the assignment axiom are**
  - $\vdash\ \{\mathtt{E} = \mathtt{x}\}\ \mathtt{V} := \mathtt{E}\ \{\mathtt{V} = \mathtt{x}\}$
  - $\vdash\ \{\mathtt{Y} = \mathtt{2}\}\ \mathtt{X} := \mathtt{2}\ \{\mathtt{Y} = \mathtt{X}\}$
  - $\vdash\ \{\mathtt{X} + \mathtt{1} = \mathtt{n} + \mathtt{1}\}\ \mathtt{X} := \mathtt{X} + \mathtt{1}\ \{\mathtt{X} = \mathtt{n} + \mathtt{1}\}$
  - $\vdash\ \{E = E\}\ \mathtt{X} := E\ \{\mathtt{X} = E\}$ **(if** `X` **does not occur in** E**)**

The Backwards Fallacy

- **Many people feel the assignment axiom is 'backwards'**

- **One common erroneous intuition is that it should be**

$$\vdash \{P\}\ V\texttt{:=}E\ \{P[V/E]\}$$

  - **where** $P[V/E]$ **denotes the result of substituting** V **for** E **in** P
  - **this has the false consequence** $\vdash$ $\{$`X=0`$\}$ `X:=1` $\{$`X=0`$\}$
    (since `(X=0)[X/1]` **is equal to** `(X=0)` **as** `1` **doesn't occur in** `(X=0)`**)**

- **Another erroneous intuition is that it should be**

$$\vdash \{P\}\ V\texttt{:=}E\ \{P[E/V]\}$$

  - **this has the false consequence** $\vdash$ $\{$`X=0`$\}$ `X:=1` $\{$`1=0`$\}$
    (which follows by taking P **to be** `X=0`**,** V **to be** `X` **and** E **to be** `1`**)**

A Forwards Assignment Axiom (Floyd)

- **This is the original semantics of assignment due to Floyd**

$$\vdash \ \{P\}\ V\texttt{:=}E\ \{\exists v.\ V = E\texttt{[}v/V\texttt{]}\ \wedge\ P\texttt{[}v/V\texttt{]}\}$$

  - **where v is a new variable (i.e. doesn't equal V or occur in P or E)**

- **Example instance**

$$\vdash \ \{\texttt{X=1}\}\ \texttt{X:=X+1}\ \{\exists v.\ \texttt{X} = \texttt{X+1[}v/\texttt{X]}\ \wedge\ \texttt{X=1[}v/\texttt{X]}\}$$

- **Simplifying the postcondition**

$$\vdash \ \{\texttt{X=1}\}\ \texttt{X:=X+1}\ \{\exists v.\ \texttt{X} = \texttt{X+1[}v/\texttt{X]}\ \wedge\ \texttt{X=1[}v/\texttt{X]}\}$$
$$\vdash \ \{\texttt{X=1}\}\ \texttt{X:=X+1}\ \{\exists v.\ \texttt{X} = v + \texttt{1}\ \wedge\ v = \texttt{1}\}$$
$$\vdash \ \{\texttt{X=1}\}\ \texttt{X:=X+1}\ \{\exists v.\ \texttt{X} = \texttt{1} + \texttt{1}\ \wedge\ v = \texttt{1}\}$$
$$\vdash \ \{\texttt{X=1}\}\ \texttt{X:=X+1}\ \{\texttt{X} = \texttt{1} + \texttt{1}\ \wedge\ \exists v.\ v = \texttt{1}\}$$
$$\vdash \ \{\texttt{X=1}\}\ \texttt{X:=X+1}\ \{\texttt{X} = \texttt{2}\ \wedge\ \texttt{T}\}$$
$$\vdash \ \{\texttt{X=1}\}\ \texttt{X:=X+1}\ \{\texttt{X} = \texttt{2}\}$$

- **Forwards Axiom equivalent to standard one but harder to use**

Precondition Strengthening

- **Recall that**

$$\frac{\vdash\ S_1,\ \ldots\ ,\ \vdash\ S_n}{\vdash\ S}$$

means $\vdash\ S$ **can be deduced from** $\vdash\ S_1,\ \ldots\ ,\ \vdash\ S_n$

- **Using this notation, the rule of precondition strengthening is**

Precondition strengthening

$$\frac{\vdash\ P \Rightarrow P', \qquad \vdash\ \{P'\}\ C\ \{Q\}}{\vdash\ \{P\}\ C\ \{Q\}}$$

- **Note the two hypotheses are different kinds of judgements**

Example

- **From**
  - $\vdash$ `X=n` $\Rightarrow$ `X+1=n+1`
    - **trivial arithmetical fact**
  - $\vdash \{\mathtt{X}+1=\mathtt{n}+1\}\ \mathtt{X}:=\mathtt{X}+1\ \{\mathtt{X}=\mathtt{n}+1\}$
    - **from earlier slide**
- **It follows by precondition strengthening that**

$$\vdash \{\mathtt{X}=\mathtt{n}\}\ \mathtt{X}:=\mathtt{X}+1\ \{\mathtt{X}=\mathtt{n}+1\}$$

- **Note that `n` is an *auxiliary* (or *ghost*) variable**

Postcondition weakening

- **Just as the previous rule allows the precondition of a partial correctness specification to be strengthened, the following one allows us to weaken the postcondition**

Postcondition weakening

$$\frac{\vdash\ \{P\}\ C\ \{Q'\}, \qquad \vdash\ Q' \Rightarrow Q}{\vdash\ \{P\}\ C\ \{Q\}}$$