Logic
Part I: Classical Logic and Its Semantics

Max Schäfer

What Is Logic?

- this course is about *formal logic*
- investigate principles of reasoning, independently of particular language, mindset, or philosophy
- based on a *formal language*, precise *deductive rules*
- resulting systems can be used to formalize mathematics or CS
- can also be studied in its own right

What to Expect

- what we will cover:
  - classical and intuitionistic logic, propositional, first order, and second order
  - their semantics and deduction systems
  - connections with programming language research
- what we will not cover:
  - using logic to win arguments with your Significant Other
  - role of logic in artificial intelligence
  - using logic in digital hardware design, automated verification, and much more

Principles of Classical Logic

- classical logic aims to model *valid reasoning*
- logical formulas represent statements that are either true or false
- proving a formula means showing that it is true

- sometimes this is easy

$$\sqrt{2} \notin \mathbb{Q}$$

- sometimes it is hard

$$\forall n.n > 2 \rightarrow \neg(\exists a, b, c.a^n + b^n = c^n)$$

- proving a formula does not "make" it true, it just *demonstrates* its truth

1 Propositional Logic

Propositional Logic

- deals with atomic *propositions* and their combinations
- e.g. consider propositions "it is raining" and "the grass is wet"
- assume "it is raining" is true and "the grass is wet" is true, then "it is raining **and** the grass is wet" is true
- assume "it is raining" implies "the grass is wet", then
  - if "it is raining" is true, then "the grass is wet" is true
  - if "the grass is wet" is false, then "it is raining" is false
  - if "the grass is wet" is true, then "it is raining" *might* (but need not) be true
- either "it is raining" is true or "it is raining" is false

Atomicity of Propositions

- we do not need to know what the propositions mean
- they could be expressed in German, for example:
  - assume "es regnet" is true and "das Gras ist nass" is true, then "es regnet **and** das Gras ist nass" is true
  - assume "es regnet" implies "das Gras ist nass", then
    - if "es regnet" is true, then "das Gras ist nass" is true
    - if "das Gras ist nass" is false, then "es regnet" is false
    - if "das Gras ist nass" is true, then "es regnet" might be true
  - either "es regnet" is true or "es regnet" is false
- observe use of **and**: not a proposition, but a *connective*

Language Independence

- we do not want to depend on syntax or grammar of some natural language
- to achieve this, propositions will be represented by capital letters $P, Q, R, \ldots$
- logical connectives will be expressed by symbols like $\wedge$, $\vee$, etc.

The Formal Language of Propositional Logic

- assume we have an alphabet $\mathcal{R}$ of *propositional letters*, denoted by P, Q, R, ...
- the set $\mathtt{PF}_{\mathcal{R}}$ of propositional formulas over $\mathcal{R}$ is defined inductively:
  1. every propositional letter is a formula
  2. the symbol $\bot$ is a formula (*falsity*)
  3. if φ and ψ are formulas, then so are
     (a) $\varphi \wedge \psi$ (*conjunction*)
     (b) $\varphi \vee \psi$ (*disjunction*)
     (c) $\varphi \to \psi$ (*implication*)

Intuitive Meaning of Propositional Logic

formula	**intuitive reading**	**is true if...**
P	P	proposition P is true
$\bot$	false	never true
$P \wedge Q$	P and Q	proposition P is true, and also proposition Q is true
$P \vee Q$	P or Q	proposition P is true, or proposition Q is true, or both are true
$P \to Q$	if P then Q	it is not the case that P is true and Q is false

Defined Connectives and Syntactic Equality

- other connectives can be defined in terms of the basic ones:
  - $\neg\varphi := \varphi \to \bot$ (*negation*)
    $\neg\varphi$ is true if φ is false
  - $\varphi \leftrightarrow \psi := (\varphi \to \psi) \wedge (\psi \to \varphi)$ (*equivalence*)
    $\varphi \leftrightarrow \psi$ is true if both φ and ψ are true, or both are false
  - $\top := \bot \to \bot$ (*truth*)
    $\top$ is always true
- $\neg$ and $\leftrightarrow$ are not true connectives, but only abbreviations; e.g., $\neg P \equiv P \to \bot$ (the *same* formula)
- adding parentheses does not make a difference: $((P \vee Q)) \equiv P \vee Q$
- however, we sometimes need parentheses to disambiguate

Precedence and Associativity

- precedence of connectives (lowest to highest) and associativity:

connective	**associativity**	**example**
$\leftrightarrow$	left	$P \leftrightarrow Q \leftrightarrow R \equiv (P \leftrightarrow Q) \leftrightarrow R$
$\rightarrow$	right	$P \rightarrow Q \rightarrow R \equiv P \rightarrow (Q \rightarrow R)$
$\vee$	left	$P \vee Q \vee R \equiv (P \vee Q) \vee R$
$\wedge$	left	$P \wedge Q \wedge R \equiv (P \wedge Q) \wedge R$
$\neg$		

- example:

$$P \vee Q \wedge P \rightarrow P \leftrightarrow \neg\neg P \vee \neg P$$

$$\equiv$$

$$((P \vee (Q \wedge P)) \rightarrow P) \leftrightarrow ((\neg\neg P) \vee (\neg P))$$

Example Formulas

- $P \rightarrow P$ is true, no matter if P is

  same for $P \rightarrow (Q \rightarrow P)$ and $P \vee \neg P$; they are *valid* (*tautologies*)

- $P \wedge Q$ may be true or false

  same for $P \rightarrow \neg P$ and $P \vee P$

- $P \wedge \neg P$ is false, no matter if P is

  same for $P \leftrightarrow \neg P$ and $\top \rightarrow \bot$

- but what about $(P \rightarrow Q \rightarrow R) \rightarrow (P \rightarrow Q) \rightarrow P \rightarrow R$? or $((P \rightarrow Q) \rightarrow P) \rightarrow P$? or $\neg(P \wedge Q) \leftrightarrow \neg P \vee \neg Q$?

Truth Value Semantics

- in general, to know whether a formula φ is true, we need to know whether its propositional letters are true
- need a truth value assignment (interpretation) $I \colon \mathcal{R} \rightarrow \mathcal{B}$, where $\mathcal{B} := \{\mathtt{T}, \mathtt{F}\}$
- given an interpretation I, define the semantics $[\![\varphi]\!]_I$ of a formula φ:
  1. if φ is some $P \in \mathcal{R}$, then $[\![\varphi]\!]_I := I(P)$
  2. if φ is $\bot$, then $[\![\varphi]\!]_I := \mathtt{F}$
  3. if φ is $\varphi_1 \wedge \varphi_2$, then
     - if $[\![\varphi_1]\!]_I = \mathtt{T}$ and $[\![\varphi_2]\!]_I = \mathtt{T}$, then $[\![\varphi]\!]_I := \mathtt{T}$
     - otherwise, $[\![\varphi]\!]_I := \mathtt{F}$
  4. if φ is $\varphi_1 \vee \varphi_2$, then

- if $[\![\varphi_1]\!]_I = \mathtt{F}$ and $[\![\varphi_2]\!]_I = \mathtt{F}$, then $[\![\varphi]\!]_I := \mathtt{F}$
- otherwise, $[\![\varphi]\!]_I := \mathtt{T}$

5. if φ is $\varphi_1 \to \varphi_2$, then
   - if $[\![\varphi_1]\!]_I = \mathtt{T}$ and $[\![\varphi_2]\!]_I = \mathtt{F}$, then $[\![\varphi]\!]_I := \mathtt{F}$
   - otherwise, $[\![\varphi]\!]_I := \mathtt{T}$

Validity and Satisfiability

- for $\varphi \in \mathtt{PF}$ and interpretation I, $\models_I \varphi$ ("I is a model for φ") if $[\![\varphi]\!]_I = \mathtt{T}$
- if there is some I such that $\models_I \varphi$, φ is called *satisfiable*
- if $\models_I \varphi$ for all I, we write $\models \varphi$ and call φ *valid*
- for a set $\Gamma \subseteq \mathtt{PF}$, write $\models_I \Gamma$ to mean that $\models_I \varphi$ for every $\varphi \in \Gamma$
- $\Gamma \models \varphi$ ("Γ semantically entails φ"): for every I, if $\models_I \Gamma$, then $\models_I \varphi$

Example

- example: $\neg(P \wedge Q) \leftrightarrow \neg P \vee \neg Q$
  - let I be an interpretation
  - if $I(P) = \mathtt{F}$, then $[\![P]\!]_I = \mathtt{F}$, so $[\![P \wedge Q]\!]_I = \mathtt{F}$ and $[\![\neg(P \wedge Q)]\!]_I = \mathtt{T}$; also, $[\![\neg P]\!]_I = \mathtt{T}$, hence $[\![\neg P \vee \neg Q]\!]_I = \mathtt{T}$
  - if $I(P) = \mathtt{T}$ and $I(Q) = \mathtt{T}$, then $[\![P \wedge Q]\!]_I = \mathtt{T}$, so $[\![\neg(P \wedge Q)]\!]_I = \mathtt{F}$; also, $[\![\neg P]\!]_I = \mathtt{F}$ and $[\![\neg Q]\!]_I = \mathtt{F}$, hence $[\![\neg P \vee \neg Q]\!]_I = \mathtt{F}$
  - if $I(P) = \mathtt{T}$ and $I(Q) = \mathtt{F}$, then $[\![\neg(P \wedge Q)]\!]_I = \mathtt{T}$; also, $[\![\neg P \vee \neg Q]\!]_I = \mathtt{T}$

  For every I, $[\![\neg(P \wedge Q)]\!]_I = [\![\neg P \vee \neg Q]\!]_I$, hence $\models \neg(P \wedge Q) \leftrightarrow \neg P \vee \neg Q$.
- note: we only considered propositional letters P and Q; others are irrelevant

Propositional Letters in a Formula

- define set $\mathrm{PL}(\varphi)$ of propositional letters that occur in a formula φ:

1. if φ is $P \in \mathcal{R}$, then $\mathrm{PL}(\varphi) := \{P\}$
2. if φ is $\bot$, then $\mathrm{PL}(\varphi) := \emptyset$
3. if φ is of the form $\varphi_1 \wedge \varphi_2$ or $\varphi_1 \vee \varphi_2$ or $\varphi_1 \to \varphi_2$, then $\mathrm{PL}(\varphi) := \mathrm{PL}(\varphi_1) \cup \mathrm{PL}(\varphi_2)$

Lemma 1 (Agreement)**.** *For every formula φ and interpretations I_1, I_2 such that $I_1(P) = I_2(P)$ for every $P \in \mathrm{PL}(\varphi)$, we have*

$$[\![\varphi]\!]_{I_1} = [\![\varphi]\!]_{I_2}$$

Proof. Let us say that I_1 and I_2 *agree* on a set $A \subseteq \mathrm{PL}$ if $I_1(a) = I_2(a)$ for every $a \in A$. Thus, we assume that I_1 and I_2 agree on $\mathrm{PL}(\varphi)$, and we want to prove that $[\![\varphi]\!]_{I_1} = [\![\varphi]\!]_{I_2}$. We proceed by induction on the structure of φ.

1. If φ is $P \in \mathcal{R}$, then $\mathrm{PL}(\varphi) = \{P\}$, hence $I_1(P) = I_2(P)$, and
$$[\![\varphi]\!]_{I_1} = I_1(P) = I_2(P) = [\![\varphi]\!]_{I_2}$$
2. If φ is $\bot$, then
$$[\![\varphi]\!]_{I_1} = \mathtt{F} = [\![\varphi]\!]_{I_2}$$
3. If φ is $\varphi_1 \wedge \varphi_2$, then $\mathrm{PL}(\varphi) = \mathrm{PL}(\varphi_1) \cup \mathrm{PL}(\varphi_2)$. Since I_1 and I_2 agree on $\mathrm{PL}(\varphi)$, they also agree on $\mathrm{PL}(\varphi_1)$ and $\mathrm{PL}(\varphi_2)$ (which both are subsets of $\mathrm{PL}(\varphi)$). By induction hypothesis, we can thus assume that
$$(*) \qquad [\![\varphi_1]\!]_{I_1} = [\![\varphi_1]\!]_{I_2}$$
and
$$(**) \qquad [\![\varphi_2]\!]_{I_1} = [\![\varphi_2]\!]_{I_2},$$
because both are structurally smaller than φ.

