Axiomatic Semantics

Computer Science and Artificial Intelligence Laboratory
MIT

October 21, 2015

Motivation

Consider the following program

```
...
if(x > y){
  t = x - y;
  while(t > 0){
    x = x - 1;
    y = y + 1;
    t = t - 1;
  }
}
```

I claim that for any values of x and y

- the loop will terminate
- when it does, if x > y, the values of x and y will be swapped

How could I prove this?

Motivation

The tools we have seen so far are insufficient

- Operational semantics
  - easy to argue that a given input will produce a given output
  - also easy to argue that all constructs in the language will preserve some property (like when we proved type soundness)
  - much harder to prove general properties of the behavior of a program on all inputs
- Type-based reasoning
  - types allow us to design custom checkers to verify specific properties
  - very good at reasoning about properties of the data pointed at by particular variables.

Axiomatic Semantics

A system for proving properties about programs

Key idea:

- we can define the semantics of a construct by describing its effect on assertions about the program state

Two components

- A language for stating assertions
  - can be First Order Logic (FOL) or a specialized logic such as separation logic.
  - many specialized languages developed over the years
    - Z, Larch, JML, Spec#
- Deductive rules for establishing the truth of such assertions

A little history

Early years: Unbridled optimism

- Heavily endorsed by the likes of Hoare and Dijkstra
- If you can prove programs correct, bugs will be a thing of the past
    - you won't even have to test your programs

The middle ages

- 1979 paper by DeMillo, Lipton and Perllis
    - proofs in math only work because there is a social process in place to get people to argue them and internalize them
    - program proofs are too boring for social process to form around them
    - programs change too fast and proofs are too brittle

The renaissance

- New generation of automated reasoning tools
- A handful of success stories
- Better appreciation of costs, benefits and limitations?

The basics

```
{A} stmt {B}
```

Precondition   Postcondition

Hoare triple

- If the precondition holds before stmt and stmt terminates postcondition will hold afterwards

This is a partial correctness assertion

- we sometimes use the notation

```
[A] stmt [B]
```

to denote a total correctness assertion

  - that means you also have to prove termination

What do assertions mean?

We first need to introduce a language

For today we will be using Winskel's IMP

$e := n \mid x \mid e_1 + e_2 \mid e_1 = e_2$

$c := x := e \mid c_1 ; c_2 \mid \text{if } e \text{ then } c_1 \text{ else } c_2$
$\quad \mid \text{while } e \text{ do } c \mid \text{skip}$

Big Step Semantics have two kinds of judgments

expressions result in values

$$\langle e, \sigma \rangle \rightarrow n$$

commands change the state

$$\langle c, \sigma \rangle \rightarrow \sigma'$$

Semantics of IMP

Commands mutate the state

$$\frac{\langle e, \sigma\rangle \to e'}{\langle X := e, \sigma\rangle \to \sigma[X \to e']}$$

$$\frac{\langle c_1, \sigma\rangle \to \sigma'' \qquad \langle c_2, \sigma''\rangle \to \sigma'}{\langle c_1; c_2, \sigma\rangle \to \sigma'}$$

$$\frac{\langle e_1, \sigma\rangle \to false \quad \langle c_f, \sigma\rangle \to \sigma'}{\langle if\ e_1 then\ c_t\ else\ c_f, \sigma\rangle \to \sigma'}$$

$$\frac{\langle e_1, \sigma\rangle \to true \quad \langle c_t, \sigma\rangle \to \sigma'}{\langle if\ e_1 then\ c_t\ else\ c_f, \sigma\rangle \to \sigma'}$$

What about loops?

Semantics of IMP

The definition for loops must be recursive

$$\frac{\langle e_1, \sigma\rangle \to false}{\langle while\ e_1\ then\ c\ \ , \sigma\rangle\ \to \sigma}$$

$$\frac{\langle e_1, \sigma\rangle \to true \quad \langle c; while\ e_1\ then\ c, \sigma\rangle \to \sigma'}{\langle while\ e_1\ then\ c\ \ , \sigma\rangle\ \to \sigma'}$$

$$\frac{\langle e_1, \sigma\rangle \to true \quad \langle c, \sigma\rangle \to \sigma'' \quad \langle while\ e_1\ then\ c, \sigma''\rangle \to \sigma'}{\langle while\ e_1\ then\ c\ \ , \sigma\rangle\ \to \sigma'}$$

What do assertions mean?

The language of assertions

- A := true | false | e1 = e2 | e1 >= e2 | A1 and A2 |
  not A | $\forall$x . A

Notation $\sigma \vDash A$ means that the assertion holds on state σ

- This is defined inductively over the structure of A.
- Ex. $\sigma \vDash A \; and \; B \quad iff \quad \sigma \vDash A \; and \; \sigma \vDash B$

What do assertions mean

Complete list

- $\sigma \vDash true \quad \sigma \nvDash false$

- $\dfrac{\langle e_1,\sigma\rangle \to v \quad \langle e_2,\sigma\rangle \to v}{\sigma \vDash e_1 = e_2} \qquad \dfrac{\langle e_1,\sigma\rangle \to v_1 \quad \langle e_2,\sigma\rangle \to v_2 \quad v_1 \le v_2}{\sigma \vDash e_1 \le e_2}$

-

- $\dfrac{\langle e_1,\sigma\rangle \to v_1 \quad \langle e_2,\sigma\rangle \to v_2 \quad v_1 \ne v_2}{\sigma \nvDash e_1 = e_2} \qquad \dfrac{\langle e_1,\sigma\rangle \to v_1 \quad \langle e_2,\sigma\rangle \to v_2 \quad v_1 > v_2}{\sigma \nvDash e_1 \le e_2}$

