

Logic

Lecture 1: intuitionistic logic

27 August 2012

柯向上

Department of Computer Science
University of Oxford

Hsiang-Shang.Ko@cs.ox.ac.uk

Formal logic

Highlights of this course:

- the use of formal languages to represent reasoning patterns;
- reasoning in the *meta-language* about the *object language* (especially by *induction*);
- the intimate connection between logic and computation.

Propositional logic

In propositional logic, we formalise and study various logical *connectives* like “and”, “or”, and “implies” that we use to combine propositions into more complex ones.

For an extreme example, the truth of the following proposition is determined by the way we use the connectives alone.

if *herba viridi* **and** *area est infectum*, **then** *area est infectum*

The actual meanings/structures of the two *atomic propositions* “*herba viridi*” and “*area est infectum*” do not matter.

Syntactic structure of propositions

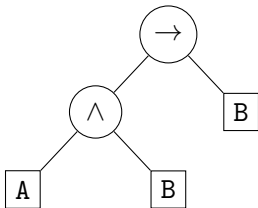
The proposition from the previous slide:

if *herba viridi* **and** *area est infectum*, **then** *area est infectum*

Since the contents of the atomic propositions do not matter, we may just replace them with simpler symbols.

if \boxed{A} **and** \boxed{B} , **then** \boxed{B}

The syntactic structure is more explicitly shown as a tree, which we call a *propositional formula*.



Building up a propositional formula

The “jigsaw pieces” we have:

- an infinite supply of propositional variable symbols \boxed{A} , \boxed{B} , \boxed{C} , ...,
- a special atomic proposition symbol $\bigcirc \perp$, and
- three connective symbols that combine two propositional formulas — *conjunction* $\bigcirc \wedge$, *disjunction* $\bigcirc \vee$, and



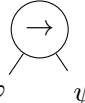
implication $\bigcirc \rightarrow$.

Even though there are an infinite number of jigsaw pieces, we can only build up a propositional formula by using a finite number of them.

Typesetting propositional formulas

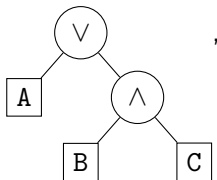
We will just use a one-dimensional syntax to describe the tree-shaped propositional formulas.

- Boxes around propositional variable symbols and the circle enclosing \perp are omitted.
- For the connective symbols, we write $\varphi \wedge \psi$, $\varphi \vee \psi$, and

$\varphi \rightarrow \psi$ to denote  ,  , and  ,

putting parentheses around φ and ψ where necessary.

Example. $A \vee (B \wedge C)$ denotes



whereas $A \vee B \wedge C$ is ambiguous.

Formal definition of propositional formulas

Let $\mathcal{PV} = \{A, B, C, \dots\}$ be an infinite set of propositional variable symbols.

Definition. The set PROP of propositional formulas is *inductively defined* by the following rules:

- $\perp : \text{PROP}$;
- $v : \text{PROP}$ if $v : \mathcal{PV}$;
- $\varphi \wedge \psi : \text{PROP}$ if $\varphi, \psi : \text{PROP}$;
- $\varphi \vee \psi : \text{PROP}$ if $\varphi, \psi : \text{PROP}$;
- $\varphi \rightarrow \psi : \text{PROP}$ if $\varphi, \psi : \text{PROP}$.

Remark. The set PROP contains exactly the propositional formulas that can be built up by using a finite number of jigsaw pieces previously specified.

Induction

We can perform *induction* on an inductively defined set.

Practically, we can

- define functions on the inductively defined set and
- prove properties about all the elements in the set

by “case analysis”.

Defining functions on PROP

Definition. The function $nAtoms : \text{PROP} \rightarrow \mathbb{N}$, which computes the number of occurrences of atomic propositional formulas in a propositional formula, is defined by

$$\begin{aligned}nAtoms \perp &= 1 \\nAtoms v &= 1 && \text{for } v : \mathcal{PV} \\nAtoms (\varphi \wedge \psi) &= nAtoms \varphi + nAtoms \psi \\nAtoms (\varphi \vee \psi) &= nAtoms \varphi + nAtoms \psi \\nAtoms (\varphi \rightarrow \psi) &= nAtoms \varphi + nAtoms \psi.\end{aligned}$$

Notation. Function application is denoted by juxtaposition, which has the highest precedence.

Well-definedness of $nAtoms$

Why is the function $nAtoms$ well-defined? That is, why can we unambiguously assign a value to each propositional formula according to the equations?

- In the definition of $nAtoms$, we have exactly one case corresponding to each rule in the definition of $PROP$.
- Every $\varphi : PROP$ is necessarily constructed using instances of the rules.
- So given any $\varphi : PROP$, we can always unambiguously find the corresponding computations specified in the definition of $nAtoms$.

Example. $nAtoms ((A \wedge B) \rightarrow (B \wedge A)) = (1 + 1) + (1 + 1) = 4$.

Sub-formulas

Definition. The function $sub : \text{PROP} \rightarrow \text{LIST PROP}$ computing the list of occurrences of *sub-formulas* in a propositional formula is defined by

$$\begin{aligned}sub \perp &= [\perp] \\sub v &= [v] && \text{for } v : \mathcal{PV} \\sub (\varphi \wedge \psi) &= [\varphi \wedge \psi] \# sub \varphi \# sub \psi \\sub (\varphi \vee \psi) &= [\varphi \vee \psi] \# sub \varphi \# sub \psi \\sub (\varphi \rightarrow \psi) &= [\varphi \rightarrow \psi] \# sub \varphi \# sub \psi.\end{aligned}$$

Example.

$$\begin{aligned}sub ((A \wedge B) \rightarrow (B \wedge A)) \\= [(A \wedge B) \rightarrow (B \wedge A), A \wedge B, A, B, B \wedge A, B, A]\end{aligned}$$

Notation. $[_]$ turns an element into a singleton list, and ‘ $\#$ ’ is list concatenation.