   Now we want to determine $[\![\varphi]\!]_{I_1}$. There are two situations:

   (a) $[\![\varphi_1]\!]_{I_1} = \mathtt{T} = [\![\varphi_2]\!]_{I_1}$, hence $[\![\varphi]\!]_{I_1} = \mathtt{T}$; but then by $(*)$ and $(**)$ we also have $[\![\varphi_1]\!]_{I_2} = \mathtt{T} = [\![\varphi_2]\!]_{I_2}$, which gives us $[\![\varphi]\!]_{I_2} = \mathtt{T}$.

   (b) either $[\![\varphi_1]\!]_{I_1} = \mathtt{F}$ or $[\![\varphi_2]\!]_{I_1} = \mathtt{F}$ (or both), hence $[\![\varphi]\!]_{I_1} = \mathtt{F}$; but then by $(*)$ and $(**)$ either $[\![\varphi_1]\!]_{I_2} = \mathtt{F}$ or $[\![\varphi_2]\!]_{I_2} = \mathtt{F}$ (or both), so $[\![\varphi]\!]_{I_2} = \mathtt{F}$ as well.

   We have thus shown that also in this case
$$[\![\varphi]\!]_{I_1} = [\![\varphi]\!]_{I_2}$$
4. The cases that φ is $\varphi_1 \vee \varphi_2$ or $\varphi_1 \to \varphi_2$ are handled similar.

In conclusion, we have shown for every $\varphi \in \mathtt{PF}$ that
$$[\![\varphi]\!]_{I_1} = [\![\varphi]\!]_{I_2}$$
for any two interpretations I_1 and I_2 that agree on $\mathrm{PL}(\varphi)$. □

Truth Tabling

- for every formula φ, $\mathrm{PL}(\varphi)$ is finite, say $|\mathrm{PL}(\varphi)| = n$
- every one of these n variables could be either true or false; this gives 2^n combinations
- to know whether φ is valid, we only need to try them all out!

Examples

P	Q	$P \wedge Q$	$\neg(P \wedge Q)$	$\neg P$	$\neg Q$	$\neg P \vee \neg Q$	$\neg(P \wedge Q) \leftrightarrow \neg P \vee \neg Q$
F	F	F	T	T	T	T	T
F	T	F	T	T	F	T	T
T	F	F	T	F	T	T	T
T	T	T	F	F	F	F	T

P	Q	$P \to Q$	$(P \to Q) \to P$	$((P \to Q) \to P) \to P$
F	F	T	F	T
F	T	T	F	T
T	F	F	T	T
T	T	T	T	T

Some Tautologies

To get some practice in using truth tables, you can show that

- $\models P \wedge Q \to R \leftrightarrow P \to Q \to R$
- $\models P \wedge Q \to P$, $\models P \wedge Q \to Q$
- $\models P \to P \vee Q$, $\models Q \to P \vee Q$
- $\models (P \vee Q) \wedge (P \to R) \wedge (Q \to R) \to R$
- $\models P \wedge (P \to Q) \to Q$
- $\models \bot \to P$

Examples II

Truth tables can be used to find counter-examples:

P	Q	$P \to Q$	$(P \to Q) \to Q$	$((P \to Q) \to Q) \to P$
F	F	T	F	T
F	T	T	T	F
T	F	F	T	T
T	T	T	T	T

They can be used to see whether two formulas are equivalent:

P	Q	$P \to Q$	$\neg P \to \neg Q$	$\neg Q \to \neg P$
F	F	T	T	T
F	T	T	F	T
T	F	F	T	F
T	T	T	T	T

Equivalence

- we say that φ and ψ are (semantically) equivalent and write $\varphi \Leftrightarrow \psi$ if, for any interpretation I, $[\![\varphi]\!]_I = [\![\psi]\!]_I$
- we have $\varphi \Leftrightarrow \psi$ iff $\models \varphi \leftrightarrow \psi$
- we have $\models \varphi$ iff $\varphi \Leftrightarrow \top$

Note
If $\varphi \equiv \psi$, then obviously $\varphi \Leftrightarrow \psi$, but *not necessarily* the other way around!

Lemma 2 ("$\Leftrightarrow$" is an equivalence relation)**.** *For formulas φ, χ, ψ, we always have*

- $\varphi \Leftrightarrow \varphi$
- $\varphi \Leftrightarrow \psi$ *iff* $\psi \Leftrightarrow \varphi$
- *if* $\varphi \Leftrightarrow \chi$ *and* $\chi \Leftrightarrow \psi$*, then* $\varphi \Leftrightarrow \psi$

Important Equivalences
For propositional letters P, Q, R, we have:

1. Associativity:
   - $P \wedge (Q \wedge R) \Leftrightarrow (P \wedge Q) \wedge R$
   - $P \vee (Q \vee R) \Leftrightarrow (P \vee Q) \vee R$
2. Commutativity:
   - $P \wedge Q \Leftrightarrow Q \wedge P$
   - $P \vee Q \Leftrightarrow Q \vee P$
3. Distributivity:
   - $P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$
   - $P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$
4. Absorption:
   - $P \wedge (P \vee Q) \Leftrightarrow P$
   - $P \vee (P \wedge Q) \Leftrightarrow P$
5. Complement:
   - $P \vee \neg P \Leftrightarrow \top$
   - $P \wedge \neg P \Leftrightarrow \bot$

Substitutions

- $\varphi[\psi/P]$: substituting a formula ψ for all occurrences of a propositional letter P in φ
- formal definition:
  1. if φ is $Q \in \mathcal{R}$, then
     (a) if P and Q are the same, then $\varphi[\psi/P] := \psi$
     (b) otherwise, $\varphi[\psi/P] := \varphi$
  2. if φ is $\bot$, then $\varphi[\psi/P] := \varphi$
  3. if φ is $\varphi_1 \wedge \varphi_2$, then $\varphi[\psi/P] := \varphi_1[\psi/P] \wedge \varphi_2[\psi/P]$
  4. if φ is $\varphi_1 \vee \varphi_2$, then $\varphi[\psi/P] := \varphi_1[\psi/P] \vee \varphi_2[\psi/P]$
  5. if φ is $\varphi_1 \to \varphi_2$, then $\varphi[\psi/P] := \varphi_1[\psi/P] \to \varphi_2[\psi/P]$
- for example: $(R \vee \neg R)[Q \to Q/R] \equiv ((Q \to Q) \vee \neg(Q \to Q))$

Substitution Lemmas

Lemma 3 (substitution preserves equivalence)**.** *If $\psi_1 \Leftrightarrow \psi_2$, then $\varphi[\psi_1/P] \Leftrightarrow \varphi[\psi_2/P]$ for any propositional letter P and formulas φ, ψ_1, ψ_2.*

Corollary 4 (substitution in tautologies)**.** *If $\models \varphi$, then $\models \varphi[\psi/P]$ for any propositional letter P and formulas φ, ψ.*

Hence the "Important Equivalences" hold for arbitrary formulas.

Algebraic Reasoning

We can use equivalences to reason about formulas (taking only the Important Equivalences as given):

- we have

$$\begin{array}{ll} & P \\ \Leftrightarrow & \{\text{by Absorption}\} \\ & P \wedge (P \vee \neg P) \\ \Leftrightarrow & \{\text{by Complement and Lemma 3}\} \\ & P \wedge \top \end{array}$$

- hence

$$\begin{array}{ll} & P \vee P \\ \Leftrightarrow & \{\text{see above and Lemma 3}\} \\ & P \vee (P \wedge \top) \\ \Leftrightarrow & \{\text{by Absorption}\} \\ & P \end{array}$$

Notice that we can use Lemma 3 to replace equivalent formulas inside another formula. In the example above, we deduced that

$$P \wedge (P \vee \neg P) \Leftrightarrow P \wedge \top$$

Why does this work? From the Important Equivalences, we know that $P \vee \neg P \Leftrightarrow \top$. Now consider the formula $P \wedge R$. We have

$$(P \wedge R)[P \vee \neg P/R] \equiv P \wedge (P \vee \neg P)$$

and

$$(P \wedge R)[\top/R] \equiv P \wedge \top.$$

So we can apply Lemma 3 with $P \wedge R$ for φ, $P \vee \neg P$ for ψ_1 and $\top$ for ψ_2, and we get

$$P \wedge (P \vee \neg P) \Leftrightarrow P \wedge \top$$

as claimed.

Further Equivalences

The following equivalences follow from the ones given before:

1. Idempotency:
   $P \vee P \Leftrightarrow P \Leftrightarrow P \wedge P$
2. Neutrality:
   - $P \vee \bot \Leftrightarrow P$
   - $P \wedge \top \Leftrightarrow P$
3. Boundedness:
   - $P \vee \top \Leftrightarrow \top$
   - $P \wedge \bot \Leftrightarrow \bot$
4. Switching:
   - $\neg\top \Leftrightarrow \bot$
   - $\neg\bot \Leftrightarrow \top$
5. De Morgan's Laws:
   - $\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$
   - $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$
6. Involution:
   $\neg\neg P \Leftrightarrow P$

Functionally Complete Sets

- for every propositional formula φ there is a formula $\varphi^\dagger$ which only uses $\to$ and $\bot$ such that $\varphi \Leftrightarrow \varphi^\dagger$:
  1. if φ is $P \in \mathcal{R}$, take $\varphi^\dagger := \varphi$
  2. if φ is $\bot$, take $\varphi^\dagger := \varphi$
  3. if φ is $\varphi_1 \to \varphi_2$, take $\varphi^\dagger := \varphi_1^\dagger \to \varphi_2^\dagger$
  4. if φ is $\varphi_1 \wedge \varphi_2$, take $\varphi^\dagger := \neg(\varphi_1^\dagger \to \neg\varphi_2^\dagger)$
  5. if φ is $\varphi_1 \vee \varphi_2$, take $\varphi^\dagger := \neg\varphi_1^\dagger \to \varphi_2^\dagger$

  for example, $(P \wedge (P \vee Q))^\dagger \equiv \neg(P \to \neg(\neg P \to Q))$
- thus, $\{\bot, \to\}$ is a *functionally complete set*
- other functionally complete sets: e.g., $\{\neg, \vee\}$, $\{\neg, \wedge\}$

2 First Order Logic

Motivation: First Order Logic

- in mathematics, we want to express propositions about individuals, e.g.