- $\dfrac{\sigma \vDash A \quad \sigma \vDash B}{\sigma \vDash A \; and \; B} \qquad \dfrac{\forall\, v.\ \sigma[x \to v] \vDash A}{\sigma \vDash \forall x.A} \qquad \dfrac{\sigma \vDash A \quad \sigma \vDash B}{\sigma \vDash A \; and \; B} \qquad \dfrac{\sigma \nvDash A}{\sigma \nvDash A \; and \; B} \qquad \dfrac{\sigma \nvDash B}{\sigma \nvDash A \; and \; B} \qquad \dfrac{\exists\, v.\ \sigma[x \to v] \nvDash A}{\sigma \nvDash \forall x.A}$

- $\dfrac{\sigma \nvDash A}{\sigma \vDash not\ A} \qquad \dfrac{\sigma \vDash A}{\sigma \nvDash not\ A}$

Partial correctness

Partial Correctness can then be defined in terms of OS

{A} c {B} iff

$$\forall\sigma\forall\sigma'(\sigma \vDash A \ \wedge \ \langle c,\sigma\rangle \rightarrow \sigma') \Rightarrow \sigma' \vDash B$$

Defining axiomatic semantics

Establishing the truth of a Hoare triple in terms of the operational semantics is impractical

The real power of AS is the ability to establish the validity of a Hoare triple by using deduction rules

- $\vdash \{A\} c \{B\}$ means we can deduce the triple from a set of basic axioms

Derivation Rules

Derivation rules for each language construct

$$\frac{}{\vdash \{A[x \to e]\}x := e\ \{A\}}$$

$$\frac{\vdash \{A \wedge b\}c_1\ \{B\} \qquad \vdash \{A \wedge not\ b\}c_2\ \{B\}}{\vdash \{A\}if\ b\ then\ c_1 else\ c_2\ \{B\}}$$

$$\frac{\vdash \{A \wedge b\}c\ \{A\}}{\vdash \{A\}while\ b\ do\ c\ \{A \wedge not\ b\}}$$

$$\frac{\vdash \{A\}c_1\ \{C\} \qquad \vdash \{C\}c_2\ \{B\}}{\vdash \{A\}c_1 ; c_2\ \{B\}}$$

Can be combined together with the rule of consequence

$$\frac{\vdash A' \Rightarrow A \vdash \{A\}c\ \{B\} \vdash B \Rightarrow B'}{\vdash \{A'\}c\ \{B'\}}$$

Soundness and Completeness

What does it mean for our deduction rules to be sound?

- You will never be able to prove anything that is not true
- truth is defined in terms of our original definition of {A} c {B}

$$\forall\sigma\forall\sigma'(\sigma \vDash A \wedge \langle c,\sigma\rangle \rightarrow \sigma') \Rightarrow \sigma' \vDash B$$

- we can prove this, but it's tricky

What does it mean for them to be complete?

- If a statement is true, we should be able to prove it via deduction

So are they complete?

- yes and no
    - They are complete relative to the logic
    - but there are no complete and consistent logics for elementary arithmetic (Gödel)

Completeness Argument

$$\forall\sigma\forall\sigma'(\sigma \vDash A \wedge \langle c,\sigma\rangle \to \sigma') \Rightarrow \sigma' \vDash B$$

$$\Rightarrow$$

$$\vdash \{A\}c\ \{B\}$$

Prove by induction on the structure of the derivation of $\langle c,\sigma\rangle \to \sigma'$

- Look at all the different ways of proving that $\langle c,\sigma\rangle \to \sigma'$
- Make sure that for each of those, I can prove $\vdash \{A\}c\{B\}$

Completeness: Base case

$$\frac{\langle e, \sigma \rangle \to e'}{\langle X := e, \sigma \rangle \to \sigma[X \to e']}$$

`Need to prove:` $(\sigma \vDash A \;\wedge\; \sigma[X \to e'] \vDash B) \;\Rightarrow\, \vdash \{A\} X \coloneqq e \,\{B\}$

I only have one rule to prove $\vdash \{A\} X \coloneqq e \,\{B\}$

$$\frac{}{\vdash \{A[x \to e]\} x := e \,\{A\}}$$

- (well, that plus the rule of consequence).

So I need to show that

- $(\sigma \vDash A \;\wedge\; \sigma[X \to e'] \vDash B) \Rightarrow (A \Rightarrow B[x \to e])$
- Equivalently $\forall\, \sigma. (\sigma \vDash A \;\wedge\; \sigma[X \to e'] \vDash B) \Rightarrow (\sigma \vDash B[x \to e])$

Completeness: An inductive case

$$\frac{\langle c_1, \sigma \rangle \to \sigma'' \qquad \langle c_2, \sigma'' \rangle \to \sigma'}{\langle c_1; c_2, \sigma \rangle \to \sigma'}$$

Need to prove: $(\sigma \vDash A \ \wedge \ \sigma' \vDash B) \ \Rightarrow \vdash \{A\} c_1; c_2 \ \{B\}$

Assuming $(\sigma \vDash A \wedge \sigma'' \vDash C) \wedge \vdash \{A\} c_1 \{C\}$ and $(\sigma'' \vDash C \wedge \sigma' \vDash B) \wedge \vdash \{C\} c_1 \{B\}$

6.820 Fundamentals of Program Analysis
Fall 2015