Induction principle on PROP

Let $P \varphi$ be a property on $\varphi : \text{PROP}$. If we can show that P can be “propagated” by every construction rule of PROP, then for any $\varphi : \text{PROP}$, a proof of $P \varphi$ can be derived in the same way as how φ is constructed.

Slightly more formally, $P \varphi$ holds for every $\varphi : \text{PROP}$ if

- $P \perp$ holds,
- $P v$ holds for every $v : \mathcal{PV}$,
- for any $\varphi, \psi \in \text{PROP}$, $P (\varphi \wedge \psi)$ holds if $P \varphi$ and $P \psi$ hold,
- for any $\varphi, \psi \in \text{PROP}$, $P (\varphi \vee \psi)$ holds if $P \varphi$ and $P \psi$ hold, and
- for any $\varphi, \psi \in \text{PROP}$, $P (\varphi \rightarrow \psi)$ holds if $P \varphi$ and $P \psi$ hold.

Inductive proof on PROP

Theorem. $|sub \varphi| = 2 \times nAtoms \varphi - 1$ for every $\varphi : \text{PROP}$.

PROOF Induction on φ .

1 $|sub \perp| = 2 \times nAtoms \perp - 1$.

2 **ASSUME** $v : \mathcal{PV}$ **PROVE** $|sub v| = 2 \times nAtoms v - 1$

3 **ASSUME** $\varphi : \text{PROP}$, $|sub \varphi| = 2 \times nAtoms \varphi - 1$,
 $\psi : \text{PROP}$, $|sub \psi| = 2 \times nAtoms \psi - 1$

PROVE $|sub (\varphi \wedge \psi)| = 2 \times nAtoms (\varphi \wedge \psi) - 1$

4 **ASSUME** the same as in **3**

PROVE $|sub (\varphi \vee \psi)| = 2 \times nAtoms (\varphi \vee \psi) - 1$

5 **ASSUME** the same as in **3**

PROVE $|sub (\varphi \rightarrow \psi)| = 2 \times nAtoms (\varphi \rightarrow \psi) - 1$

6 QED.

PROOF By **1**–**5** and the induction principle on PROP.

$|sub \varphi| = 2 \times nAtoms \varphi - 1$ for every $\varphi : PROP$

1 $|sub \perp| = 2 \times nAtoms \perp - 1$.

PROOF Evaluate the left-hand and right-hand sides.

1.1 $|sub \perp| = 1$.

1.2 $2 \times nAtoms \perp - 1 = 1$.

1.3 QED.

PROOF **1.1** and **1.2**, and indeed $1 = 1$.

2 **ASSUME** $v : PV$

PROVE $|sub v| = 2 \times nAtoms v - 1$

PROOF Similar to **1**.

$|sub \varphi| = 2 \times nAtoms \varphi - 1$ for every $\varphi : PROP$

3 **ASSUME** $\varphi : PROP, |sub \varphi| = 2 \times nAtoms \varphi - 1,$
 $\psi : PROP, |sub \psi| = 2 \times nAtoms \psi - 1$

PROVE $|sub (\varphi \wedge \psi)| = 2 \times nAtoms (\varphi \wedge \psi) - 1$

PROOF We reason:

$$\begin{aligned} & |sub (\varphi \wedge \psi)| \\ = & \quad \{ \text{definition of } sub \} \\ & |[\varphi \wedge \psi] \# sub \varphi \# sub \psi| \\ = & \quad \{ \text{list size} \} \\ & 1 + |sub \varphi| + |sub \psi| \\ = & \quad \{ \text{induction hypothesis} \} \\ & 1 + 2 \times nAtoms \varphi - 1 + 2 \times nAtoms \psi - 1 \\ = & \quad \{ \text{arithmetic} \} \\ & 2 \times (nAtoms \varphi + nAtoms \psi) - 1 \\ = & \quad \{ \text{definition of } nAtoms \} \\ & 2 \times nAtoms (\varphi \wedge \psi) - 1. \end{aligned}$$

$|sub \varphi| = 2 \times nAtoms \varphi - 1$ for every $\varphi : PROP$

4 **ASSUME** $\varphi : PROP, |sub \varphi| = 2 \times nAtoms \varphi - 1,$
 $\psi : PROP, |sub \psi| = 2 \times nAtoms \psi - 1$

PROVE $|sub (\varphi \vee \psi)| = 2 \times nAtoms (\varphi \vee \psi) - 1$

PROOF Similar to 3.

5 **ASSUME** $\varphi : PROP, |sub \varphi| = 2 \times nAtoms \varphi - 1,$
 $\psi : PROP, |sub \psi| = 2 \times nAtoms \psi - 1$

PROVE $|sub (\varphi \rightarrow \psi)| = 2 \times nAtoms (\varphi \rightarrow \psi) - 1$

PROOF Similar to 3.

Meta-connectives

For this course, we simply take the negation of a propositional formula φ to be $\varphi \rightarrow \perp$, which can be shortened to $\neg\varphi$.

We can define other connectives in the same way. For example,

$$\top := \neg\perp = \perp \rightarrow \perp$$

and

$$\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi).$$

Note that ' \neg ', ' \top ', and ' \leftrightarrow ' are defined in the meta-language.

Syntax vs. semantics

The propositional formulas are defined to be a certain type of trees and nothing more. They are merely symbols amenable to mechanic manipulation.

We need to *interpret* the symbols to something we intuitively understand to make them meaningful (useful) to us.

The Brouwer–Heyting–Kolmogorov interpretation

The *BHK interpretation* is one possible (informal) interpretation of the elements of PROP.

What is a proposition? From the *intuitionist* position, a proposition is an expression of what counts as its *proof*.

- There is no proof of \perp .
- A proof of $\varphi \wedge \psi$ is a proof of φ and a proof of ψ .
- A proof of $\varphi \vee \psi$ is either a proof of φ or a proof of ψ .
- A proof of $\varphi \rightarrow \psi$ is a method which constructs a proof of ψ given a proof of φ .

A proposition is considered true if there is a proof of it, and is considered false if there is a proof of its negation.