  For every n, if $n > 0$ then for all m we have $m + n > m$.
- in the example, the individuals are numbers, ranged over by variables n, m
- we use constants (like 0) and functions (like $+$, arity 2) to construct *terms*
- relations (like $>$, arity 2) can be used to form *atomic propositions* about terms
- atomic propositions are used to construct more complex propositions
- first order logic (FOL) formalizes such statements in an abstract setting

The Approach of First Order Logic (FOL)

- first order logic formalizes reasoning about statements that can refer to individuals through *individual variables*
- a fixed set of function symbols acts on the individuals
- a fixed set of relation symbols expresses predicates on the individuals
- more complex statements can be formed by connectives like $\wedge, \vee, \to, \neg$ and the quantifiers $\forall, \exists$
- first order logic is sufficient to formalize great parts of mathematics, for example arithmetic

The Language of FOL

- a first order *signature* $\Sigma = \langle \mathtt{F}, \mathcal{R} \rangle$ describes a language with
  - *function letters* $f \in \mathtt{F}$ with arity $\alpha(f) \in \mathbb{N}$
  - *relation letters* $r \in \mathcal{R}$ with arity $\alpha(r) \in \mathbb{N}$
- terms $\mathtt{T}(\Sigma, \mathcal{V})$ over Σ and a set $\mathcal{V}$ of *individual variables* are inductively defined:
  - $\mathcal{V} \subseteq \mathtt{T}(\Sigma, \mathcal{V})$
  - for $f \in \mathtt{F}$ of arity n, $t_1, \ldots, t_n \in \mathtt{T}(\Sigma, \mathcal{V})$, also $f(t_1, \ldots, t_n) \in \mathtt{T}(\Sigma, \mathcal{V})$
- for a 0-ary constant d, we write $d()$ simply as d

Example

Signature $\Sigma_{\mathrm{ar}} = \langle \mathtt{F}_{\mathrm{ar}}, \mathcal{R}_{\mathrm{ar}} \rangle$ of arithmetic:

- $\mathtt{F}_{\mathrm{ar}} = \{\mathbf{0}, s, +, \cdot\}$, where $\alpha(\mathbf{0}) = 0$, $\alpha(s) = 1$, $\alpha(+) = \alpha(\cdot) = 2$
- $\mathcal{R}_{\mathrm{ar}} = \{\approx\}$, where $\alpha(\approx) = 2$
- examples for terms from $\mathtt{T}(\Sigma_{\mathrm{ar}}, \{x, y\})$:
  $\mathbf{0}$, $s(\mathbf{0})$, $s(s(\mathbf{0}))$, $\ldots$, $s(x)$, $+(s(x), y)$, $s(+(x, y))$, $\ldots$
- but not $\mathbf{0}(\mathbf{0})$ or $+(s(\mathbf{0}))$
- $+(x, y)$ usually written $x + y$, but still $+(x, y) \equiv x + y$

The Language of FOL (II)

- an *atom* is of the form $r(t_1, \ldots, t_n)$, where $r \in \mathcal{R}$, $\alpha(r) = n$, $t_1, \ldots, t_n \in \mathtt{T}(\Sigma, \mathcal{V})$

  write just r if $\alpha(r) = 0$
- set $\mathtt{FOL}_{\Sigma,\mathcal{V}}$ of formulas is inductively defined:
  1. every atom is a formula
  2. if φ, ψ are formulas then
     (a) $\varphi \wedge \psi$ is a formula
     (b) $\varphi \vee \psi$ is a formula
     (c) $\varphi \to \psi$ is a formula
  3. if $x \in \mathcal{V}$ and φ is a formula, then
     (a) $\forall x.\varphi$ is a formula (*universal quantifier*)
     (b) $\exists x.\varphi$ is a formula (*existential quantifier*)
  4. $\bot$ is a formula

  The quantifiers $\forall$ and $\exists$ have the lowest precedence of all connectives.

Example

Taking Σ_{ar} and $\mathcal{V} = \{x, y, d, d'\}$, the following are atoms (again, we use infix notation):

- $x \approx y$
- $x + y \approx y + x$
- $s(s(\mathbf{0})) \cdot x \approx x + x$

And here are some formulas:

- $\neg(x \approx s(x))$
- $(\exists d.x + d \approx y) \to (\exists d'.s(x) + d' \approx y) \vee s(x) \approx y$
- $\forall x.x + x \approx x \cdot x$

Intuitive Semantics of the Quantifiers

- $\forall x.\varphi$ should be understood as "for all values of x, φ holds"
- $\exists x.\varphi$ should be understood as "there is a value of x such that φ holds"
- so the formula
$$\forall x.x \approx \mathbf{0} \vee \exists y.x \approx s(y)$$
could be understood as

  > every x is either equal to zero, or there exists a number y such that x is its successor

- **however**, this interpretation relies on an intuitive interpretation of the function symbols s and $\mathbf{0}$ and the relation symbol $\approx$; it is certainly not true for all interpretations of these symbols!

Free and Bound Variables

- an appearance of an individual variable is called *bound* if it is within the scope of a quantifier, otherwise it is *free*

  e.g. (free variables are **bold**):

$$\mathbf{x} \approx s(\mathbf{y}) \qquad \exists x.x \approx s(\mathbf{y}) \qquad \forall y.\exists x.x \approx s(y)$$

- the same variable can appear both free and bound:
$$(\forall x.R(x, \mathbf{z}) \to (\exists y.S(y, x))) \wedge T(\mathbf{x})$$
- a formula is called *closed* when no variable occurs free in it
- the names of bound variables only serve to connect them with their quantifier, one name is as good as another (details later)

The Set of Free Variables

- definition of the set of free variables:

  1. $\mathrm{FV}(x) = \{x\}$ for $x \in \mathcal{V}$
  2. $\mathrm{FV}(f(t_1, \ldots, t_n)) = \bigcup_{i \in \{1,\ldots,n\}} \mathrm{FV}(t_i)$
  3. $\mathrm{FV}(r(t_1, \ldots, t_n)) = \bigcup_{i \in \{1,\ldots,n\}} \mathrm{FV}(t_i)$
  4. $\mathrm{FV}(\bot) = \emptyset$
  5. $\mathrm{FV}(\varphi \wedge \psi) = \mathrm{FV}(\varphi \vee \psi) = \mathrm{FV}(\varphi \to \psi) = \mathrm{FV}(\varphi) \cup \mathrm{FV}(\psi)$
  6. $\mathrm{FV}(\forall x.\varphi) = \mathrm{FV}(\varphi) \setminus \{x\}$
  7. $\mathrm{FV}(\exists x.\varphi) = \mathrm{FV}(\varphi) \setminus \{x\}$

  For example:

- $\mathrm{FV}(x) = \{x\}$, $\mathrm{FV}(\mathbf{0}) = \emptyset$, $\mathrm{FV}(s(\mathbf{0})) = \emptyset$
- $\mathrm{FV}(x \approx \mathbf{0} \vee x \approx s(y)) = \{x, y\}$
- $\mathrm{FV}(x \approx \mathbf{0} \vee (\exists y.x \approx s(y))) = \{x\}$
- $\mathrm{FV}(\forall x.x \approx \mathbf{0} \vee (\exists y.x \approx s(y))) = \emptyset$

Substitution in Terms and Formulas

- the operation of substituting a term t for a variable x in a term s (written $s[t/x]$) is defined as follows:

  1. $y[t/x] = \begin{cases} t & \text{if } x \equiv y, \\ y & \text{otherwise} \end{cases}$
  2. $(f(t_1, \ldots, t_n))[t/x] = f(t_1[t/x], \ldots, t_n[t/x])$

- on formulas, the definition is

  1. $(r(t_1, \ldots, t_n))[t/x] = r(t_1[t/x], \ldots, t_n[t/x])$
  2. $\bot[t/x] = \bot$
  3. $(\varphi \circ \psi)[t/x] = (\varphi[t/x]) \circ (\psi[t/x])$, for $\circ \in \{\wedge, \vee, \to\}$
  4. $(Qy.\varphi)[t/x] = \begin{cases} Qy.\varphi & \text{if } x \equiv y, \\ Qy.(\varphi[t/x]) & \text{if } x \not\equiv y,\ y \notin \mathrm{FV}(t) \end{cases}$, $Q \in \{\forall, \exists\}$

Note that substitution on formulas is not always defined!

Example

- $x[s(\mathbf{0})/x] \equiv s(\mathbf{0})$, $y[s(\mathbf{0})/x] \equiv y$
- $(x \approx \mathbf{0} \vee x \approx s(y))[s(\mathbf{0})/x] \equiv s(\mathbf{0}) \approx \mathbf{0} \vee s(\mathbf{0}) \approx s(y)$
- $(x \approx \mathbf{0} \vee (\exists y.x \approx s(y)))[s(\mathbf{0})/x] \equiv s(\mathbf{0}) \approx \mathbf{0} \vee (\exists y.s(\mathbf{0}) \approx s(y))$
- $(x \approx \mathbf{0} \vee x \approx s(y))[s(y)/x] \equiv s(y) \approx \mathbf{0} \vee s(y) \approx s(y)$
- $(x \approx \mathbf{0} \vee (\exists y.x \approx s(y)))[s(y)/x]$ is not defined
- $(\forall x.x \approx \mathbf{0} \vee (\exists y.x \approx s(y)))[s(\mathbf{0})/x] \equiv (\forall x.x \approx \mathbf{0} \vee (\exists y.x \approx s(y)))$

Substitution Lemmas (II)

Lemma 5 (trivial substitution)**.** *For any formula φ and variable x, $\varphi[x/x] \equiv \varphi$.*

Proof. We first prove an auxiliary result: For any term t and variable x,

$$(*) \qquad t[x/x] \equiv t$$

This is proved by induction on the structure of t.

1. If t is a variable, it is either equal to x or it is not. In the former case:

   $$t[x/x] \equiv x[x/x] \equiv x \equiv t$$

   In the latter case:

   $$t[x/x] \equiv t$$

   So the statement holds in either case.

2. If t is of the form $f(t_1, \ldots, t_n)$ for a function symbol f of arity $n \in \mathbb{N}$ and terms $t_1, \ldots, t_n$, we can assume that $t_i[x/x] \equiv t_i$ for all $i \in \{1, \ldots, n\}$. But then

   $$t[x/x] \equiv (f(t_1, \ldots, t_n))[x/x] \equiv f(t_1[x/x], \ldots, t_n[x/x]) \equiv f(t_1, \ldots, t_n) \equiv t$$

   Note that this result also holds in the case of $n = 0$.

This proves $(*)$. Now we prove the main result: For any formula φ and variable x,

$$(**) \qquad \varphi[x/x] \equiv \varphi$$

This is also proved by induction, this time on the structure of φ.

1. If φ is of the form $r(t_1, \ldots, t_m)$ for a relation symbol r of arity $m \in \mathbb{N}$ and terms $t_1, \ldots, t_m$, we know by $(*)$ that $t_j[x/x] \equiv t_j$ for all $j \in \{1, \ldots, m\}$. But then

   $$\varphi[x/x] \equiv (r(t_1, \ldots, t_n))[x/x] \equiv r(t_1[x/x], \ldots, t_n[x/x]) \equiv r(t_1, \ldots, t_n) \equiv \varphi$$

   Again, this result also holds for $m = 0$.

2. If φ is $\bot$, the result is immediate.

3. If φ is of the form $\varphi_1 \circ \varphi_2$ for $\circ \in \{\wedge, \vee, \rightarrow\}$, we can assume that $\varphi_1[x/x] \equiv \varphi_1$ and $\varphi_2[x/x] \equiv \varphi_2$, since both φ_1 and φ_2 are structurally smaller than φ. That means

$$\varphi[x/x] \equiv (\varphi_1 \circ \varphi_2)[x/x] \equiv \varphi_1[x/x] \circ \varphi_2[x/x] \equiv \varphi_1 \circ \varphi_2 \equiv \varphi$$

4. If φ is of the form $\forall y.\varphi'$, we distinguish between two cases:

   - if $x \equiv y$, then

   $$\varphi[x/x] \equiv (\forall x.\varphi')[x/x] \equiv \forall x.\varphi' \equiv \varphi$$

   - otherwise $x \not\equiv y$, hence $y \notin \{x\} = \mathrm{FV}(x)$, so

   $$\varphi[x/x] \equiv (\forall y.\varphi')[x/x] \equiv \forall y.\varphi'[x/x] \equiv \forall y.\varphi' \equiv \varphi$$

   where we again can assume that $\varphi'[x/x] \equiv \varphi'$ by induction hypothesis

5. The case that φ is of the form $\exists y.\varphi'$ is handled in the same manner.

In conclusion, we have proved $(*)$ by induction. □

Lemma 6 (substitution of variable that does not occur free)**.** *For any formula φ, variable $x \notin \mathrm{FV}(\varphi)$, and term t, $\varphi[t/x] \equiv \varphi$, if the result of this substitution is defined at all.*

Alpha Equivalence

- for a quantifier $Q \in \{\forall, \exists\}$, $Qx.\varphi$ *alpha reduces* to $Qy.\varphi'$ if $\varphi' \equiv \varphi[y/x]$
- φ is called *alpha equivalent* to ψ (written $\varphi \equiv_\alpha \psi$), if ψ results from φ by any number of alpha reductions on subformulas of φ
- Examples:
  - $(\forall x.R(x,x)) \equiv_\alpha (\forall y.R(y,y))$
  - $(\forall x.\exists x.S(x)) \equiv_\alpha (\forall y.\exists x.S(x)) \equiv_\alpha (\forall y.\exists z.S(z))$
  - $(\forall x.\exists y.T(x,y)) \not\equiv_\alpha (\forall x.\exists x.T(x,x))$

Notice that alpha reduction *never* changes the names of free variables.

