Introduction

Arnd Poetzsch-Heffter

Software Technology Group
Fachbereich Informatik
Technische Universität Kaiserslautern

Sommersemester 2010
Outline

1 Introduction
   • Overview

2 Language: Syntax and Semantics
   • Syntax
   • Semantics

3 Proof Systems/Logical Calculi
   • Introduction
   • Hilbert Calculus
   • Natural Deduction

4 Summary
Overview

Motivation
- Specifications: Models and properties
- How do we express/specify facts?
- What is a proof? What is a formal proof?
- How do we prove a specified fact?
- Why formal? What is the role of a theorem prover?

Goals
- recapitulate logic
- introduce/review basic concepts
Outline

1 Introduction
   - Overview

2 Language: Syntax and Semantics
   - Syntax
   - Semantics

3 Proof Systems/Logical Calculi
   - Introduction
   - Hilbert Calculus
   - Natural Deduction

4 Summary
Aspects of syntax

- used to designate things and express facts
- terms and formulas are formed from variables and function symbols
- function symbols map a tuple of terms to another term
- constant symbols: no arguments
- constant can be seen as functions with zero arguments
- predicate symbols are considered as boolean functions
- set of variables

Example (Natural Numbers)

- constant symbol: 0
- function symbol suc : \(\mathbb{N} \rightarrow \mathbb{N}\)
Example (Symbols)

- $\mathcal{V} = \{a, b, c, \ldots\}$ is a set of propositional variables
- two function symbols: $\neg$ and $\rightarrow$

Example (Language)

- each $P \in \mathcal{V}$ is a formula
- if $\phi$ is a formula, then $\neg \phi$ is a formula
- if $\phi$ and $\psi$ are formulas, then $\phi \rightarrow \psi$ is a formula
Syntax and Semantics

Syntactic Sugar

Purpose

- additions to the language that do not affect its expressiveness
- more practical way of description

Example

Abbreviations in Propositional Logic

- True denotes $\phi \implies \phi$
- False denotes $\neg True$
- $\phi \lor \psi$ denotes $(\neg \phi) \implies \psi$
- $\phi \land \psi$ denotes $\neg((\neg \phi) \lor (\neg \psi))$
- $\phi \leftrightarrow \psi$ denotes $((\phi \implies \psi) \land (\psi \implies \phi))$
### Purpose
- syntax only specifies the structure of terms and formulas
- symbols and terms are assigned a meaning
- variables are assigned a value
- in particular, propositional variables are assigned a truth value

### Bottom-Up Approach
- assignments give variables a value
- terms/formulas are evaluated based on the meaning of the function symbols
Interpretations/Structures

Example (Assignment in Propositional Logic)

A variable assignment in propositional logic is a mapping

- \( I : \mathcal{V} \rightarrow \{ \text{true}, \text{false} \} \)

Example (Denotation of Propositional Logic)

The denotation \( V \) takes an assignment \( I \) and a formula and yields a true or false:

- if \( \phi \in \mathcal{V} \): \( V(\phi) = I(\phi) \)
- \( V(\neg \phi) = f_\neg(V(\phi)) \)
- \( V(\phi \rightarrow \psi) = f_\rightarrow(V(\phi), V(\psi)) \)

where

\[
\begin{array}{|c|c|}
\hline
f_\neg & \text{false} & \text{true} \\
\hline
\text{false} & \text{true} & \text{false} \\
\text{true} & \text{false} & \text{true} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
f_\rightarrow & \text{false} & \text{true} \\
\hline
\text{false} & \text{true} & \text{true} \\
\text{true} & \text{false} & \text{true} \\
\hline
\end{array}
\]
Validity

Definition (Validity of formulas in propositional logic)
- A formula $\phi$ is valid w.r.t. an assignment $I$ if $V/I\phi$ evaluates to true for all assignments $I$.
- Notation: $\models \phi$

Example (Tautology in Propositional Logic)
- $\phi = a \lor \neg a$ (where $a \in \mathcal{V}$) is valid
  - $I(a) = \text{false}: V(a \lor \neg a) = \text{true}$
  - $I(a) = \text{true}: V(a \lor \neg a) = \text{true}$
Outline

1. Introduction
   - Overview

2. Language: Syntax and Semantics
   - Syntax
   - Semantics

3. Proof Systems/Logical Calculi
   - Introduction
   - Hilbert Calculus
   - Natural Deduction

4. Summary
Introduction

General Concept

- purely syntactical manipulations based on designated transformation rules
- starting point: set of formulas, often a given set of axioms
- deriving new formulas by deduction rules from given formulas $\Gamma$
- $\phi$ is provable from $\Gamma$ if $\phi$ can be obtained by a finite number of derivation steps assuming the formulas in $\Gamma$
- notation: $\Gamma \vdash \phi$ means $\phi$ is provable from $\Gamma$
- notation: $\vdash \phi$ means $\phi$ is provable from a given set of axioms
### Proof System Styles

<table>
<thead>
<tr>
<th>Proof System Style</th>
<th>Easy to Understand</th>
<th>Hard to Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hilbert Style</strong></td>
<td>easy to understand</td>
<td>hard to use</td>
</tr>
<tr>
<td><strong>Natural Deduction</strong></td>
<td>easy to use</td>
<td>hard to understand</td>
</tr>
</tbody>
</table>