The Brouwer–Heyting–Kolmogorov interpretation

Example. A proof of $(A \wedge B) \rightarrow (B \wedge A)$ is a method that converts a proof of $A \wedge B$ into one of $B \wedge A$.

Does such a method exist? Yes! Assume that a proof of $A \wedge B$ is given. Since a proof of $A \wedge B$ is a proof of A and a proof of B , we certainly have a proof of B and a proof of A , which together constitute a proof of $B \wedge A$. This method serves as a proof of $(A \wedge B) \rightarrow (B \wedge A)$.

Natural deduction

“Assume that a proof of $A \wedge B$ is given. Since a proof of $A \wedge B$ is a proof of A and a proof of B , we certainly have a proof of B and a proof of A , which together constitute a proof of $B \wedge A$. This method serves as a proof of $(A \wedge B) \rightarrow (B \wedge A)$.”

The above reasoning can be rendered as a *derivation* in *natural deduction* as follows:

$$\frac{\frac{\frac{A \wedge B \vdash A \wedge B}{A \wedge B \vdash B} (\wedge ER) \quad \frac{\frac{A \wedge B \vdash A \wedge B}{A \wedge B \vdash A} (\wedge EL)}{A \wedge B \vdash B \wedge A} (\wedge I)}{\vdash (A \wedge B) \rightarrow (B \wedge A)} (\rightarrow I)$$

Deduction rule

A derivation in natural deduction consists of pre-specified deduction rules like

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi} (\wedge I)$$

A node $\Gamma \vdash \varphi$ (read “ Γ entails φ ”; the symbol ‘ \vdash ’ is called “turnstile”) can be intuitively understood as “the proposition φ can be proved under *context* Γ ” where the context Γ is a (possibly infinite) list of propositions whose proofs are assumed to exist.

Nodes above the line are called *premises* and the node below the line is called the *conclusion*.

The rule is given a name $(\wedge I)$, which stands for “ \wedge -introduction”.

Introduction and elimination rules

For each connective we have

- *introduction rule(s)* which introduce the connective in the conclusion, and
- *elimination rule(s)* which eliminate the connective in a premise.

For example, the introduction rule for ' \wedge ' is

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi} (\wedge I)$$

as we have seen, while there are two elimination rules for ' \wedge ':

$$\frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi} (\wedge EL) \quad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi} (\wedge ER)$$

The BHK interpretation of a connective is reflected in its introduction rule(s).

Assumption rule

We also have the *assumption rule*

$$\frac{}{\Gamma \vdash \varphi} \text{ (assum)}$$

which has a *side condition* that $\varphi \in \Gamma$.

The name of the rule can be omitted in derivations as this cannot cause confusion.

Introducing and eliminating ' \rightarrow '

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi} (\rightarrow I) \qquad \frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi} (\rightarrow E)$$

Example.

$$\frac{\frac{\frac{\Gamma \vdash A \rightarrow B \rightarrow C}{\Gamma \vdash B \rightarrow C} (\rightarrow E) \quad \frac{\Gamma \vdash A}{\Gamma \vdash B} (\rightarrow E)}{\Gamma} \quad \frac{\frac{\frac{\frac{\frac{\frac{\Gamma}{A \rightarrow B \rightarrow C, B, A \vdash C} (\rightarrow I)}{A \rightarrow B \rightarrow C, B \vdash A \rightarrow C} (\rightarrow I)}{A \rightarrow B \rightarrow C \vdash B \rightarrow A \rightarrow C} (\rightarrow I)}{\vdash (A \rightarrow B \rightarrow C) \rightarrow (B \rightarrow A \rightarrow C)} (\rightarrow I)}}{\Gamma} (\rightarrow E)$$

Notation. We stipulate that ' \rightarrow ' associates to the right, so $A \rightarrow B \rightarrow C$ is shorthand for $A \rightarrow (B \rightarrow C)$.

Introducing and eliminating '∨'

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi} \text{ (}\forall\text{IL)} \quad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi} \text{ (}\forall\text{IR)}$$

$$\frac{\Gamma \vdash \varphi \vee \psi \quad \Gamma, \varphi \vdash \vartheta \quad \Gamma, \psi \vdash \vartheta}{\Gamma \vdash \vartheta} \text{ (}\forall\text{E)}$$

Example.

$$\frac{\overline{A \vee B \vdash A \vee B} \quad \frac{\overline{A \vee B, A \vdash A}}{A \vee B, A \vdash B \vee A} \text{ (}\forall\text{IR)} \quad \frac{\overline{A \vee B, B \vdash B}}{A \vee B, B \vdash B \vee A} \text{ (}\forall\text{IL)}}{\frac{A \vee B \vdash B \vee A}{\vdash A \vee B \rightarrow B \vee A} \text{ (}\rightarrow\text{I)}}$$

Notation. We stipulate that '→' has lower precedence than '∧' and '∨', so $A \vee B \rightarrow B \vee A$ is shorthand for $(A \vee B) \rightarrow (B \vee A)$.

Eliminating \perp

There is no introduction rule for \perp . The elimination rule is a form of the *principle of explosion*.

$$\frac{\Gamma \vdash \perp}{\Gamma \vdash \varphi} (\perp E)$$

Example.

$$\frac{\frac{\frac{\frac{A \vee B, \neg A, A \vdash \neg A}{A \vee B, \neg A \vdash A \vee B}}{A \vee B, \neg A, A \vdash \perp} (\perp E) \quad \frac{\frac{A \vee B, \neg A, A \vdash A}{A \vee B, \neg A, B \vdash B} (\vee E)}{A \vee B, \neg A \vdash B} (\rightarrow I) \quad \frac{\frac{A \vee B \vdash \neg A \rightarrow B}{\vdash A \vee B \rightarrow \neg A \rightarrow B} (\rightarrow I)}{A \vee B, \neg A, A \vdash B} (\rightarrow E)}{A \vee B, \neg A, B \vdash B} (\vee E)$$

Notation. We stipulate that ' \neg ' has higher precedence than ' \wedge ', ' \vee ', and ' \rightarrow ', so $\neg A \rightarrow B$ is shorthand for $(\neg A) \rightarrow B$.