Renaming Away

- we do not distinguish between alpha equivalent formulas
- hence, we can use alpha reduction to rename problematic bound variables such that substitution is always defined
- example:
$$(x \approx \mathbf{0} \vee \exists y.x \approx s(y))[s(y)/x]$$
is not defined, but
$$x \approx \mathbf{0} \vee \exists y.x \approx s(y) \equiv_\alpha x \approx \mathbf{0} \vee \exists z.x \approx s(z)$$
thus we can define
$$\begin{array}{rcl}(x \approx \mathbf{0} \vee \exists y.x \approx s(y))[s(y)/x] & := & (x \approx \mathbf{0} \vee \exists z.x \approx s(z))[s(y)/x] \\ & \equiv & s(y) \approx \mathbf{0} \vee \exists z.s(y) \approx s(z)\end{array}$$

Motivation: Semantics of FOL

- like in propositional logic, in FOL we do not care what functions or relations the symbols in Σ stand for
- thus, we do not know if $\forall x.x \approx \mathbf{0}$ is true
- but some sentences are intuitively true, e.g.
$$(\forall x.\forall y.R(x,y) \to R(y,x)) \to R(a,b) \to R(b,a)$$
- how do we evaluate, e.g., $\forall x.\neg R(x,x)$?
  - we need to know what x can stand for, and for which of these values R is true
  - then we would like to evaluate $\neg R(x,x)$, where x is bound to any of its possible values
- thus, we need to consider not only the interpretation of the function and relation symbols, but also variable bindings

Semantics: Structures, Interpretations and Assignments

- a (first order) structure $\mathcal{M} = \langle D, I\rangle$ for a signature Σ consists of
  - a non-empty set D, the *domain*
  - an interpretation $I = \langle [\![\,]\!]_{\mathtt{F}}, [\![\,]\!]_{\mathcal{R}}\rangle$ such that
    - for every $f \in \mathtt{F}$ with $\alpha(f) = n$, $[\![f]\!]_{\mathtt{F}} \colon D^n \to D$
    - for every $r \in \mathcal{R}$ with $\alpha(r) = n$, $[\![r]\!]_{\mathcal{R}} \colon D^n \to \mathcal{B}$
- a *variable assignment* on I is a function $\sigma \colon \mathcal{V} \to D$
  We write $\sigma[x := t]$ for the assignment
$$y \mapsto \begin{cases} t & \text{if } x \equiv y \\ \sigma(y) & \text{otherwise}\end{cases}$$

Semantics: Interpreting Terms and Formulas

- interpretation of terms over $\mathcal{M}$ and σ:
  - $[\![x]\!]_{\mathcal{M},\sigma} = \sigma(x)$
  - $[\![f(t_1, \ldots, t_n)]\!]_{\mathcal{M},\sigma} = [\![f]\!]_{\mathtt{F}}([\![t_1]\!]_{\mathcal{M},\sigma}, \ldots, [\![t_n]\!]_{\mathcal{M},\sigma})$
- interpretation of formulas:
  - $[\![r(t_1, \ldots, t_n)]\!]_{\mathcal{M},\sigma} = [\![r]\!]_{\mathcal{R}}([\![t_1]\!]_{\mathcal{M},\sigma}, \ldots, [\![t_n]\!]_{\mathcal{M},\sigma})$
  - $[\![\bot]\!]_{\mathcal{M},\sigma}$, $[\![\varphi \wedge \psi]\!]_{\mathcal{M},\sigma}$, etc.: as before
  - $[\![\forall x.\varphi]\!]_{\mathcal{M},\sigma} = \begin{cases} \mathtt{T} & \text{if, for all } d \in D,\ [\![\varphi]\!]_{\mathcal{M},\sigma[x:=d]} = \mathtt{T}, \\ \mathtt{F} & \text{otherwise} \end{cases}$
  - $[\![\exists x.\varphi]\!]_{\mathcal{M},\sigma} = \begin{cases} \mathtt{T} & \text{if there is } d \in D \text{ with } [\![\varphi]\!]_{\mathcal{M},\sigma[x:=d]} = \mathtt{T}, \\ \mathtt{F} & \text{otherwise} \end{cases}$

Example

A structure $\mathcal{M} = \langle \mathbb{N}, \langle [\![\,]\!]_{\mathtt{F}}, [\![\,]\!]_{\mathcal{R}} \rangle \rangle$ for Σ_{ar}

- $[\![\mathbf{0}]\!]_{\mathtt{F}} = 0$
- $[\![s]\!]_{\mathtt{F}}(n) = n + 1$
- $[\![+]\!]_{\mathtt{F}}(m, n) = m + n$
- $[\![\cdot]\!]_{\mathtt{F}}(m, n) = m \cdot n$
- $[\![\approx]\!]_{\mathcal{R}}(m, n) = \begin{cases} \mathtt{T} & \text{if } m = n \\ \mathtt{F} & \text{otherwise} \end{cases}$

Consider $\sigma = \{x \mapsto 0, y \mapsto 1\}$, then

- $[\![x]\!]_{\mathcal{M},\sigma} = 0$, $[\![y + y]\!]_{\mathcal{M},\sigma} = 2$, $[\![s(s(\mathbf{0}))]\!]_{\mathcal{M},\sigma} = 2$, $[\![y \approx s(s(\mathbf{0}))]\!]_{\mathcal{M},\sigma} = \mathtt{T}$
- $[\![\bot]\!]_{\mathcal{M},\sigma} = \mathtt{F}$, $[\![\neg(x \approx y)]\!]_{\mathcal{M},\sigma} = \mathtt{T}$
- $[\![\exists d.y \approx x + d]\!]_{\mathcal{M},\sigma} = \mathtt{T}$
- $[\![\exists x.\forall y.\neg(x \approx y) \wedge (\exists d.x \approx y + d)]\!]_{\mathcal{M},\sigma} = \mathtt{F}$

Satisfiability and Validity

- $\models_{\mathcal{M},\sigma} \varphi$: $[\![\varphi]\!]_{\mathcal{M},\sigma} = \mathtt{T}$
- $\models_{\mathcal{M}} \varphi$ ("M is a model for φ"): $\models_{\mathcal{M},\sigma} \varphi$ for any σ
- $\models \varphi$ ("φ is valid"): $\models_{\mathcal{M}} \varphi$ for any structure $\mathcal{M}$

- $\Gamma \models \varphi$: any $\mathcal{M}$ and σ such that $[\![\gamma]\!]_{\mathcal{M},\sigma} = \mathtt{T}$ for every $\gamma \in \Gamma$ also gives $[\![\varphi]\!]_{\mathcal{M},\sigma} = \mathtt{T}$
- analogously, $\varphi \Leftrightarrow \psi$ means that $[\![\varphi]\!]_{\mathcal{M},\sigma} = [\![\psi]\!]_{\mathcal{M},\sigma}$ for any $\mathcal{M}$ and σ

Example: $\models \exists x.D(x) \to (\forall y.D(y))$ ("Drinker Paradox")

Proof. Take the signature $\Sigma_D = \langle \emptyset, \{D\} \rangle$ with $\alpha(D) = 1$ and the set $\mathcal{V}_D = \{x, y\}$ of variables; the Drinker Paradox is clearly a formula in $\mathtt{FOL}_{\Sigma_D,\mathcal{V}_D}$.

Now assume we are given an arbitrary structure $\mathcal{M} = \langle X, \langle [\![\,]\!]_{\mathtt{F}}, [\![\,]\!]_{\mathcal{R}} \rangle \rangle$ and a variable assignment $\sigma \colon \mathcal{V}_D \to X$. By our definition of semantics, X is a non-empty set; pick an element $x_0 \in X$.

Observe that $[\![D]\!]_{\mathcal{R}}$ is a function from X to $\mathcal{B}$, i.e. $D(x)$ is either $\mathtt{T}$ or $\mathtt{F}$ for every $x \in X$. We now distinguish two cases:

- If $[\![D]\!]_{\mathcal{R}}(x)$ is $\mathtt{T}$ for all $x \in X$, then

  $$[\![D(y)]\!]_{\mathcal{M},\sigma[x:=x_0][y:=x]} = \mathtt{T}$$

  for all $x \in X$, hence

  $$[\![\forall y.D(y)]\!]_{\mathcal{M},\sigma[x:=x_0]} = \mathtt{T}$$

  Certainly also

  $$[\![D(x)]\!]_{\mathcal{M},\sigma[x:=x_0]} = \mathtt{T}$$

  and thus

  $$[\![D(x) \to (\forall y.D(y))]\!]_{\mathcal{M},\sigma[x:=x_0]} = \mathtt{T}$$

  This shows that

  $$[\![\exists x.D(x) \to (\forall y.D(y))]\!]_{\mathcal{M},\sigma} = \mathtt{T}.$$

- Otherwise, $[\![D]\!]_{\mathcal{R}}(x_1)$ is $\mathtt{F}$ for some x_1, hence

  $$[\![D(x)]\!]_{\mathcal{M},\sigma[x:=x_1]} = \mathtt{F}$$

  But then,

  $$[\![D(x) \to (\forall y.D(y))]\!]_{\mathcal{M},\sigma[x:=x_1]} = \mathtt{T}$$

  and consequently

  $$[\![\exists x.D(x) \to (\forall y.D(y))]\!]_{\mathcal{M},\sigma} = \mathtt{T}.$$

In conclusion, we have shown that $\models_{\mathcal{M},\sigma} \exists x.D(x) \to (\forall y.D(y))$ for arbitrary $\mathcal{M}$ and σ, thus establishing

$$\models \exists x.D(x) \to (\forall y.D(y)).$$

□

Caution: This is not the same as $\not\models (\exists x.D(x)) \to (\forall y.D(y))$!

Basic Results

From now on, we fix some signature Σ and a set $\mathcal{V}$ of variables.

Lemma 7 (agreement lemma)**.** *Let $\mathcal{M}$ be a structure for Σ, φ a formula, and σ, σ' variable assignments such that $\sigma(x) = \sigma'(x)$ for all $x \in \mathrm{FV}(\varphi)$. Then $\models_{\mathcal{M},\sigma} \varphi$ iff $\models_{\mathcal{M},\sigma'} \varphi$.*

Corollary 8. *The interpretation of a closed formula is independent of variable assignments.*

Lemma 9 (alpha equivalent formulas are semantically equivalent)**.** *Alpha equivalent formulas evaluate to the same truth value.*

Some Equivalences of FOL

- $(\forall x.\varphi) \Leftrightarrow \neg(\exists x.\neg\varphi)$
- $(\forall x.\varphi \wedge \psi) \Leftrightarrow (\forall x.\varphi) \wedge (\forall x.\psi)$
- $(\exists x.\varphi \vee \psi) \Leftrightarrow (\exists x.\varphi) \vee (\exists x.\psi)$
- $(\forall x.\forall y.\varphi) \Leftrightarrow (\forall y.\forall x.\varphi)$
- $(\exists x.\exists y.\varphi) \Leftrightarrow (\exists y.\exists x.\varphi)$
- $(\exists x.\forall y.\varphi) \rightarrow (\forall y.\exists x.\varphi)$, but *not* vice versa

Truth Tables for FOL?

