Introduction to Semantics

Dr. Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Objectives

- Define *judgment* and explain its purpose in programming languages.
- Use *proof rules* to define judgments inductively.
- Use *proof trees* to prove properties about complex syntactic objects.

This presentation draws from Robert Harper's first chapters in [Har12].

Judgments

- A *judgment* is an assertion about a syntactic object.
- Examples:

Judgments

- A *judgment* is an assertion about a syntactic object.
- Examples:
  - 3 is odd

Judgments

- A *judgment* is an assertion about a syntactic object.
- Examples:
  - 3 is odd
  - $2 + 3 \Downarrow 5$

Judgments

- A *judgment* is an assertion about a syntactic object.
- Examples:
  - 3 is odd
  - $2 + 3 \Downarrow 5$
  - $\vdash 2.4 > 3.5 : \texttt{Bool}$

The Parts of a Rule

- We can also define judgments inductively.
- Let $J, J_1, J_2, \ldots J_n$ be a set of judgments.
- Then we can have a *rule* as follows:

$$\frac{J_1 \qquad J_2 \qquad \cdots \qquad J_n}{J}\ \textsc{Label}$$

The Parts of a Rule

- We can also define judgments inductively.
- Let $J, J_1, J_2, \ldots J_n$ be a set of judgments.
- Then we can have a *rule* as follows:

$$\frac{J_1 \qquad J_2 \qquad \cdots \qquad J_n}{J}\ \textsc{Label}$$

- The $J_1 \ldots J_n$ are called *assumptions* or *premises*.
- J is called a *conclusion*.

Axioms

- It's possible for there to be no assumptions!
- Such a rule is called an *axiom*.

$$\frac{}{J}\ \textsc{Label}$$

Side Conditions

- If a premise is not a judgment, we sometimes write it as a *side condition*.

$$\frac{}{x \text{ is even}}\ \textsc{Mod0}, x \mod 2 = 0$$

Example: Even and Odd Numbers with Addition

$$\frac{}{x \text{ is even}}\ \textsc{Mod0},\ x \mod 2 = 0 \qquad \frac{}{x \text{ is odd}}\ \textsc{Mod1},\ x \mod 2 = 1$$

$$\frac{x \text{ is even} \quad y \text{ is even}}{x + y \text{ is even}}\ \textsc{Even+Even} \qquad \frac{x \text{ is odd} \quad y \text{ is odd}}{x + y \text{ is even}}\ \textsc{Odd+Odd}$$

$$\frac{x \text{ is even} \quad y \text{ is odd}}{x + y \text{ is odd}}\ \textsc{Even+Odd} \qquad \frac{x \text{ is odd} \quad y \text{ is even}}{x + y \text{ is odd}}\ \textsc{Odd+Even}$$

Example: Even and Odd Numbers with Multiplication

$$\frac{x \text{ is even} \qquad y \text{ is even}}{x \times y \text{ is even}}\ \textsc{Even}\times\textsc{Even} \qquad \frac{x \text{ is odd} \qquad y \text{ is odd}}{x \times y \text{ is odd}}\ \textsc{Odd}\times\textsc{Odd}$$

$$\frac{x \text{ is even} \qquad y \text{ is odd}}{x \times y \text{ is even}}\ \textsc{Even}\times\textsc{Odd} \qquad \frac{x \text{ is odd} \qquad y \text{ is even}}{x \times y \text{ is even}}\ \textsc{Odd}\times\textsc{Even}$$

Building Proof Trees

- We can use these rules to prove judgments about objects inductively.

$$\dfrac{\dfrac{}{4 \text{ is even}}\ \textsc{Mod0}, 4 \mod 2 = 0 \qquad \dfrac{}{7 \text{ is odd}}\ \textsc{Mod1}, 7 \mod 2 = 1}{4 + 7 \text{ is odd}}\ \textsc{Even+Odd}$$

- There are two ways you can use proof trees.
  - Prove a property you already know.
  - Infer a property you don't already know.

Using Proof Trees to Prove

How to use it:

- Start with the judgment you want to prove.

$$4 + 7 \text{ is odd}$$

Using Proof Trees to Prove

How to use it:

- Start with the judgment you want to prove.
- Decide which rule applies.

$$\frac{}{4 + 7 \text{ is odd}}\ \textsc{Even+Odd}$$

Using Proof Trees to Prove

How to use it:

- Start with the judgment you want to prove.
- Decide which rule applies.
- Recursively prove first subexpression.

$$\dfrac{\dfrac{}{4 \text{ is even}}\ \textsc{Mod0},\quad 4 \mod 2 = 0}{4 + 7 \text{ is odd}}\ \textsc{Even+Odd}$$

Using Proof Trees to Prove

How to use it:

- Start with the judgment you want to prove.
- Decide which rule applies.
- Recursively prove first subexpression.
- Recursively prove second subexpression.

$$\dfrac{\dfrac{}{4 \text{ is even}}\ \textsc{Mod}0, 4 \mod 2 = 0 \qquad \dfrac{}{7 \text{ is odd}}\ \textsc{Mod}1, 7 \mod 2 = 1}{4 + 7 \text{ is odd}}\ \textsc{Even+Odd}$$

References

[Har12] Robert Harper. *Practical Foundations for Programming Languages*. 2012, p. 496. DOI: 10.1017/CBO9781139342131.