UTA

Agenda

- Quick Review
- Finish Logic Review
- Program Proof

UTA

Quick Review

- What are the syntactic objects we can use in first-order logic?

Deductive Verification

- Introduction
- Hoare's Logic
- An Example Proof
- Summary

Deductive Verification

... is to formally verify the correctness of a program using a proof system.

Given a program P and an initial condition I, this approach proves that the expected final condition is the logical consequence of executing P from I.

Program Correctness

- Partial correctness: If the initial condition holds, and if the program terminates, then the input-output claim holds.
- Termination: If the initial condition holds, then the program terminates
- Total correctness: If the initial condition holds, then the program terminates, and the input-output claim holds.

Benefits

- Establishes or enhances the correctness of a program
  - Finding errors in the code and correct them
  - A better understanding of the verified algorithm
- Identifies invariants that can be used to check code consistency during the different development stages.
- A proof system can be used as a formalism to formally define the semantics of a language.

UTA

Limitations

- Can't be completely automated
  - often requires a great deal of domain expertise
- Highly time-consuming
  - only performed on key algorithms
- Errors may crop into the proof
  - who will verify the verifier?
- Has limited scalability
  - difficult to verify large programs

UTA

Deductive Verification

- Introduction
- Hoare Logic
- An Example Proof
- Summary

UTA

While Program

A while program consists of three types of statements:

```
S ::= v:= e | S ; S
      | if p then S else S fi
      | while p do S
```


UTA

Hoare Triple

A Hoare Triple is written as {p} S {q}, where S is a program segment, and p and q are two first order formulas.

The meaning of {p} S {q} is as follows: If execution of S starts with a state satisfying p, and if S terminates, then a state satisfying q is reached.

Assignment Axiom

$$\{p[e/v]\}\ v := e\ \{p\}$$

For example:

{y+5=10} y:=y+5 {y=10}

{y+y<z} x:=y {x+y<z}

{2*(y+5)>20} y:=2*(y+5) {y>20}

Composition Rule

$$\frac{\{p\}\ S1\ \{r\},\ \{r\}\ S2\ \{q\}}{\{p\}\ S1;\ S2\ \{q\}}$$

For example: if the antecedents are

1. {x+1=y+2} x:=x+1 {x=y+2}
2. {x=y+2} y:=y+2 {x=y}

Then the consequent is

{x+1=y+2} x:=x+1; y:=y+2 {x=y}

UTA

If-Then-Else Rule

$$\frac{\{p \wedge c\}\ S1\ \{q\},\ \{p \wedge \neg c\}\ S2\ \{q\}}{\{p\}\ \text{if } c \text{ then } S1 \text{ else } S2 \text{ fi } \{q\}}$$

If the antecedents are

1: $\{(y1 > 0 \wedge y2 > 0 \wedge y1 \mathrel{!}= y2) \wedge (y1 > y2)\}\ y1 := y1 - y2\ \{y1 > 0 \wedge y2 > 0\}$
2: $\{(y1 > 0 \wedge y2 > 0 \wedge y1 \mathrel{!}= y2) \wedge \neg (y1 > y2)\}\ y2 := y2 - y1\ \{y1 > 0 \wedge y2 > 0\}$

Then the consequent is:

$\{(y1 > 0 \wedge y2 > 0 \wedge y1 \mathrel{!}= y2)\}$
if $y1 > y2$ then $y1 := y1 - y2$
else $y2 := y2 - y1$
$\{y1 > 0 \wedge y2 > 0\}$

UTA

While Rule (1)

The while rule uses an invariant, which needs to hold before and after each iteration.

$$\frac{\{p \wedge c\}\ S\ \{p\}}{\{p\}\ \text{while } c \text{ do } S \text{ end } \{p \wedge \neg c\}}$$

While Rule (2)

$$\frac{\{(y1 > 0 \wedge y2 > 0) \wedge y1 \neq y2\}\ \text{if } y1 > y2 \text{ then } y1 := y1 - y2 \text{ else } y2 := y2 - y1\ \{y1 > 0 \wedge y2 > 0\}}{\{y1 > 0 \wedge y2 > 0\}\ \text{while } y1 \neq y2 \text{ do if } y1 > y2 \text{ then } y1 := y1 - y2 \text{ else } y2 := y2 - y1\ \text{end}\ \{y1 > 0 \wedge y2 > 0\} \wedge \{y1 == y2\}}$$