- for $\varphi \in \texttt{PF}$, we can always find out whether $\models \varphi$ by drawing a truth table
- how about $\varphi \in \texttt{FOL}$?
  - we need to consider all possible structures
  - in particular, all possible domains, all possible functions over them
  - but domains could be infinite...
- unfortunate truth:

  Theorem 10 (Undecidability of First Order Logic)**.** *Given an arbitrary first order formula φ, it is undecidable whether $\models \varphi$.*

Index

Logic
Part II: Intuitionistic Logic and Natural Deduction

Max Schäfer

Principles of Intuitionistic Logic

- classical logic is non-constructive: the proof of the Drinker Paradox does not tell us who the drinker is (and if they drink)
- it relies on a notion of *truth* that is somewhat disconnected from the formulas of the logic
- intuitionistic logic emphasizes *provability*, examines ways to construct proofs of formulas
- proofs of complex formulas are always formulated in terms of proofs of their parts
- we are *not* interested in whether formulas are true

1 Intuitionistic Propositional Logic

The Brouwer-Heyting-Kolmogorov Interpretation

- formulas of intuitionistic propositional logic are the same as in the classical case
- their meaning is explained in terms of their proofs (not in terms of truth):
  - a proof of $\varphi \wedge \psi$ is a proof of φ together with a proof of ψ
  - a proof of $\varphi \vee \psi$ is a proof of φ or a proof of ψ
  - a proof of $\varphi \to \psi$ is a procedure that can be seen to produce a proof of ψ from a proof of φ
  - there is no proof of $\bot$

Examples

For three propositional letters P, Q, R we can prove

- $P \to P$

  Given a proof u of P, we can produce a proof of P, namely u itself. This process is a proof of $P \to P$.

- $P \wedge Q \to P$

  Assume we have a proof v of $P \wedge Q$. Then we can extract from it a proof of P, since it must contain both a proof of P and a proof of Q. So we have a procedure for constructing a proof of P from a proof of $P \wedge Q$; this is a proof of $P \wedge Q \to P$.

- $P \to (P \vee Q)$
- $P \to (Q \to P)$
- $(P \to (Q \to R)) \to (P \to Q) \to (P \to R)$

Comparison with Classical Propositional Logic

Comparison of "a formula is true" and "a formula has a proof":

- in CL, to show that $\varphi \vee \psi$ is true, we can

  1. assume that φ is false
  2. then show that ψ is true

  in the second step, we can use the fact that φ is false

- in IL, to give a proof of $\varphi \vee \psi$, we must

  1. either give a proof of φ (no matter whether ψ has one)
  2. or give a proof of ψ (no matter whether φ has one)

For other connectives, the difference is not so marked.

Comparison: Example

- in CL, φ is true if $\neg\varphi$ is false and vice versa
- in IL, if $\neg\varphi$ has a proof then there can be no proof of φ and vice versa:

  Assume we have a proof u of $\neg\varphi$. Because $\neg\varphi \equiv \varphi \to \bot$ this means that u is a procedure that produces a proof of $\bot$ given a proof of φ. But there is no proof of $\bot$, hence there can be no proof of φ.

  Assume that we have a proof v of φ. Then there can be no proof of $\neg\varphi$. For assume that we had a proof w of $\neg\varphi$; then w could produce a proof of $\bot$ from v. But this is impossible.

Comparison: Further Examples

- $\varphi \vee \neg\varphi$ is true in CL; for assume φ is false, then $\neg\varphi$ is true
- $\varphi \vee \neg\varphi$ *does not seem provable* in IL
- in CL, if $\neg\neg\varphi$ is true then so is φ; hence $\neg\neg\varphi \to \varphi$ is a classical tautology

- in IL, there *does not seem to be a way* to get a proof of φ from a proof of $\neg\neg\varphi$
- in CL, $\bot$ is never true; in IL, $\bot$ never has a proof
- in CL, $\bot \to \varphi$ is true for any φ
- in IL, $\bot \to \varphi$ is vacuously provable for any φ (*ex falso quodlibet*, EFQ)

Excursus: Why EFQ?

- in many fields of mathematics, there are contradictory propositions from which anything is derivable
- for example, if $1 = 0$ were true, then
  - $2 = 1 + 1 = 0 + 0 = 0$, $3 = 1 + 1 + 1 = 0$,...
  - hence: for all $n \in \mathbb{N}$, $n = 0$
  - but also: for all $r \in \mathbb{R}$, $r = r \cdot 1 = r \cdot 0 = 0$

  Thus, any equality between numbers holds, all functions are equal!
- in intuitionistic logic, $\bot$ abstractly represents such a proposition

Formalization: First Step

- we want to formalize the process of forming a proof, in particular a good way to handle *assumptions* (e.g., naming them)
- a diagrammatic *derivation* set out in tree-shape shows how the proof of a complex formula depends on simpler proofs
- in the course of a derivation, assumptions can temporarily be made and later discharged (see examples involving implication)

Example

Here is an informal proof of $P \wedge Q \to Q \wedge P$:

1. Assume we have a proof of $P \wedge Q$.
2. This proof contains of a proof of P.
3. It also contains a proof of Q.
4. So if we take the proof of Q and put it together with the proof of P, we obtain a proof of $Q \wedge P$.
5. We have shown how to construct a proof of $Q \wedge P$ from a proof of $P \wedge Q$. This constitutes a proof of $P \wedge Q \to Q \wedge P$.

The Example in Natural Deduction

$$\dfrac{\dfrac{\dfrac{[u\colon P \wedge Q]}{Q} \qquad \dfrac{[u\colon P \wedge Q]}{P}}{Q \wedge P}}{P \wedge Q \to Q \wedge P}$$

- the derivation is a tree with assumptions at the leaves
- assumptions are labeled (here with "u")
- the levels correspond to the steps of the informal proof
- derivation steps may *discharge* assumptions (as in the final step)
- discharged assumptions are enclosed in brackets

The Calculus NJ of Natural Deduction (Propositional Part)

- the assumption rule: assumptions can be added to the current node at any time

$$x\colon \varphi$$

- for the connectives, there are introduction and elimination rules
  - the introduction rules specify how to construct proofs
  - the elimination rules specify how to extract the information contained in a proof

The Rules for Conjunction

Conjunction Introduction:

$$(\wedge\mathrm{I})\ \dfrac{\varphi \qquad \psi}{\varphi \wedge \psi}$$

φ and ψ are *premises*, $\varphi \wedge \psi$ is the *conclusion*

Conjunction Elimination:

$$(\wedge\mathrm{E}_l)\ \dfrac{\varphi \wedge \psi}{\varphi}$$

$$(\wedge\mathrm{E}_r)\ \dfrac{\varphi \wedge \psi}{\psi}$$

Example

$$\dfrac{\dfrac{\dfrac{u\colon P \wedge (Q \wedge R)}{P}\,(\wedge \mathrm{E}_l) \qquad \dfrac{\dfrac{u\colon P \wedge (Q \wedge R)}{Q \wedge R}\,(\wedge \mathrm{E}_r)}{Q}\,(\wedge \mathrm{E}_l)}{P \wedge Q}\,(\wedge \mathrm{I}) \qquad \dfrac{\dfrac{u\colon P \wedge (Q \wedge R)}{Q \wedge R}\,(\wedge \mathrm{E}_r)}{R}\,(\wedge \mathrm{E}_r)}{(P \wedge Q) \wedge R}\,(\wedge \mathrm{I})$$

Usually, it is easier to find derivations bottom-up starting from the conclusion.

The Rules for Disjunction

Disjunction Introduction:

$$\dfrac{\varphi}{\varphi \vee \psi}\,(\vee \mathrm{I}_l)$$

$$\dfrac{\psi}{\varphi \vee \psi}\,(\vee \mathrm{I}_r)$$

Disjunction Elimination:

$$\dfrac{\varphi \vee \psi \qquad \begin{matrix}[v\colon \varphi] \\ \vdots \\ \vartheta\end{matrix} \qquad \begin{matrix}[w\colon \psi] \\ \vdots \\ \vartheta\end{matrix}}{\vartheta}\,(\vee \mathrm{E}^{v,w})$$

All open assumptions from the left subderivation are also open in the two right subderivations.

Example

$$\dfrac{u\colon P \vee Q \qquad \dfrac{[v\colon P]}{Q \vee P}\,(\vee \mathrm{I}_r) \qquad \dfrac{[w\colon Q]}{Q \vee P}\,(\vee \mathrm{I}_l)}{Q \vee P}\,(\vee \mathrm{E}^{v,w})$$

In the same manner, we can prove $(P \vee Q) \vee R$ from the assumption $P \vee (Q \vee R)$.

The Rules for Implication

Implication Introduction:

$$\dfrac{\begin{matrix}[x\colon \varphi] \\ \vdots \\ \psi\end{matrix}}{\varphi \to \psi}\,(\to \mathrm{I}^x)$$

Implication Elimination (*modus ponens*, MP):

$$\dfrac{\varphi \to \psi \qquad \varphi}{\psi}\,(\to \mathrm{E})$$

Examples

$$(\to\mathrm{I}^u)\ \frac{[u\colon P]}{P \to P}$$

$$(\to\mathrm{I}^w)\ \frac{[w\colon Q]}{\dfrac{[v\colon P]}{Q \to P}}$$
$$(\to\mathrm{I}^v)\ \frac{}{P \to Q \to P}$$

The Rules for Falsity

Falsity Introduction:

there is no introduction rule for falsity

Falsity Elimination (EFQ):

$$(\bot\mathrm{E})\ \frac{\bot}{\varphi}$$

Example:

$$(\bot\mathrm{E})\ \frac{[u\colon \bot]}{P}$$
$$(\to\mathrm{I}^u)\ \frac{}{\bot \to P}$$

Further Examples

$$(\to\mathrm{E})\ \frac{[u\colon P \to Q] \qquad [v\colon P]}{(\to\mathrm{E})\ \dfrac{Q \qquad\qquad [w\colon \neg Q]}{(\to\mathrm{I}^v)\ \dfrac{\bot}{\neg P}}}$$
$$(\to\mathrm{I}^w)\ \frac{}{\neg Q \to \neg P}$$
$$(\to\mathrm{I}^u)\ \frac{}{(P \to Q) \to \neg Q \to \neg P}$$

Further Examples

$$(\to\mathrm{E})\ \frac{[u\colon (P \vee Q) \to R] \qquad (\vee\mathrm{I}_l)\ \dfrac{[v\colon P]}{P \vee Q}}{(\to\mathrm{I}^v)\ \dfrac{R}{P \to R}} \qquad (\to\mathrm{E})\ \frac{[u\colon (P \vee Q) \to R] \qquad (\vee\mathrm{I}_r)\ \dfrac{[w\colon Q]}{P \vee Q}}{(\to\mathrm{I}^w)\ \dfrac{R}{Q \to R}}$$
$$(\wedge\mathrm{I})\ \frac{}{(P \to R) \wedge (Q \to R)}$$
$$(\to\mathrm{I}^u)\ \frac{}{(P \vee Q \to R) \to (P \to R) \wedge (Q \to R)}$$

Derivability and Theorems

- a context Γ is a set of assumptions, i.e. $\Gamma \equiv x_1 \colon \varphi_1, \ldots, x_n \colon \varphi_n$ where all the x_i are mutually distinct
- we write $\Gamma_1 \subseteq \Gamma_2$ to indicate that every assumption in Γ_1 also occurs in Γ_2
- the *range* of Γ, written $|\Gamma|$, is the set of assumption formulas in Γ, i.e. the φ_i
- we write $\Gamma \vdash_{\mathrm{NJ}} \varphi$ to mean that φ can be derived from assumptions Γ using the rules of NJ

  for example, $u \colon P \to Q, v \colon \neg Q \vdash_{\mathrm{NJ}} \neg P$
- if $\vdash_{\mathrm{NJ}} \varphi$ (i.e., φ is derivable without assumptions), then φ is a *theorem* of NJ