Derivability

Let NJ denote the deduction system that consists exactly of all the previous deduction rules.

A derivation in NJ is *closed* if there are no premises left.

A formula φ is *derivable* from a list Γ of formulas in NJ if there is a closed derivation in NJ whose conclusion is $\Gamma \vdash \varphi$. In this case we write $\Gamma \vdash_{\text{NJ}} \varphi$.

A formula φ is called a *theorem* if $\emptyset \vdash_{\text{NJ}} \varphi$. In this case we simply write $\vdash_{\text{NJ}} \varphi$.

Remark on negation and disjunction

We can derive

- $\neg\neg(\varphi \vee \neg\varphi)$ but not $\varphi \vee \neg\varphi$ (*law of excluded middle*),
- $\varphi \rightarrow \neg\neg\varphi$ but not $\neg\neg\varphi \rightarrow \varphi$ (*principle of indirect proof*),
- $\neg\varphi \vee \neg\psi \rightarrow \neg(\varphi \wedge \psi)$ but not $\neg(\varphi \wedge \psi) \rightarrow \neg\varphi \vee \neg\psi$, and
- $(\varphi \rightarrow \psi) \rightarrow (\neg\psi \rightarrow \neg\varphi)$ but not $(\neg\psi \rightarrow \neg\varphi) \rightarrow (\varphi \rightarrow \psi)$

Treat negation and disjunction with care; in particular, do not unconsciously cancel out double negations!

Syntactic nature of deduction systems

Even though we used the BHK interpretation to explain the rules of natural deduction, the deduction system itself is really just a game of symbols, whose rules we must strictly follow.

Do not arbitrarily invent new rules!

An alternative elimination rule for ' \rightarrow '

The following rule is sound according to the BHK interpretation, but is not one of the pre-specified rules in NJ.

$$\frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi \quad \Gamma, \psi \vdash \vartheta}{\Gamma \vdash \vartheta} (\rightarrow E')$$

We can, however, reduce it to the introduction rule and the standard elimination rule for ' \rightarrow '.

$$\frac{\frac{\Gamma, \psi \vdash \vartheta}{\Gamma \vdash \psi \rightarrow \vartheta} (\rightarrow I) \quad \frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi} (\rightarrow E)}{\Gamma \vdash \vartheta} (\rightarrow E)$$

This shows the *admissibility* of $(\rightarrow E')$.

Definition. A deduction rule is *admissible* if a closed derivation of its conclusion can be constructed from closed derivations of its premises.

A richer structure of propositions

When talking about mathematical structures like natural number arithmetic, we use statements like

for every x , if $x \neq 0$ then **there exists** y such that $suc\ y = x$

that involve *quantification* over individuals, which is not present in the language of propositional logic.

(The function *suc* is the *successor* function on natural numbers.)

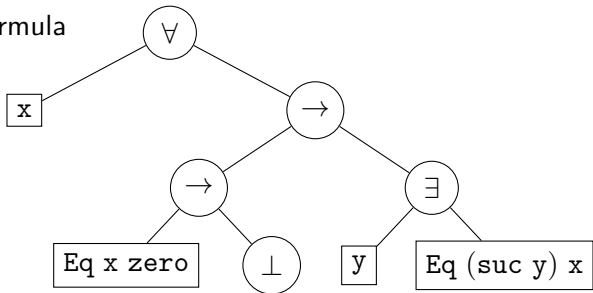
This motivates us to extend propositional logic with first-order quantification, the result of which is called *first-order logic*.

Adding quantification to the language

We represent the statement

for every x , if $x \neq 0$ then **there exists** y such that $\text{suc } y = x$

as the formula



which is simply typeset as

$$\forall x. \neg(\text{Eq } x \text{ zero}) \rightarrow \exists y. \text{Eq } (\text{suc } y) \ x$$

Notation. The scope of a quantifier extends as far as possible.

Substitution

Variables are to be *substituted* for. For example, from

$$\forall x. \neg(\text{Eq } x \text{ zero}) \rightarrow \exists y. \text{Eq } (\text{suc } y) x$$

we should be able to deduce

$$\neg(\text{Eq } (\text{suc } \text{zero}) \text{ zero}) \rightarrow \exists y. \text{Eq } (\text{suc } y) (\text{suc } \text{zero})$$

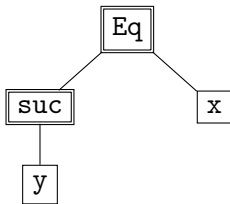
by substituting “suc zero” for the variable x .

The structure of the atomic formula “Eq (suc y) x ” must be refined so the variable x can be substituted.

Sub-atomic structure

In the atomic formula $\text{Eq}(\text{suc } y) x$,

- 'Eq' is a *predicate symbol* that accepts two *terms*, and
- 'suc' is a *function symbol* that can be used to construct more complex terms, which can contain variables.



Each symbol has an associated natural number called its *arity*, which specifies the number of sub-terms the symbol expects.

Terms

Let $\mathcal{IV} = \{x, y, z, \dots\}$ be an infinite set of individual variable symbols.

Definition. Given a set \mathcal{F} of symbols with arities, the set $\text{TERM}_{\mathcal{F}}$ of *terms* is inductively defined by the following rules:

- $v : \text{TERM}_{\mathcal{F}}$ if $v : \mathcal{IV}$;
- for any $f : \mathcal{F}$ with arity n ,
 $f t_1 \dots t_n : \text{TERM}_{\mathcal{F}}$ if $t_1, \dots, t_n : \text{TERM}_{\mathcal{F}}$.

Example. For terms in Peano/Heyting arithmetic, we choose $\mathcal{F} := \{\text{zero}/0, \text{suc}/1, \text{add}/2, \text{mult}/2\}$.

