Lecture 2
Examples

• Which programs satisfy $\{\text{true}\}P\{\text{false}\}$?

• Which programs satisfy $<\text{true}>P<\text{false}>$?
Logical Variables in Specifications

Example 1:
Specify a program with a single variable $x$ whose value at the end of the computation is twice its value at the beginning
Logical Variables in Specifications

Solution: add fresh variables which are
- not part of the program and therefore
- their value does not change during the execution of the program

These variables are called logical variables

Convention: We use logical variable $X$ to preserve the value of variable $x$
Logical Variables in Specifications

Example 2:
Program which returns in variable z the multiplication of variables x and y

Convention:
Assertions \( q_1, q_2 \) are now defined over \( \bar{x} \) that includes both program variables and logical variables.
Ingredients for Formal Verification

1. Specification language
   • With formal semantics

2. Programming language
   • with formal semantics

3. Proof rules
   • For proving “Program P has the property \( \varphi \)“
While Programs: Syntax

\[ S ::= x := e \mid \text{skip} \mid S_1 ; S_2 \mid \]
\[ \quad \text{if } B \text{ then } S_1 \text{ else } S_2 \text{ fi} \mid \]
\[ \quad \text{while } B \text{ do } S \text{ od} \]

- **B** - condition - predicate over program variables
- **e** - expression over program variables
- **skip**, **x := e** - atomic actions
- The rest - compound actions
Operational semantics of while Programs

• **Configuration** of a program is a pair $C=\langle S, \sigma \rangle$ such that $S$ is a while program and $\sigma$ is a program state.

• The program $E$ is the empty program such that for every program $S$:
  $E;S = S;E = S$

• A configuration is **halting** if $S=E$
Relation → Over Configurations

We defined
\[ \langle S, \sigma \rangle \rightarrow \langle S', \sigma' \rangle \]

Was defined in details in the tutorial
For example:
\[ \langle x:=e, \sigma \rangle \rightarrow \langle E, \sigma[x\leftarrow \sigma(e)] \rangle \]
Relation $\rightarrow$ Over Configurations

• $\rightarrow^*$ denotes the transitive closure of $\rightarrow$

• Computation from configuration $C$ is denoted $\pi(C)$

• $\pi(C)$ is a maximal series of configurations $C_0, C_1, \ldots$ such that $C = C_0$ and for every $i \geq 0$ $C_i \rightarrow C_{i+1}$
Definition of $\text{val}(\pi)$ - revisited

- $\text{val}(\pi(C)) =$
  - $\sigma'$ if $C \xrightarrow{*} (E, \sigma')$
  - $\perp$ if $\pi(C)$ is infinite

- The **meaning** or **semantics** of a program $S$ is:
  $M[S](\sigma) = \text{val}(\pi(S,\sigma))$
Ingredients for Formal Verification

1. Specification language
   • With formal semantics

2. Programming language
   • with formal semantics

3. Proof rules
   • For proving “Program P has the property $\phi$”
Hoare Proof System $H$ for Partial Correctness

- Deductive proof system:
  - Axioms, inference rules

- Modular proof system
Hoare Proof System for Partial Correctness

Assignment axiom:
\{q[x\leftarrow e]\} x:=e \{q\}

Skip axiom:
\{q\} skip \{q\}
Hoare Proof System for Partial Correctness

Inference rule for sequential composition

\[(\text{SEQ}) \quad \frac{\{p\} S_1 \{r\} \quad \{r\} S_2 \{q\}}{\{p\} S_1; S_2 \{q\}}\]

Can be extended to n programs

\[S_1; S_2; \ldots; S_n\]
Hoare Proof System for Partial Correctness

Example: prove the correction of swap

\[ \vdash_{H} \{ x=X \land y=Y \} \ z:=x; \ x:=y; \ y:=z \ \{ x=Y \land y=X \} \]

1. \[ \{ x=Y \land z=X \} \ y:=z \ \{ x=Y \land y=X \} \ (ASS) \]
2. \[ \{ y=Y \land z=X \} \ x:=y \ \{ x=Y \land z=X \} \ (ASS) \]
3. \[ \{ y=Y \land z=X \} \ x:=y; \ y:=z \ \{ x=Y \land y=X \} \ (SEQ \ 1,2) \]
4. \[ \{ y=Y \land x=X \} \ z:=x \ \{ y=Y \land z=X \} \ (ASS) \]
5. \[ \{ y=Y \land x=X \} \ z:=x; \ x:=y; y:=z \ \{ x=Y \land y=X \} \ (SEQ \ 3,4) \]
Hoare Proof System for Partial Correctness

Inference rule for if

\[
\frac{\{p \land B\} \; S_1 \; \{q\}, \; \{p \land \neg B\} \; S_2 \; \{q\}}{\{p\} \; \text{if } B \; \text{then } S_1 \; \text{else } S_2 \; \text{fi } \; \{q\}} \]

(p is the loop invariant)

Inference rule for while

\[
\frac{\{p \land B\} \; S \; \{p\}}{\{p\} \; \text{while } B \; \text{do } S \; \text{od } \; \{p \land \neg B\}} \]

P is the loop invariant
Hoare Proof System for Partial Correctness

Inference rule for consequence (CONS)

\[ p \rightarrow p_1 , \{p_1\} \quad S \quad \{q_1\} , \quad q_1 \rightarrow q \]

\[ \{p\} \quad S \quad \{q\} \]

Example of non-completeness of H without CONS:

\[ \{x=0\} \quad \text{while true do } x:=x-1 \quad \text{od } \{x=0 \land \text{false}\} \]
Hoare Proof System for Partial Correctness

Example: correction proof
\[ \vdash_H \{ x=0 \} \text{ while } x<10 \text{ do } x:=x+1 \text{ od } \{ x=10 \} \]

Choose an invariant: \( p = x \leq 10 \)
1. \( \{ x+1 \leq 10 \} \ x:=x+1 \ \{ x \leq 10 \} \) (ASS)
2. \( x \leq 10 \land x < 10 \rightarrow x+1 \leq 10 \) (ARITH)
3. \( \{ x \leq 10 \land x < 10 \} \ x:=x+1 \ \{ x \leq 10 \} \) (CONS 1,2)
4. \( \{ x \leq 10 \} \text{ while } x<10 \text{ do } x:=x+1 \text{ od } \{ x \leq 10 \land \neg(x<10) \} \) (REP)
5. \( x=0 \rightarrow ( x \leq 10 ) \) (ARITH)
6. \( ( x \leq 10 \land \neg(x<10) ) \rightarrow x=10 \) (ARITH)
7. \( \{ x=0 \} \text{ while } x<10 \text{ do } x:=x+1 \text{ od } \{ x=10 \} \) (CONS 4,5,6)
Soundness of Hoare proof system (H):

If $\vdash_H \{p\} S \{q\}$ then $\models \{p\} S \{q\}$