Hoare Triples

- Syntax: $\{P\}\ S\ \{Q\}$
- Intended meaning: If statement S is executed starting in an initial state satisfying P and it terminates, the resulting final state will satisfy Q

Exercise

- Write a specification
- … for a program S that swaps the values of two variables A and B

Exercise

- Is the following valid?

```
{ n > 0 }
sum := 0;
i := 0;
while (i < n) {
    sum := sum – 1;
    i := i – 1;
}
{ sum == n*(n+1)/2}
```

Program Correctness

- Partial correctness vs. total correctness
- Proving program termination usually harder and requires different techniques

Exercise

- What will really happen if we run this program (in practice today)?

```
{ n > 0 }
sum := 0;
i := 0;
while (i < n) {
    sum := sum – 1;
    i := i – 1;
}
{ sum == n*(n+1)/2}
```

Integers in practice

- Integer variables & arithmetic
  - Classical (standard) theory
  - Reality: 32-bit and 64-bit integers
  - Bit-vector arithmetic theory
  - Modular arithmetic

Arrays

- Specify that an array A is sorted
- How can we model arrays?
- The logic we have considered so far is untyped. We can extend this to a typed version (called many-sorted logic)
  - "sort" ≈ "type"
- We will stick to plain untyped logic here

The Array Datatype

- Function symbols used to model arrays
  - length (unary)
    - length(A)
    - We can use A.length as syntactic sugar
  - lookup (binary)
    - lookup(A,i)
    - We can use A[i] as syntactic sugar
  - update (ternary)
    - update(A,i,v)
    - We can use A[i ← v] as syntactic sugar
    - same as array A except at index i where it contains the value v
  - isArray (unary)
    - type information

Exercise

- Specify that an array A is sorted
- Specify that a program S sorts an input array A
- What are appropriate axioms for the array datatype?

Pointers & Heap

- Memory can be modeled as one (or more) giant array
- Naïve modeling may not be sufficient
  - Due to incompleteness
- Separation Logic
- Pointer analysis
- TVLA
  - An abstract-interpretation based approach to verification in the presence of dynamic structures

Hoare Logic

- Inference rules for proving $\vdash \{P\}\, S\, \{Q\}$
- Combines
  - Rules for standard mathematical logic with
  - Rules for reasoning about programming language constructs

Hoare Logic

Basic Constructs

- Assignment statement: $x := e$
- Statement sequencing: $S_1; S_2$
- Conditional: $if\ (E)\ then\ S_1\ else\ S_2$
- Iteration: $while\ (e)\ do\ S$

Statement Sequencing

$$\frac{?}{\vdash \{P\}\ S_1; S_2\ \{Q\}}$$

$$\frac{\vdash \{P\}\ S_1\ \{R\}, \qquad \vdash \{R\}\ S_2\ \{Q\}}{\vdash \{P\} S_1; S_2 \{Q\}}$$

Conditional Statement

$$\frac{?}{\vdash \{P\}\ \text{if } (E) \text{ then } S_1 \text{ else } S_2\ \{Q\}}$$

$$\frac{\vdash \{P \wedge E\}\ S_1\ \{Q\}, \qquad \vdash \{P \wedge \neg E\}\ S_2\ \{Q\}}{\vdash \{P\}\ \text{if } (E) \text{ then } S_1 \text{ else } S_2\ \{Q\}}$$

Iteration

$$\frac{?}{\vdash \{P\}\ \text{while } (E) \text{ do } S\ \{Q\}}$$

$$\frac{\vdash \{P \wedge E\}\ S\ \{P\}, \qquad \vdash (P \wedge \neg E) \Rightarrow Q}{\vdash \{P\}\ \text{while } (E) \text{ do } S\ \{Q\}}$$

Assignment

$$\frac{?}{\vdash \{P\}\ x := E\ \{Q\}}$$

$$\frac{}{\vdash \{Q[x \to E]\}\ x := E\ \{Q\}}$$

Exercise

- Try out the assignment rule
- $\{?\}\ \ y \coloneqq 2 * x\ \{\, y > 10 \,\}$
- $\{?\}\ x \coloneqq x + 1\ \{\, x > 10 \,\}$

Concurrency

- Separation Logic