First-order formulas

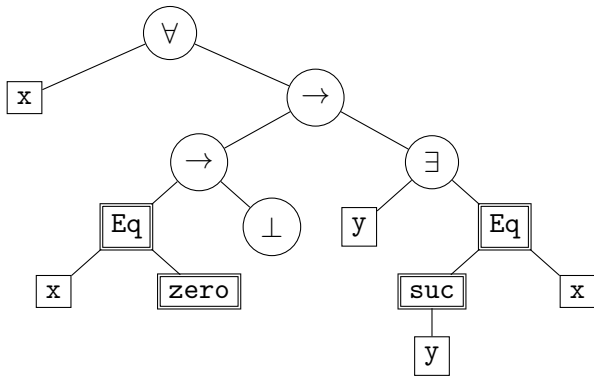
Definition. A *signature* \mathcal{S} is a pair of sets $(\mathcal{P}, \mathcal{F})$ of symbols with arities, where elements of \mathcal{P} are called *predicate symbols* and elements of \mathcal{F} are called *function symbols*.

Definition. Given a signature $\mathcal{S} = (\mathcal{P}, \mathcal{F})$, the set $\text{FORM}_{\mathcal{S}}$ of *first-order formulas* is defined by the following rules:

- $\perp : \text{FORM}_{\mathcal{S}}$;
- for any $p : \mathcal{P}$ with arity n ,
 $p t_1 \dots t_n : \text{FORM}_{\mathcal{S}}$ if $t_1, \dots, t_n : \text{TERM}_{\mathcal{F}}$;
- $\varphi \wedge \psi : \text{FORM}_{\mathcal{S}}$ if $\varphi, \psi : \text{FORM}_{\mathcal{S}}$;
- $\varphi \vee \psi : \text{FORM}_{\mathcal{S}}$ if $\varphi, \psi : \text{FORM}_{\mathcal{S}}$;
- $\varphi \rightarrow \psi : \text{FORM}_{\mathcal{S}}$ if $\varphi, \psi : \text{FORM}_{\mathcal{S}}$;
- $\forall v. \varphi : \text{FORM}_{\mathcal{S}}$ if $v : \mathcal{IV}$ and $\varphi : \text{FORM}_{\mathcal{S}}$;
- $\exists v. \varphi : \text{FORM}_{\mathcal{S}}$ if $v : \mathcal{IV}$ and $\varphi : \text{FORM}_{\mathcal{S}}$.

First-order formulas

Example. The signature for Peano/Heyting arithmetic consists of $\mathcal{P} := \{ \text{Eq}/2 \}$ and $\mathcal{F} := \{ \text{zero}/0, \text{suc}/1, \text{add}/2, \text{mult}/2 \}$.

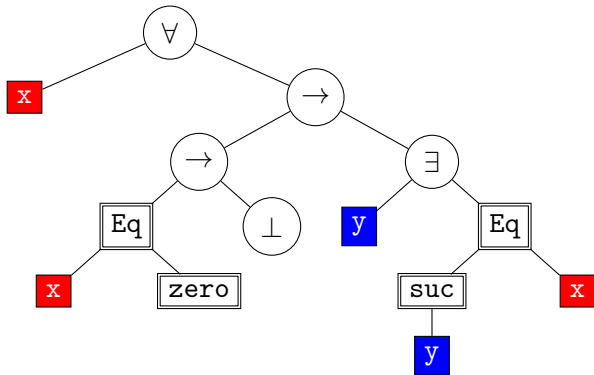


$$\forall x. \neg(\text{Eq } x \text{ zero}) \rightarrow \exists y. \text{Eq } (\text{suc } y) x$$

Variable binding

Variable binding refers to the association of variable occurrences with quantifiers.

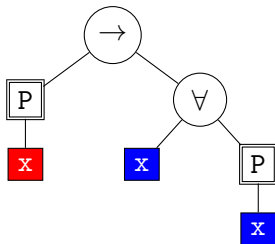
$$\forall x. \neg(\text{Eq } x \text{ zero}) \rightarrow \exists y. \text{Eq} (\text{suc } y) x$$



Free/bound occurrences of variables

A quantifier does not necessarily bind all occurrences of the associated variable, but only the *free occurrences*.

$$P\ x \rightarrow \forall\ x.\ P\ x$$

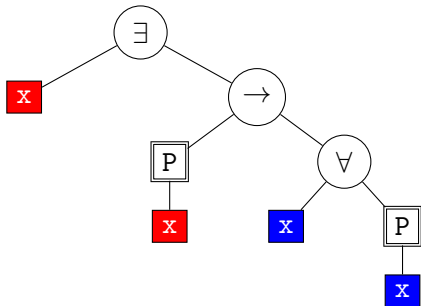


The first occurrence of x is free, whereas the last one is bound by ' $\forall x$ '.

Free/bound occurrences of variables

A quantifier does not necessarily bind all occurrences of the associated variable, but only the *free occurrences*.

$$\exists x. (P x \rightarrow \forall x. P x)$$



The outermost '∃x' binds the previously free occurrence of x, which now becomes bound.

Free variables

Definition. Let $\mathcal{S} = (\mathcal{P}, \mathcal{F})$ be a signature. The function $FV: \text{FORM}_{\mathcal{S}} \rightarrow \text{LIST } \mathcal{IV}$ computing the list of free variables in a first-order formula is defined by

$$\begin{aligned}FV \perp &= \emptyset \\FV (p \ t_1 \ \dots \ t_n) &= FV \ t_1 \cup \dots \cup FV \ t_n \quad \text{for } p : \mathcal{P} \\FV (\varphi \wedge \psi) &= FV \ \varphi \cup FV \ \psi \\FV (\varphi \vee \psi) &= FV \ \varphi \cup FV \ \psi \\FV (\varphi \rightarrow \psi) &= FV \ \varphi \cup FV \ \psi \\FV (\forall v. \ \varphi) &= FV \ \varphi \setminus v \\FV (\exists v. \ \varphi) &= FV \ \varphi \setminus v,\end{aligned}$$

where $FV: \text{TERM}_{\mathcal{F}} \rightarrow \text{LIST } \mathcal{IV}$ is defined by

$$\begin{aligned}FV \ v &= [v] \quad \text{for } v : \mathcal{IV} \\FV (f \ t_1 \ \dots \ t_n) &= FV \ t_1 \cup \dots \cup FV \ t_n \quad \text{for } f : \mathcal{F}.\end{aligned}$$

Free variables

Notation. 'U' concatenates two lists and removes duplicate elements. '\ ' takes a list and an element and returns the list with the element removed.