Some Theorems

Theorems:

- $(\varphi \to \chi \to \psi) \to (\chi \to \varphi \to \psi)$
- $\varphi \to \chi \to \psi \leftrightarrow \varphi \wedge \chi \to \psi$
- $(\varphi \to \varphi \to \psi) \wedge \varphi \to \psi$

(Apparently) Non-Theorems:

- $\varphi \vee \neg\varphi$
- $\neg\neg\varphi \to \varphi$
- $\neg(\varphi \wedge \psi) \to \neg\varphi \vee \neg\psi$
- $(\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$

Theorems:

- $\neg\neg(\varphi \vee \neg\varphi)$
- $\varphi \to \neg\neg\varphi$
- $\neg\varphi \vee \neg\psi \to \neg(\varphi \wedge \psi)$
- $(\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$

Properties of NJ(I)

Lemma 1 (Weakening)**.** *For any context* Γ *and formula* φ*, if* $\Gamma \vdash_{\mathrm{NJ}} \varphi$ *and* $\Gamma \subseteq \Gamma'$*, then* $\Gamma' \vdash_{\mathrm{NJ}} \varphi$*.*

Proof. Assume Γ, Γ', and φ are given such that $\Gamma \vdash_{\mathrm{NJ}} \varphi$ and $\Gamma \subseteq \Gamma'$. Thus there must be a derivation D with open assumptions Γ and conclusion φ. We construct a derivation D' with open assumptions Γ' and conclusion φ by induction on the structure of D.

1. If the last step of D was an application of the assumption rule, then $\varphi \in |\Gamma|$, hence also $\varphi \in |\Gamma'|$, and we can take D' to also be an application of the assumption rule.

2. If the last step of D was an application of $(\wedge\mathrm{I})$, then φ must be of the form $\varphi_1 \wedge \varphi_2$, and D looks like this:

$$\dfrac{\begin{array}{cc}\Gamma & \Gamma \\ \vdots & \vdots \\ D_1 & D_2 \\ \vdots & \vdots \\ \varphi_1 & \varphi_2\end{array}}{\varphi_1 \wedge \varphi_2}\,(\wedge\mathrm{I})$$

   Since the subderivations D_1 and D_2 are shorter than D, we can assume by induction hypothesis that there are derivations D_1' and D_2' showing $\Gamma' \vdash_{\mathrm{NJ}} \varphi_1$ and $\Gamma' \vdash_{\mathrm{NJ}} \varphi_2$. We can paste them together to obtain a derivation of $\Gamma' \vdash_{\mathrm{NJ}} \varphi$:

$$\dfrac{\begin{array}{cc}\Gamma' & \Gamma' \\ \vdots & \vdots \\ D_1' & D_2' \\ \vdots & \vdots \\ \varphi_1 & \varphi_2\end{array}}{\varphi_1 \wedge \varphi_2}\,(\wedge\mathrm{I})$$

3. All other cases are handled similarly; for example, let us consider the case that D ends in an application of $(\vee\mathrm{E}^{v,w})$. Then D looks like this:

$$\dfrac{\begin{array}{ccc}\Gamma & \Gamma,[v\colon \psi_1] & \Gamma,[w\colon \psi_2]\\ \vdots & \vdots & \vdots\\ D_1 & D_2 & D_3\\ \vdots & \vdots & \vdots\\ \psi_1 \vee \psi_2 & \varphi & \varphi\end{array}}{\varphi}$$

Note that we have explicitly annotated all the subderivations with the sets of open and closed assumptions. We will have to make sure that v and w are such that Γ' does not contain any assumptions labeled v and w; if it does, we choose different names for v and w and consistently replace them everywhere in D_2 and D_3.

Now, $\Gamma \subseteq \Gamma'$, and hence $\Gamma, v\colon \psi_1 \subseteq \Gamma', v\colon \psi_1$ and $\Gamma, w\colon \psi_1 \subseteq \Gamma', w\colon \psi_1$. Since all three of D_1, D_2, and D_3 are subderivations of D we can assume that there are derivations D_1', D_2', D_3' showing that $\Gamma' \vdash_{\mathrm{NJ}} \psi_1 \vee \psi_2$, $\Gamma', v\colon \psi_1 \vdash_{\mathrm{NJ}} \varphi$, and $\Gamma', w\colon \psi_2 \vdash_{\mathrm{NJ}} \varphi$. By pasting these three derivations together, we obtain a derivation of $\Gamma' \vdash_{\mathrm{NJ}} \varphi$.

□

Theorem 2 (Soundness Theorem)**.** *The system* NJ *is sound: If* $\vdash_{\mathrm{NJ}} \varphi$ *then* $\models \varphi$*, i.e. all theorems are propositional tautologies.*

Consequences of the Soundness Theorem

Corollary 3. *If* $\Gamma \vdash_{\mathrm{NJ}} \varphi$ *then* $|\Gamma| \models \varphi$*.*

Corollary 4. *The system* NJ *is consistent, i.e. there is a propositional formula* φ *such that we do not have* $\vdash_{\mathrm{NJ}} \varphi$*.*

Proof: Indeed, take $\bot$. If we could derive $\vdash_{\mathrm{NJ}} \bot$, then by the soundness lemma $\models \bot$. But that is not the case.

Properties of NJ**(II)**

- is natural deduction complete for classical semantics, i.e. does $\models \varphi$ imply $\vdash_{\mathrm{NJ}} \varphi$?
- no: there are classical tautologies (e.g., $P \vee \neg P$) without a proof in natural deduction
- but we obtain completeness if we replace $(\bot\mathrm{E})$ with

$$(\mathrm{DN})\ \frac{\neg\neg\varphi}{\varphi}$$

The resulting system is called NK.

2 Intuitionistic First Order Logic

Intuitionistic First Order Logic

- the language of intuitionistic first order logic is the same as with classical logic
- the BHK interpretation can be extended to quantified formulas:
  - a proof of $\forall x.\varphi$ is a procedure that can be seen, for every value a, to produce a proof of φ with x standing for a
  - a proof of $\exists x.\varphi$ is a value a for x together with a proof of φ for this value
- NJ contains introduction and elimination rules for the quantifiers

Comparison with Classical Propositional Logic

Comparison of "a formula is true" and "a formula has a proof" (ctd.):

- in CL, to show that $\exists x.\varphi$ is true, we can
  1. assume that φ is false for all x
  2. then derive a contradiction from this assumption
- in IL, to give a proof of $\exists x.\varphi$, we must present a concrete value for x (called a *witness*) and a proof that φ holds for this x

The existential quantifier of intuitionistic logic is *constructive*.

Rules for the Universal Quantifier

Universal Introduction:

$$(\forall \text{I})\ \frac{\varphi}{\forall x.\varphi}$$

where x cannot occur free in any open assumption

Universal Elimination:

$$(\forall \text{E})\ \frac{\forall x.\varphi}{\varphi[t/x]}$$

for any term t

Example

For any φ, we can build the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{u\colon \forall x.\forall y.\varphi}{\forall y.\varphi}\,(\forall \mathrm{E})}{\varphi}\,(\forall \mathrm{E})}{\forall x.\varphi}\,(\forall \mathrm{I})}{\forall y.\forall x.\varphi}\,(\forall \mathrm{I})$$

The following attempt to derive $\vdash_{\mathrm{NJ}} P(x) \to P(y)$ fails due to the side condition ($x \in \mathrm{FV}(P(x))$):

$$\dfrac{\dfrac{\dfrac{[u\colon P(x)]}{\forall x.P(x)}\,(\forall \mathrm{I})}{P(y)}\,(\forall \mathrm{E})}{P(x) \to P(y)}\,(\to \mathrm{I}^u)$$

Rules for the Existential Quantifier

Existential Introduction:

$$\dfrac{\varphi[t/x]}{\exists x.\varphi}\,(\exists \mathrm{I})$$

for any term t

Existential Elimination:

$$\dfrac{\exists x.\varphi \qquad \begin{array}{c}[u\colon \varphi]\\ \vdots \\ \psi\end{array}}{\psi}\,(\exists \mathrm{E}^u)$$

where x cannot occur free in any open assumptions on the right and in ψ

All open assumptions from the left subderivation are also open in the right subderivation.

Example

For any φ, we can build the following derivation:

$$\dfrac{u\colon \exists x.\exists y.\varphi \qquad \dfrac{[v\colon \exists y.\varphi] \qquad \dfrac{\dfrac{[w\colon \varphi]}{\exists x.\varphi}\,(\exists \mathrm{I})}{\exists y.\exists x.\varphi}\,(\exists \mathrm{I})}{\exists y.\exists x.\varphi}\,(\exists \mathrm{E}^w)}{\exists y.\exists x.\varphi}\,(\exists \mathrm{E}^v)$$

The following attempt to derive $(\exists x.P(x)) \to (\forall x.P(x))$ fails due to the variable condition:

$$\dfrac{\dfrac{u\colon \exists x.P(x) \qquad [v\colon P(x)]}{P(x)}\,(\exists \mathrm{E}^v)}{\forall x.P(x)}\,(\forall \mathrm{I})$$

Example

For any φ and ψ where $x \notin \mathrm{FV}(\varphi)$, we have

$$\varphi \vee \exists x.\psi \vdash_{\mathrm{NJ}} \exists x.\varphi \vee \psi :$$

$$\dfrac{t\colon \varphi \vee (\exists x.\psi) \qquad \dfrac{\dfrac{[u\colon \varphi]}{\varphi \vee \psi}\,(\vee \mathrm{I}_l)}{\exists x.\varphi \vee \psi}\,(\exists \mathrm{I}) \qquad \dfrac{[v\colon \exists x.\psi] \qquad \dfrac{\dfrac{[w\colon \psi]}{\varphi \vee \psi}\,(\vee \mathrm{I}_r)}{\exists x.\varphi \vee \psi}\,(\exists \mathrm{I})}{\exists x.\varphi \vee \psi}\,(\exists \mathrm{E}^w)}{\exists x.\varphi \vee \psi}\,(\vee \mathrm{E}^{u,v})$$

Example

The following attempt to derive $\forall x.\exists y.x < y \vdash_{\mathrm{NJ}} \exists x.\forall y.x < y$ fails:

$$\dfrac{\dfrac{u\colon \forall x.\exists y.x < y}{\exists y.x < y}\,(\forall \mathrm{E}) \qquad \dfrac{\dfrac{[v\colon x < y]}{\forall x.x < y}\,(\forall \mathrm{I})}{\exists y.\forall x.x < y}\,(\exists \mathrm{I})}{\exists y.\forall x.x < y}\,(\exists \mathrm{E}^v)$$

Soundness and Completeness of NJ

Theorem 5 (Soundness Theorem)**.** NJ *is sound with respect to the classical semantics.*

Theorem 6 (Completeness Theorem)**.** NK *with the quantifier rules is complete with respect to the classical semantics.*

3 Intuitionistic Second Order Propositional Logic

Second Order Propositional Logic (SOPL)

- a different extension of propositional logic: quantify over *propositions*
- for example:
  - $\forall P.P \to P$: "all propositions imply themselves"
  - $\exists P.P \leftrightarrow Q \wedge R$: "there is a proposition P that is equivalent to the conjunction of propositions Q and R"
- this use of $\forall$ and $\exists$ is very different from FOL!