...
Hilbert-Style Deduction Rules

**Definition (Deduction Rule)**
- deduction rule $d$ is a $n + 1$-tuple

\[
\frac{\phi_1 \ldots \phi_n}{\psi}
\]

- formulas $\phi_1 \ldots \phi_n$, called premises of rule
- formula $\psi$, called conclusion of rule
Hilbert-Style Proofs

Definition (Proof)

- let $D$ be a set of deduction rules, including the axioms as rules without premisses
- proofs in $D$ are (natural) trees such that
  - axioms are proofs
  - if $P_1, \ldots, P_n$ are proofs with roots $\phi_1 \ldots \phi_n$ and $\phi_1 \cdots \phi_n \psi$ is in $D$, then $P_1 \cdots P_n \psi$ is a proof in $D$
- can also be written in a line-oriented style
## Hilbert-Style Deduction Rules

<table>
<thead>
<tr>
<th>Axioms</th>
</tr>
</thead>
<tbody>
<tr>
<td>let $\Gamma$ be a set of axioms, $\psi \in \Gamma$, then $\overline{\psi}$ is a proof</td>
</tr>
<tr>
<td>axioms allow to construct trivial proofs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modus Ponens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule example: $\frac{\phi \rightarrow \psi}{\phi}$</td>
</tr>
<tr>
<td>if $\phi \rightarrow \psi$ and $\phi$ have already been proven, $\psi$ can be deduced</td>
</tr>
</tbody>
</table>
Example (Hilbert Proof)

- language formed with the four proposition symbols $P, Q, R, S$
- axioms: $P, Q, Q \rightarrow R, P \rightarrow (R \rightarrow S)$

```
\begin{align*}
P \rightarrow (R \rightarrow S) & \quad P \\
\hline
R \rightarrow S & \\
\hline
\end{align*}
```

```
\begin{align*}
Q \rightarrow R & \quad Q \\
\hline
R & \\
\hline
S & \\
\hline
\end{align*}
```
Definition (Axioms of Propositional Logic)

All instantiations of the following schemas:

- \( A \rightarrow (B \rightarrow A) \)
- \( (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)) \)
- \( (\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B) \)
- Where \( A, B, C \) are arbitrary propositions
Motivation

- Introducing a hypothesis is a natural step in a proof.
- Hilbert proofs do not permit this directly.
- Can be only encoded by using $\rightarrow$.
- Proofs are much longer and not very natural.

Natural Deduction

- Alternative definition where introduction of a hypothesis is a deduction rule.
- Deduction step can modify not only the proven propositions but also the assumptions $\Gamma$. 

Natural Deduction Rules

Definition (Natural Deduction Rule)

- deduction rule $d$ is a $n+1$-tuple
  $$\frac{}{
\Gamma_1 \vdash \phi_1 \quad \cdots \quad \Gamma_n \vdash \phi_n
\}
\Gamma \vdash \psi$$
- pairs of $\Gamma$ (set of formulas) and $\phi$ (formulas): sequents
- proof: tree of sequents with rule instantiations as nodes
Natural Deduction Rules

- rich set of rules
- *elimination rules* eliminate a logical symbol from a premise
- *introduction rules* introduce a logical symbol into the conclusion
- reasoning from assumptions
### Definition (Natural Deduction Rules for Propositional Logic)

#### \( \lor \)-introduction

\[
\begin{array}{c}
\Gamma \vdash \phi \\
\hline
\Gamma \vdash \phi \lor \psi
\end{array}
\quad
\begin{array}{c}
\Gamma \vdash \psi \\
\hline
\Gamma \vdash \phi \lor \psi
\end{array}
\]

#### \( \lor \)-elimination

\[
\begin{array}{c}
\Gamma \vdash \phi \lor \psi \\
\Gamma, \phi \vdash \xi \\
\hline
\Gamma \vdash \xi
\end{array}
\quad
\begin{array}{c}
\Gamma, \psi \vdash \xi \\
\hline
\Gamma \vdash \xi
\end{array}
\]

#### \( \to \)-introduction

\[
\begin{array}{c}
\Gamma, \phi \vdash \psi \\
\hline
\Gamma \vdash \phi \to \psi
\end{array}
\]

#### \( \to \)-elimination

\[
\begin{array}{c}
\Gamma \vdash \phi \to \psi \\
\Gamma \vdash \phi \\
\hline
\Gamma \vdash \psi
\end{array}
\]
Outline

1. Introduction
   - Overview

2. Language: Syntax and Semantics
   - Syntax
   - Semantics

3. Proof Systems/Logical Calculi
   - Introduction
   - Hilbert Calculus
   - Natural Deduction

4. Summary
Theorem-Proving Fundamentals

- syntax: symbols, terms, formulas
- semantics: (mathimacal structures,) variable assignments, denotations for terms and formulas
- proof system/(logical) calculus: axioms, deduction rules, proofs, theories

Fundamental Principle of Logic: “Establish truth by calculation” (APH, 2010)