Variable capture

Naive substitution can result in undesired change of variable binding.

$$\begin{aligned} & (\neg(\text{Eq } x \text{ zero}) \rightarrow \exists y. \text{Eq } (\text{suc } y) x) [y/x] \\ & \neq \neg(\text{Eq } y \text{ zero}) \rightarrow \exists y. \text{Eq } (\text{suc } y) y \end{aligned}$$

Instead, we should perform *α -conversion*, which allows us to change names of bound variables to *fresh* ones, where necessary.

$$\begin{aligned} & (\neg(\text{Eq } x \text{ zero}) \rightarrow \exists y. \text{Eq } (\text{suc } y) x) [y/x] \\ & = (\neg(\text{Eq } x \text{ zero}) \rightarrow \exists z. \text{Eq } (\text{suc } z) x) [y/x] \\ & = \neg(\text{Eq } y \text{ zero}) \rightarrow \exists z. \text{Eq } (\text{suc } z) y \end{aligned}$$

α -equivalence

We casually adopt *α -equivalence* between first-order formulas: formulas that differ only in the naming of bound variables are equated.

We are thus allowed to freely change the names of bound variables to fresh ones.

Example.

$$\begin{aligned} & \neg(\text{Eq } x \text{ zero}) \rightarrow \exists y. \text{Eq } (\text{suc } y) \ x \\ & = \neg(\text{Eq } x \text{ zero}) \rightarrow \exists z. \text{Eq } (\text{suc } z) \ x \\ & \neq \neg(\text{Eq } x \text{ zero}) \rightarrow \exists x. \text{Eq } (\text{suc } x) \ x \end{aligned}$$

Definition of substitution

Definition. Let $\mathcal{S} = (\mathcal{P}, \mathcal{F})$ be a signature, $t : \text{TERM}_{\mathcal{F}}$, and $v : \mathcal{IV}$. The function $_ [t/v] : \text{FORM}_{\mathcal{S}} \rightarrow \text{FORM}_{\mathcal{S}}$, which substitutes t for v in a first-order formula, is defined by

$$\begin{aligned} \perp [t/v] &= \perp \\ (p \ t_1 \ \dots \ t_n) [t/v] &= p \ (t_1 [t/v]) \ \dots \ (t_n [t/v]) \quad \text{for } p : \mathcal{P} \\ (\varphi \wedge \psi) [t/v] &= \varphi [t/v] \wedge \psi [t/v] \\ (\varphi \vee \psi) [t/v] &= \varphi [t/v] \vee \psi [t/v] \\ (\varphi \rightarrow \psi) [t/v] &= \varphi [t/v] \rightarrow \psi [t/v] \\ (\forall u. \varphi) [t/v] &= \forall u. \varphi [t/v] \quad \text{where } u \neq v \text{ and } u \notin \text{FV } t \\ (\exists u. \varphi) [t/v] &= \exists u. \varphi [t/v] \quad \text{where } u \neq v \text{ and } u \notin \text{FV } t, \end{aligned}$$

where $_ [t/v] : \text{TERM}_{\mathcal{F}} \rightarrow \text{TERM}_{\mathcal{F}}$ is defined by

$$\begin{aligned} u [t/v] &= \text{if } u = v \text{ then } t \text{ else } u \quad \text{for } u : \mathcal{IV} \\ (f \ t_1 \ \dots \ t_n) [t/v] &= f \ (t_1 [t/v]) \ \dots \ (t_n [t/v]) \quad \text{for } f : \mathcal{F}. \end{aligned}$$

Example of substitution

$$\begin{aligned} & (\neg(\text{Eq } x \text{ zero}) \rightarrow \exists y. \text{Eq } (\text{suc } y) x) [\text{add } x \text{ y/x}] \\ = & \quad \{ \text{case '}\rightarrow\text{' } \} \\ & (\neg(\text{Eq } x \text{ zero})) [\text{add } x \text{ y/x}] \rightarrow (\exists y. \text{Eq } (\text{suc } y) x) [\text{add } x \text{ y/x}] \\ = & \quad \{ \alpha\text{-conversion } \} \\ & (\neg(\text{Eq } x \text{ zero})) [\text{add } x \text{ y/x}] \rightarrow (\exists z. \text{Eq } (\text{suc } z) x) [\text{add } x \text{ y/x}] \\ = & \quad \{ \text{case '}\exists\text{' } \} \\ & (\neg(\text{Eq } x \text{ zero})) [\text{add } x \text{ y/x}] \rightarrow \exists z. (\text{Eq } (\text{suc } z) x) [\text{add } x \text{ y/x}] \\ = & \quad \{ \text{predicate symbol } \} \\ & (\neg(\text{Eq } x \text{ zero})) [\text{add } x \text{ y/x}] \rightarrow \\ & \quad \exists z. \text{Eq } ((\text{suc } z) [\text{add } x \text{ y/x}]) (x [\text{add } x \text{ y/x}]) \end{aligned}$$

Example of substitution

$$\begin{aligned} & (\neg(\text{Eq } x \text{ zero}) \rightarrow \exists y. \text{Eq } (\text{suc } y) x) [\text{add } x \text{ y/x}] \\ = & \quad \{ \text{previous slide} \} \\ & (\neg(\text{Eq } x \text{ zero})) [\text{add } x \text{ y/x}] \rightarrow \\ & \quad \exists z. \text{Eq } ((\text{suc } z) [\text{add } x \text{ y/x}]) (x [\text{add } x \text{ y/x}]) \\ = & \quad \{ \text{function symbol} \} \\ & (\neg(\text{Eq } x \text{ zero})) [\text{add } x \text{ y/x}] \rightarrow \\ & \quad \exists z. \text{Eq } (\text{suc } (z [\text{add } x \text{ y/x}])) (x [\text{add } x \text{ y/x}]) \\ = & \quad \{ \text{variable (twice)} \} \\ & (\neg(\text{Eq } x \text{ zero})) [\text{add } x \text{ y/x}] \rightarrow \exists z. \text{Eq } (\text{suc } z) (\text{add } x \text{ y}) \\ = & \quad \{ \dots \} \\ & \neg(\text{Eq } (\text{add } x \text{ y}) \text{ zero}) \rightarrow \exists z. \text{Eq } (\text{suc } z) (\text{add } x \text{ y}) \end{aligned}$$

BHK interpretation of quantifiers

We assume a set \mathcal{D} , called the *domain of discourse*, over which we quantify.