The Language of SOPL

- assume we have an alphabet $\mathcal{R}$ of *propositional letters*, denoted by P, Q, R, ...
- the set $\mathtt{PF}^2_{\mathcal{R}}$ of second order propositional formulas over $\mathcal{R}$ is defined inductively:
  1. every propositional letter is a formula
  2. the symbol $\bot$ is a formula (*falsity*)
  3. if φ and ψ are formulas, then so are
     - $\varphi \wedge \psi$ (*conjunction*)
     - $\varphi \vee \psi$ (*disjunction*)
     - $\varphi \to \psi$ (*implication*)
  4. if φ is a formula and P is a propositional letter, then $\forall P.\varphi$ and $\exists P.\varphi$ are formulas (*second order universal and existential quantifier*)
- define $\mathrm{FV}(\varphi)$ similar to FOL

Classical SOPL

- it is easy to give SOPL a classical semantics
- but remember that in classical logic every proposition is either true or false, so
$$\forall P.\varphi \Leftrightarrow \varphi[\top/P] \wedge \varphi[\bot/P]$$
and
$$\exists P.\varphi \Leftrightarrow \varphi[\top/P] \vee \varphi[\bot/P]$$
- so SOPL is actually "the same" as propositional logic
- only advantage: shorter formulas

Intuitionistic SOPL

- BHK interpretation for second order quantifiers:
  - a proof of $\forall P.\varphi$ is a procedure that, for every proposition p, can be seen to produce a proof of φ with P standing for p
  - a proof of $\exists P.\varphi$ is a proposition p and a proof of φ with P standing for p
- based on this interpretation, construct system NJ^2 of natural deduction for SOPL by taking propositional rules of NJ and rules for second-order quantifiers

Rules for the Universal Quantifier in NJ^2

Universal Introduction:

$$(\forall \mathrm{I})\ \frac{\varphi}{\forall P.\varphi}$$

where P cannot occur free in any open assumption

Universal Elimination:

$$(\forall \mathrm{E})\ \frac{\forall P.\varphi}{\varphi[\psi/P]}$$

for any formula ψ

Example

$\vdash_{\mathrm{NJ}^2} \forall P.\forall Q.\forall R.(P \to R) \wedge (Q \to R) \to ((P \vee Q) \to R)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\vee \mathrm{E}^{v,w})}\dfrac{[u\colon P \vee Q] \qquad \dfrac{(\wedge \mathrm{E}_l)\dfrac{[a\colon (P \to R) \wedge (Q \to R)]}{P \to R} \qquad [v\colon P]}{(\to \mathrm{E})\dfrac{}{R}} \qquad \dfrac{(\wedge \mathrm{E}_r)\dfrac{[a\colon (P \to R) \wedge (Q \to R)]}{Q \to R} \qquad [w\colon Q]}{(\to \mathrm{E})\dfrac{}{R}}}{R}}{(\to \mathrm{I}^u)\dfrac{}{(P \vee Q) \to R}}}{(\to \mathrm{I}^a)\dfrac{}{(P \to R) \wedge (Q \to R) \to ((P \vee Q) \to R)}}}{(\forall \mathrm{I})\dfrac{}{\forall R.(P \to R) \wedge (Q \to R) \to ((P \vee Q) \to R)}}}{(\forall \mathrm{I})\dfrac{}{\forall Q.\forall R.(P \to R) \wedge (Q \to R) \to ((P \vee Q) \to R)}}}{(\forall \mathrm{I})\dfrac{}{\forall P.\forall Q.\forall R.(P \to R) \wedge (Q \to R) \to ((P \vee Q) \to R)}}$$

Rules for the Existential Quantifier

Existential Introduction:

$$(\exists \mathrm{I})\ \frac{\varphi[\psi/P]}{\exists P.\varphi}$$

for any formula ψ

Existential Elimination:

$$(\exists \mathrm{E}^u)\ \frac{\exists P.\varphi \qquad \begin{array}{c}[u\colon \varphi] \\ \vdots \\ \psi\end{array}}{\psi}$$

where P cannot occur free in any open assumptions on the right and in ψ

All open assumptions from the left subderivation are also open in the right subderivation.

Surprise: We Only Need $\forall$ and $\to$!

- as it turns out:
  - $\vdash_{\mathrm{NJ}^2} \bot \leftrightarrow (\forall P.P)$
  - $\vdash_{\mathrm{NJ}^2} \varphi \wedge \psi \leftrightarrow (\forall P.(\varphi \to \psi \to P) \to P)$
  - $\vdash_{\mathrm{NJ}^2} \varphi \vee \psi \leftrightarrow (\forall P.(\varphi \to P) \to (\psi \to P) \to P)$
  - $\vdash_{\mathrm{NJ}^2} (\exists P.\varphi) \leftrightarrow (\forall Q.(\forall P.\varphi \to Q) \to Q)$ for $Q \notin \mathrm{FV}(\varphi)$
- so we can rewrite any formula to only use $\forall$ and $\to$ without impact on provability!
- in contrast, all connectives are independent in (intuitionistic) propositional and first order logic

Index

Logic

Part III: Basic Proof Theory and Curry-Howard Correspondence

Max Schäfer

1 Logics, Semantics, and Deductive Systems

What's a Logic?

- a logic has a certain *language* in which formulas of the logic can be formulated
- the language is usually given by an inductive definition, involving one or more kinds of variables, connectives, quantifiers, etc.
- formulas of the language have some sort of intended meaning

Examples of Logics

- classical propositional logic (CPL): formulas express true or false propositions
- intuitionistic propositional logic (IPL): same language, formulas express abstract problems or statements to be proved
- classical first order logic (CFOL): same intended meaning as propositional case, more expressive formula language with quantification over individuals
- intuitionistic first order logic (IFOL): analogous
- second order propositional logic (SOL): also in classical and intuitionistic variants
- other logics: minimal logic, linear logic and its varieties, modal logics, temporal logic, Horn logic, rewrite logic, ...

Semantics

- a *semantics* for a logic interprets the logic's formulas in some mathematical domain
- this is one way of pinning down the intuitive meaning of the formulas

- many semantics define a modelling relation $\Gamma \models \varphi$
- examples:
  - Algebraic semantics: classical logic can be interpreted in Boolean algebras, particularly the algebra of truth values; intuitionistic logic can be interpreted in Heyting algebras
  - Kripke structures: describe "possible worlds"; often used to give semantics for modal and temporal logics, but also useful for intuitionistic logic
  - Categorical semantics: very flexible and powerful semantics, can be used with just about any logic

Deductive Systems

- a *deductive system* allows to infer *judgments* about formulas of a logic
- many deductive systems define an entailment relation $\Gamma \vdash \varphi$
- deductions are often written in tree form, but sometimes also in linear fashion
- this is a different way of determining the meaning of the formulas

Examples of Deductive Systems

- Natural Deduction:
  - NJ (intuitionistic logic): can infer judgments $\Gamma \vdash \varphi$ meaning that φ is provable if all of the formulas in Γ are provable
  - NK (for classical logic): $\Gamma \vdash \varphi$ means that φ is true if all formulas in Γ are true
- Sequent Calculus:
  - LJ (intuitionistic logic): same judgments as NJ, different rules
  - LK (classical logic): judgment $\Gamma \vdash \Delta$ means that some formula from Δ is true whenever all the formulas from Γ are true
- Hilbert Systems: given a list of axioms, theorems can be inferred by a (small) number of rules
- many more

Consistency

- *consistency* is an important property of a deductive system
- it means that the deductive systems does *not* allow derivations of *every* formula; i.e., there are formulas that are not derivable
- often, consistency is also defined by saying that $\bot$ cannot be derived, or that no contradiction of the form $\varphi \wedge \neg\varphi$ can be derived
- for NJ (and many other systems), these definitions are equivalent, but *not for all systems!*

Connection Between Semantics and Deductive Systems

Assume we have a semantics and a deductive system (for the same logic).

- if every derivable judgment yields a semantically true statement, then the deduction system is *sound*
- if every semantically true statement corresponds to a derivable judgment, then the deduction system is *complete*
- for example, NJ is sound for the truth value semantics, because $\Gamma \vdash_{\mathrm{NJ}} \varphi$ implies $|\Gamma| \models \varphi$
- often, completeness is a lot harder to prove than soundness; sometimes it is impossible to achieve

Comparison Between Logics

Of the logics we have seen so far

- CPL is a sub-logic of CFOL, i.e. every theorem/tautology of CPL is also a theorem/tautology of CFOL; not the other way around (because of different syntax)
- IPL is a sub-logic of CPL; not the other way around (because of different semantics)
- IPL is a sub-logic of IFOL; not the other way around (because of different syntax)

Comparison Between Classical And Intuitionistic Logic

- classical logic seems stronger than intuitionistic logic: $\vdash_{\mathrm{NK}} P \vee \neg P$, but $\nvdash_{\mathrm{NJ}} P \vee \neg P$
- this is a "misunderstanding" of the classical connectives:
  - classical $\varphi \vee \psi$ should intuitionistically be read as $\neg(\neg\varphi \wedge \neg\psi)$
  - classical $\exists x.\varphi$ should intuitionistically be read as $\neg(\forall x.\neg\varphi)$

- translation from classical φ to intuitionistic φ^*:
  - $P^* := \neg\neg P$ for $P \in \mathcal{R}$, $\bot^* := \bot$
  - $(\varphi \wedge \psi)^* := \varphi^* \wedge \psi^*$, $(\varphi \to \psi)^* := \varphi^* \to \psi^*$
  - $(\varphi \vee \psi)^* := \neg(\neg\varphi^* \wedge \neg\psi^*)$
  - $(\exists x.\varphi)^* := \neg(\forall x.\neg\varphi^*)$
  - $(\forall x.\varphi)^* := \forall x.\varphi^*$
- we can prove: $\vdash_{\mathrm{NK}} \varphi \quad$ iff $\quad \vdash_{\mathrm{NJ}} \varphi^*$
- classically, $\varphi \Leftrightarrow \varphi^*$ (but not intuitionistically!)

2 Proof Normalization

Underivability Results

- showing derivability is easy, underivability is much harder
- how do we show that $\nvdash_{\mathrm{NJ}} ((P \to Q) \to P) \to P$?
- induction on derivations does not (immediately) work: too many choices, no obvious induction hypothesis
- idea: show that all derivations can be brought into a certain normal form, then do induction on normal forms only

Detours in NJ($\to$)

- consider the subset NJ($\to$) of NJ dealing only with implication (rules $(\to\mathrm{I})$ and $(\to\mathrm{E})$)
- take a derivation of $\vdash_{\mathrm{NJ}} (P \to P \to Q) \to ((R \to R) \to P) \to Q$:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[a\colon P \to P \to Q] \qquad [u\colon P]}{P \to Q}(\to\mathrm{E}) \qquad [u\colon P]}{Q}(\to\mathrm{E})}{P \to Q}(\to\mathrm{I}^u) \qquad \dfrac{[b\colon (R \to R) \to P] \qquad \dfrac{[v\colon R]}{R \to R}(\to\mathrm{I}^v)}{P}(\to\mathrm{E})}{Q}(\to\mathrm{E})}{((R \to R) \to P) \to Q}(\to\mathrm{I}^b)}{(P \to P \to Q) \to ((R \to R) \to P) \to Q}(\to\mathrm{I}^a)
$$

- could be simplified by plugging the derivation of P onto the two assumptions $u\colon P$

Normalized Derivation

$$\dfrac{\dfrac{\dfrac{[a\colon P \to P \to Q] \qquad \dfrac{[b\colon (R \to R) \to P] \qquad \dfrac{[v\colon R]}{R \to R}\,(\to\mathrm{I}^{v})}{P}\,(\to\mathrm{E}) \qquad \dfrac{[b\colon (R \to R) \to P] \qquad \dfrac{[v\colon R]}{R \to R}\,(\to\mathrm{I}^{v})}{P}\,(\to\mathrm{E})}{P \to Q}\,(\to\mathrm{E})}{Q}}{((R \to R) \to P) \to Q}\,(\to\mathrm{I}^{b})}{(P \to P \to Q) \to ((R \to R) \to P) \to Q}\,(\to\mathrm{I}^{a})$$

- in this derivation, the left premise of an application of $(\to\mathrm{E})$ is never derived by $(\to\mathrm{I})$
- if we walk from the conclusion upwards and always choose the left premise of any rule application, we first encounter several $(\to\mathrm{I})$s, then some $(\to\mathrm{E})$s, and finally an assumption

Normalization for NJ(→)

- a *detour* is a derivation fragment like this:

$$\dfrac{\dfrac{\begin{array}{c}[u\colon \varphi]\\ \vdots\\ \psi\end{array}}{\varphi \to \psi}\,(\to\mathrm{I}^{u}) \qquad \begin{array}{c}\vdots\\ \varphi\end{array}}{\psi}\,(\to\mathrm{E})$$

- a detour can be eliminated by substituting the derivation of φ on the right for every assumption $u\colon \varphi$ on the left
- a derivation without detours is called *normal*

Theorem 1 (Strong Normalization for NJ(→))**.** *Every derivation in* NJ*(→) can be brought into normal form by eliminating detours in some arbitrary order.*

Example: Underivability

- assume $\vdash_{\mathrm{NJ}(\to)} ((P \to Q) \to P) \to P$; then we should be able to construct a normal derivation for it
- choices are dictated by normal form, lead to derivation fragment

$$\dfrac{\dfrac{[u\colon (P \to Q) \to P] \qquad \dfrac{\begin{array}{c}[u\colon (P \to Q) \to P] \qquad [v\colon P]\\ ?\\ Q\end{array}}{P \to Q}\,(\to\mathrm{I}^{v})}{P}\,(\to\mathrm{E})}{((P \to Q) \to P) \to P}\,(\to\mathrm{I}^{u})$$

Cannot be completed!