Strengthening Rule

This rule is used to strengthen a precondition:

$$\frac{p \rightarrow r,\ \{r\}\ S\ \{q\}}{\{p\}\ S\ \{q\}}$$

Note that this rule is often used together with the assignment axiom: To prove $\{p\}\ v := e\ \{q\}$, we show that $p \rightarrow q\ [e/v]$.

Weakening Rule

This rule is used to weaken a postcondition:

$$\frac{\{p\}\ S\ \{r\},\ r \rightarrow q}{\{p\}\ S\ \{q\}}$$

Deductive Verification

- Introduction
- Hoare Logic
- An Example Proof
- Summary

UTA

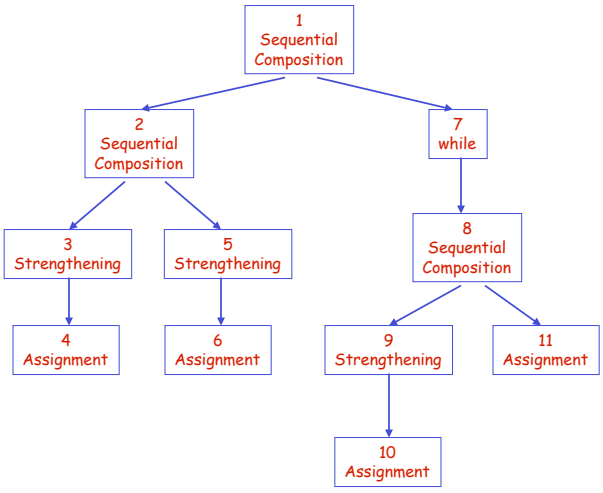
The Program

Below is an program that computes the integer division of x1 by x2.

```
{x1 >= 0 ∧ x2 > 0}
y1 := 0;                    } S1
y2 := x1;
while y2 >= x2 do
   y1 := y1 + 1;            } S3
   y2 := y2 - x2;
end                                  } S2
{x1 ≡ y1 × x2 + y2 ∧ y2 >= 0 ∧ y2 < x2}
```


UTA

A Proof Tree



UTA

Proof Steps (1)

- Goal 1. Sequential Composition

$$\frac{\{x1 >= 0 \wedge x2 > 0\}\ S1\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0\} \quad \{x1 = y1 \times x2 + y2 \wedge y2 >= 0\}\ S2\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0 \wedge y2 < x2\}}{\{x1 >= 0 \wedge x2 > 0\}\ S1\ ;\ S2\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0 \wedge y2 < x2\}}$$

- Goal 2. Sequential Composition

$$\frac{\{x1 >= 0 \wedge x2 > 0\}\ y1 := 0\ \{x1 >= 0 \wedge x2 > 0 \wedge y1 = 0\} \quad \{x1 >= 0 \wedge x2 > 0 \wedge y1 = 0\}\ y2 := x1\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0\}}{\{x1 >= 0 \wedge x2 > 0\}\ S1\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0\}}$$

UTA

Proof Steps (2)

- Goal 3. Strengthening

$$\frac{\{x1 >= 0 \wedge x2 > 0\} \rightarrow \{x1 >= 0 \wedge x2 > 0 \wedge 0 = 0\} \quad \{x1 >= 0 \wedge x2 > 0 \wedge 0 = 0\}\ y1 := 0\ \{x1 >= 0 \wedge x2 > 0 \wedge y1 = 0\}}{\{x1 >= 0 \wedge x2 > 0\}\ y1 := 0\ \{x1 >= 0 \wedge x2 > 0 \wedge y1 = 0\}}$$

- Goal 4. Assignment

$$\{x1 >= 0 \wedge x2 > 0 \wedge 0 = 0\}\ y1 := 0\ \{x1 >= 0 \wedge x2 > 0 \wedge y1 = 0\}$$