- A proof of $\forall v. \varphi$ is a method that, for every $d : \mathcal{D}$, produces a proof of φ about d .
- A proof of $\exists v. \varphi$ is a value $d : \mathcal{D}$ (called the *witness*) and a proof of φ about d .

To obtain a deduction system for intuitionistic first-order logic, we extend NJ with introduction and elimination rules for ' \forall ' and ' \exists '.

Introducing and eliminating '∀'

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall v. \varphi} (\forall I) \quad \frac{\Gamma \vdash \forall v. \varphi}{\Gamma \vdash \varphi [t/v]} (\forall E)$$

(∀I) has a side condition that $v \notin FV \Gamma$, where

$$FV \Gamma := \bigcup [FV \varphi \mid \varphi \in \Gamma].$$

Example.

$$\frac{\frac{\frac{\frac{\frac{\frac{\forall x. \forall y. P x y \vdash \forall x. \forall y. P x y}{\forall x. \forall y. P x y \vdash \forall y. P x y} (\forall E)}{\forall x. \forall y. P x y \vdash P x y} (\forall E)}{\forall x. \forall y. P x y \vdash \forall x. P x y} (\forall I)}{\forall x. \forall y. P x y \vdash \forall y. \forall x. P x y} (\forall I)}{\vdash (\forall x. \forall y. P x y) \rightarrow \forall y. \forall x. P x y} (\rightarrow I)$$

Non-example of $(\forall I)$

$$\frac{\frac{\frac{\text{Eq } x \text{ zero} \vdash \text{Eq } x \text{ zero}}{\text{Eq } x \text{ zero} \vdash \forall x. \text{Eq } x \text{ zero}} (\forall I)}{\vdash \text{Eq } x \text{ zero} \rightarrow \forall x. \text{Eq } x \text{ zero}} (\rightarrow I)}{\vdash \forall x. \text{Eq } x \text{ zero} \rightarrow \forall x. \text{Eq } x \text{ zero}} (\forall I)}{\vdash \text{Eq } \text{zero} \text{ zero} \rightarrow \forall x. \text{Eq } x \text{ zero}} (\forall E)$$

The topmost $(\forall I)$ is illegal, since x appears free in $\text{Eq } x \text{ zero}$.

Introducing and eliminating '∃'

$$\frac{\Gamma \vdash \varphi [t/v]}{\Gamma \vdash \exists v. \varphi} (\exists I) \qquad \frac{\Gamma \vdash \exists v. \varphi \quad \Gamma, \varphi \vdash \psi}{\Gamma \vdash \psi} (\exists E)$$

(∃E) has a side condition that $v \notin FV\Gamma \cup FV\psi$.

Example.

$$\frac{\frac{\Gamma, \forall y. P x y \vdash \forall y. P x y}{\Gamma, \forall y. P x y \vdash P x y} (\forall E)}{\Gamma \vdash \exists x. \forall y. P x y} (\exists I) \qquad \frac{\Gamma, \forall y. P x y \vdash \forall y. P x y}{\Gamma, \forall y. P x y \vdash \exists x. P x y} (\exists I) \qquad \frac{\Gamma \vdash \exists x. \forall y. P x y \quad \Gamma, \forall y. P x y \vdash \exists x. P x y}{\Gamma \vdash \exists x. \forall y. P x y \rightarrow \forall y. \exists x. P x y} (\exists E)$$

$$\frac{\frac{\Gamma \vdash \exists x. \forall y. P x y \vdash \exists x. P x y}{\Gamma \vdash \exists x. \forall y. P x y \vdash \forall y. \exists x. P x y} (\forall I)}{\vdash (\exists x. \forall y. P x y) \rightarrow \forall y. \exists x. P x y} (\rightarrow I)$$

Non-example of $(\exists E)$

$$\frac{\frac{\frac{\overline{\exists x. P x \vdash \exists x. P x} \quad \overline{\exists x. P x, P x \vdash P x}}{\exists x. P x \vdash P x} (\exists I)}{\exists x. P x \vdash \forall x. P x} (\forall I)}{\vdash (\exists x. P x) \rightarrow \forall x. P x} (\rightarrow I) \quad (\exists E)$$

The topmost $(\exists E)$ is illegal, since x appears free in $P x$ in the conclusion.

Remark on negation and the existential quantifier

We can derive

$$(\exists v. \neg\varphi) \rightarrow (\neg\forall v. \varphi) \quad \text{but not} \quad (\neg\forall v. \varphi) \rightarrow (\exists v. \neg\varphi).$$

Do not unconsciously convert a negated universal statement into an existential statement!

Heyting arithmetic

The signature for Heyting arithmetic consists of $\mathcal{P} := \{ \text{Eq}/2 \}$ and $\mathcal{F} := \{ \text{zero}/0, \text{suc}/1, \text{add}/2, \text{mult}/2 \}$.

We write $t_1 \equiv t_2$ for $\text{Eq } t_1 \ t_2$, $t_1 + t_2$ for $\text{add } t_1 \ t_2$, and $t_1 \times t_2$ for $\text{mult } t_1 \ t_2$.

Properties about these constants are postulated by the *Peano axioms*.