- we conclude: $\not\vdash_{\mathrm{NJ}(\to)} ((P \to Q) \to P) \to P$

Normalization for NJ

- a similar normalization result holds for full NJ (and even for NJ^2)
- roughly: always do eliminations first, then introductions
- important corollaries:
  - if $\vdash_{\mathrm{NJ}} \varphi \vee \psi$, then either $\vdash_{\mathrm{NJ}} \varphi$ or $\vdash_{\mathrm{NJ}} \psi$
    thus, $\not\vdash_{\mathrm{NJ}} P \vee \neg P$
  - if $\vdash_{\mathrm{NJ}} \exists x.\varphi$, then $\vdash_{\mathrm{NJ}} \varphi[t/x]$ for some term t
    shows constructiveness of $\exists$
  - there is an algorithm to decide whether $\vdash_{\mathrm{NJ}} \varphi$ for $\varphi \in \mathtt{PF}$
  - there is a proof search procedure which will find a proof of $\vdash_{\mathrm{NJ}} \varphi$ (for arbitrary φ) if it exists
    this procedure may not terminate if there is no proof (undecidability!)

3 The Curry-Howard Correspondence

Propositional Logic and Simply Typed Lambda Calculus

Remember the rules for $\to$ in NJ:

Implication Introduction:

$$(\to\mathrm{I}^x)\ \frac{\begin{array}{c}[x\colon \varphi]\\ \vdots\\ \psi\end{array}}{\varphi \to \psi}$$

Implication Elimination:

$$(\to\mathrm{E})\ \frac{\varphi \to \psi \qquad \varphi}{\psi}$$

Here is a variant with explicit contexts:

Implication Introduction:

$$(\to\mathrm{I}^x)\ \frac{\Gamma, x\colon \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}$$

Implication Elimination:

$$(\to\mathrm{E})\ \frac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$$

Propositional Logic and Simply Typed Lambda Calculus (ctd.)

Here is a variant with explicit contexts:

Type inference rules for Simply Typed Lambda Calculus:

Implication Introduction:

$$(\to\mathrm{I}^x)\ \frac{\Gamma, x\colon \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}$$

Abstraction:

$$(\textsc{abs})\ \frac{\Gamma, x\colon \varphi \vdash M\colon \psi}{\Gamma \vdash (\lambda x\colon \varphi.M)\colon \varphi \to \psi}$$

Implication Elimination:

$$(\to\mathrm{E})\ \frac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$$

Application:

$$(\textsc{app})\ \frac{\Gamma \vdash M\colon \varphi \to \psi \qquad \Gamma \vdash N\colon \varphi}{\Gamma \vdash (M\,N)\colon \psi}$$

Proof Terms

- propositions and types have the same structure
- simply typed lambda terms of type φ represent NJ proofs of proposition φ (*proof terms*)
- they provide a concrete "implementation" of the BHK interpretation: a proof term of an implication $\varphi \to \psi$ is a lambda term which, when applied to a proof term of φ, yields a proof term of ψ
- we can annotate derivations in NJ with their proof terms

Example: Lambda Terms and Derivation Trees

- this is the derivation tree corresponding to $\lambda x\colon A.x$:

$$(\to\mathrm{I}^x)\ \frac{[x\colon A]}{(\lambda x\colon A.x)\colon A \to A}$$

- this is for $\lambda s\colon A \to A.\lambda z\colon A.s(sz)$:

$$(\to\mathrm{I}^{s})\ \frac{(\to\mathrm{I}^{z})\ \dfrac{(\to\mathrm{E})\ \dfrac{[s\colon A\to A]\qquad (\to\mathrm{E})\ \dfrac{[s\colon A\to A]\qquad [z\colon A]}{(s\,z)\colon A}}{(s\,(s\,z))\colon A}}{(\lambda z\colon A.s\,(s\,z))\colon A\to A}}{(\lambda s\colon A\to A.\lambda z\colon A.s\,(s\,z))\colon (A\to A)\to A\to A}$$

Normalization

- normalization in NJ corresponds to normalization in the lambda calculus; detour elimination is beta reduction
- example (rule labels omitted for brevity):

$$\frac{\dfrac{[f\colon A\to A]}{(\lambda f\colon A\to A.f)\colon (A\to A)\to A\to A}\qquad \dfrac{[x\colon A]}{(\lambda x\colon A.x)\colon A\to A}}{((\lambda f\colon A\to A.f)\,(\lambda x\colon A.x))\colon A\to A}$$

$$\Downarrow$$

$$\frac{[x\colon A]}{(\lambda x\colon A.x)\colon A\to A}$$

  corresponds to
  $(\lambda f\colon A\to A.f)\,(\lambda x\colon A.x)\longrightarrow_{\beta}(\lambda x\colon A.x)$
- thus, programs written in simply typed lambda calculus always terminate!

Connectives and Datatypes

- conjunction corresponds to pairing:

$$(\wedge\mathrm{I})\ \frac{s\colon\varphi\qquad t\colon\psi}{(s,t)\colon\varphi\wedge\psi}$$

$$(\wedge\mathrm{E}_{l})\ \frac{p\colon\varphi\wedge\psi}{(\mathbf{fst}\,p)\colon\varphi}$$

$$(\wedge\mathrm{E}_{r})\ \frac{p\colon\varphi\wedge\psi}{(\mathbf{snd}\,p)\colon\psi}$$

  Compare Haskell:

- `(_,_) : a -> b -> (a, b)`
- `fst : (a, b) -> a`
- `snd : (a, b) -> b`

- likewise, disjunction is disjoint sum, comparable to Haskell

```
data Sum a b = Inl a | Inr b
```

Second Order Propositional Logic and System F

- terms of System F (polymorphic lambda calculus) are proof terms of intuitionistic second order propositional logic
- example:

$$\dfrac{\dfrac{\dfrac{\dfrac{[x\colon A]}{(\mathbf{inl}_B\, x)\colon A \vee B}}{(\lambda x\colon B.\mathbf{inl}_B\, x)\colon A \to A \vee B}}{(\Lambda B.\lambda x\colon B.\mathbf{inl}_B\, x)\colon \forall B.a \to A \vee B}}{(\Lambda A.\Lambda B.\lambda x\colon B.\mathbf{inl}_B\, x)\colon \forall A.\forall B.a \to A \vee B}$$

- many datatypes (booleans, numbers, lists, trees,...) can be encoded in System F
- all programs implemented in this system terminate

First Order Logic and Dependent Types

- in second order logic, there are formulas depending on formulas ($\forall P.P \vee \neg P$); they correspond to types depending on types (polymorphic types)

  e.g. lists of integers
- in first order logic, there are formulas depending on terms ($\forall x.P(x)$); they correspond to types depending on values (dependent types)

  e.g. lists of length 10
- both features improve type safety; e.g., consider function **tail**
  - polymorphic typing: $\mathbf{tail} : \forall\alpha.\mathbf{list}\,\alpha \to \mathbf{list}\,\alpha$

    ensures that the resulting list has the same element type as the argument list
  - dependent typing: $\mathbf{tail} : \forall n.\mathbf{intlist}\,(1+n) \to \mathbf{intlist}\,n$

    ensures that the resulting list is one element shorter than the argument list, and that the function cannot be given an empty argument

Higher Order Logic/Constructive Type Theory

- in higher order logic (HOL), we have *both* polymorphic and dependent types; programs and proofs about them are expressed in the same system
- the $\exists$ quantifier supports program specification:
  - proof of $\exists x.P(x)$ is a pair (a, p), where a is an individual and p a proof of $P(a)$
  - expression of type $\exists x.P(x)$ is a pair (a, p), where a is another expression/program fulfilling specification $P(a)$, as proved by p
  - for example, if $P(f)$ expresses "f is a function that sorts its input list", then $\exists f.P(f)$ is a pair (f', p), where
    - f' is a function (the implementation)
    - p is a proof of $P(f')$, i.e. a proof that the implementation fulfills the specification
- example systems: Coq, Isabelle, Agda

Programming in Higher Order Logic

- programming in many HOL systems is similar to functional programming
- e.g., lists with length in Agda:

```
data List (A : Set) : Nat -> Set where
  []   : List A 0
  _::_ : {n : Nat} -> A -> List A n -> List A (1 + n)
```

- `head` function:

```
tail : {A : Set}{n : Nat} -> List A (1 + n) -> List A n
tail (_ :: xs) = xs
```

- example typings:
  - `23 : Nat` `42 : Nat`
  - `23::[] : List Nat 1` `23::(42::[]) : List Nat 2`
  - `tail (23::(42::[])) : List Nat 1`
  - `tail []` cannot be typed, rejected by compiler!
- Agda is strongly normalizing, so all Agda programs always terminate (and Agda is not Turing complete!)

Curry-Howard Correspondence

in logic:	**in programming languages**:
formulas	types
proofs	lambda terms
implication	function type
conjunction	pair type
disjunction	disjoint sum type
propositional logic	simply typed lambda calculus
(first order) quantification	dependent types
(second order) quantification	polymorphic types
proof normalization	term reduction
induction	fold
classical logic	programs with "jumps"

Recommended Reading

- Dirk van Dalen: Logic and Structure

  introductory text, uses natural deduction for classical logic

- Jean-Yves Girard, Paul Taylor, Yves Lafont: Proofs and Types

  introductory text, uses natural deduction for intuitionistic logic, emphasis on second order logic and System F

- Greg Restall: Proof Theory and Philosophy

  natural deduction for different logics, emphasis on proof theory

- Morten Sorensen, Pavel Urzyczyn: Lectures on the Curry-Howard Isomorphism

  very comprehensive, covers different deduction systems and semantics for propositional, first, and higher order logic

Conclusion

- modern mathematics (and with it all of modern science) uses logic, mostly informally
- we can, however, formalize it
- there is not one "true" logical system; different systems are useful for different purposes
- for any given system, there can be multiple explanations (semantics)
- for any given system, there can be multiple deduction systems
- we can study the capabilities of these systems and their relationship

- logic is used in computer science in many different areas, from artificial intelligence to program verification
- in this course, we have only touched upon the basics; there is a lot more out there!

Index