UTA

Proof Steps (3)

- Goal 5. Strengthening

$$\frac{\{x1 >= 0 \wedge x2 > 0 \wedge y1 = 0\} \rightarrow \{x1 = y1 \times x2 + x1 \wedge x1 >= 0\} \quad \{x1 = y1 \times x2 + x1 \wedge x1 >= 0\}\ y2 := x1\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0\}}{\{x1 >= 0 \wedge x2 > 0 \wedge y1 = 0\}\ y2 := x1\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0\}}$$

- Goal 6. Assignment

$$\{x1 = y1 \times x2 + x1 \wedge x1 >= 0\}\ y2 := x1\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0\}$$

UTA

Proof Steps (4)

- Goal 7. While

$$\frac{\{x1 = y1 \times x2 + y2 \wedge y2 >= 0 \wedge y2 >= x2\}\ S3\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0\}}{\{x1 = y1 \times x2 + y2 \wedge y2 >= 0\}\ S2\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0 \wedge y2 < x2\}}$$

- Goal 8. Sequential Composition

$$\frac{\{x1 = y1 \times x2 + y2 \wedge y2 >= x2\}\ y1 := y1 + 1\ \{x1 = y1 \times x2 + y2 - x2 \wedge y2 - x2 >= 0\} \quad \{x1 = y1 \times x2 + y2 - x2 \wedge y2 - x2 >= 0\}\ y2 := y2 - x2\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0\}}{\{x1 = y1 \times x2 + y2 \wedge y2 >= x2\}\ S3\ \{x1 = y1 \times x2 + y2 \wedge y2 >= 0\}}$$

UTA

Proof Steps (5)

- Goal 9. Strengthening

$$\frac{\{x1 = y1 \times x2 + y2 \wedge y2 >= x2\} \rightarrow \{x1 = (y1 + 1) \times x2 + y2 - x2 \wedge y2 - x2 >= 0\} \quad \{x1 = (y1 + 1) \times x2 + y2 - x2 \wedge y2 - x2 >= 0\}\ y1 := y1 + 1\ \{x1 = y1 \times x2 + y2 - x2 \wedge y2 - x2 >= 0\}}{\{x1 = y1 \times x2 + y2 \wedge y2 >= x2\}\ y1 := y1 + 1\ \{x1 = y1 \times x2 + y2 - x2 \wedge y2 - x2 >= 0\}}$$

- Goal 10. Assignment

$$\{x1 = (y1 + 1) \times x2 + y2 - x2 \wedge y2 - x2 >= 0\}\ y1 := y1 + 1\ \{x1 = y1 \times x2 + y2 - x2 \wedge y2 - x2 >= 0\}$$

- Goal 11. Assignment

$$\{x1 = y1 \times x2 + y2 - x2 \wedge y2 - x2 >= 0\}\ y2 := y2 - x2\ \{x1 = y1 \times x2 + y2 - x2 \wedge y2 >= 0\}$$

UTA

Annotation

```
{ x1 >= 0 ∧ x2 > 0 }
y1 := 0;
{x1 >= 0 ∧ x2 > 0 ∧ y1 ≡ 0}
y2 : = x1;
{x1 ≡ y1 × x2 + y2 ∧ y2 >= 0}
while y2 >= x2 do
   {x1 ≡ y1 × x2 + y2 ∧ y2 >= x2}
   y1 := y1 + 1;
   {x1 ≡ y1 × x2 + y2 - x2 ∧ y2 - x2 >= 0}
   y2 := y2 - x2;
   {x1 ≡ y1 × x2 + y2 ∧ y2 >= 0}
end
{x1 ≡ y1 × x2 + y2 ∧ y2 >= 0 ∧ y2 < x2}
```

Deductive Verification

- Introduction
- Hoare Logic
- An Example Proof
- Summary

Summary

- Deductive verification can establish and/or enhance the correctness of a program.
- The verification process is highly time-consuming, and can't be completely automated.
- The key to apply Hoare's logic is to find the right "loop invariants".
- The proof process is usually conducted using backward reasoning.