Peano axioms

The first three axioms make 'Eq' an equivalence relation.

$$\textit{reflexivity} \quad := \quad \forall x. x \equiv x$$

$$\textit{transitivity} \quad := \quad \forall x. \forall y. \forall z. x \equiv y \wedge y \equiv z \rightarrow x \equiv z$$

$$\textit{symmetry} \quad := \quad \forall x. \forall y. x \equiv y \rightarrow y \equiv x$$

Peano axioms

The next three axioms are about zero and 'suc'.

$$\textit{disjointness} \quad := \quad \forall x. \neg(\text{suc } x \equiv \text{zero})$$

$$\textit{injectivity} \quad := \quad \forall x. \forall y. \text{suc } x \equiv \text{suc } y \rightarrow x \equiv y$$

$$\textit{congruence} \quad := \quad \forall x. \forall y. x \equiv y \rightarrow \text{suc } x \equiv \text{suc } y$$

Peano axioms

The following four axioms characterise 'plus' and 'mult'.

$$\textit{additionZ} \quad := \quad \forall y. \text{zero} + y \equiv y$$

$$\textit{additionS} \quad := \quad \forall x. \forall y. (\text{suc } x) + y \equiv \text{suc } (x + y)$$

$$\textit{multiplicationZ} \quad := \quad \forall y. \text{zero} \times y \equiv \text{zero}$$

$$\textit{multiplicationS} \quad := \quad \forall x. \forall y. (\text{suc } x) \times y \equiv x + (x \times y)$$

Peano axioms

Finally there is an *axiom scheme* that generates instances of the induction principle on natural numbers: for every formula φ and variable v there is an axiom

$$\begin{aligned} \text{induction}_{\varphi, v} &:= \\ \text{closure } (\varphi [\text{zero}/v] \wedge (\forall v. \varphi \rightarrow \varphi [\text{suc } v/v]) &\rightarrow \forall v. \varphi) \end{aligned}$$

Definition. The *universal closure* of a formula ψ is defined by

$$\text{closure } \psi := \forall v_1. \dots \forall v_n. \psi \quad \text{where } FV \psi = [v_1, \dots, v_n].$$

Example: $1 + 1 = 2$

Let **HA** be the list containing exactly the Peano axioms. We show that **HA** \vdash_{NJ} $\text{suc zero} + \text{suc zero} \equiv \text{suc} (\text{suc zero})$.

$$\frac{\frac{\frac{\frac{\text{HA} \vdash \text{zero} + \text{zero} = \text{zero} \quad \text{HA} \text{ transitivity} \quad (42)}{\text{HA} \vdash \text{zero} + \text{suc zero} = \text{suc zero} \quad (43)}{\text{HA} \vdash \text{zero} + \text{suc zero} + \text{suc zero} = \text{suc} (\text{suc zero}) \quad (44)}{\text{HA} \vdash \text{suc zero} + \text{suc zero} = \text{suc} (\text{suc zero}) \quad (45)} \quad \frac{\frac{\text{HA} \vdash \text{zero} + \text{zero} = \text{zero} \quad \text{HA} \text{ addition} \quad (46)}{\text{HA} \vdash \text{suc zero} + \text{zero} = \text{suc zero} \quad (47)}{\text{HA} \vdash \text{suc zero} + \text{suc zero} = \text{suc} (\text{suc zero}) \quad (48)} \quad \frac{\frac{\frac{\text{HA} \vdash \text{zero} + \text{suc zero} = \text{suc zero} \quad \text{HA} \text{ congruence} \quad (49)}{\text{HA} \vdash \text{suc zero} + \text{suc zero} = \text{suc} (\text{suc zero}) \quad (50)}{\text{HA} \vdash \text{suc zero} + \text{suc zero} = \text{suc} (\text{suc zero}) \quad (51)} \quad \frac{\text{HA} \vdash \text{suc zero} + \text{suc zero} = \text{suc} (\text{suc zero}) \quad (52)}{\text{HA} \vdash \text{suc zero} + \text{suc zero} = \text{suc} (\text{suc zero}) \quad (53)}{\text{HA} \vdash \text{suc zero} + \text{suc zero} = \text{suc} (\text{suc zero}) \quad (54)}$$

Informally:

- The left-hand side $\text{suc zero} + \text{suc zero}$ of ' \equiv ' is transformed into $\text{suc} (\text{zero} + \text{suc zero})$ by *additionS*.
- The sub-term $\text{zero} + \text{suc zero}$ is just suc zero by *additionZ*, so by *congruence* we can derive that $\text{suc} (\text{zero} + \text{suc zero})$ is equal to $\text{suc} (\text{suc zero})$.
- The above two equations are concatenated by *transitivity*.

Example: $\mathbf{HA} \vdash_{\text{NJ}} \forall x. x \equiv \text{zero} \vee \exists y. x \equiv \text{suc } y$

This requires induction to analyse x .

$$\frac{\frac{\frac{\frac{\text{HA} \vdash \text{reflexivity}}{\text{HA} \vdash \text{zero} \equiv \text{zero}}}{\text{HA} \vdash \text{zero} \equiv \text{zero} \vee \exists y. \text{zero} \equiv \text{suc } y}}{\text{HA} \vdash \text{induction}_{x=\text{zero} \vee \exists y. x \equiv \text{suc } y, x}}}{\text{HA} \vdash (\text{zero} \equiv \text{zero} \vee \exists y. \text{zero} \equiv \text{suc } y) \wedge (\forall x. x \equiv \text{zero} \vee \exists y. x \equiv \text{suc } y \rightarrow \text{suc } x \equiv \text{zero} \vee \exists y. \text{suc } x \equiv \text{suc } y)}}{\text{HA} \vdash \forall x. x \equiv \text{zero} \vee \exists y. x \equiv \text{suc } y}$$

Informally:

- We invoke the induction principle on the formula $\varphi := x \equiv \text{zero} \vee \exists y. x \equiv \text{suc } y$ and variable x .
- The first proof obligation $\varphi [\text{zero}/x]$ is discharged by choosing the left-hand side $\text{zero} \equiv \text{zero}$ of ‘ \vee ’ and instantiating *reflexivity*.
- For the second proof obligation $\forall x. \varphi \rightarrow (\varphi [\text{suc } x/x])$, we choose the right-hand side $\exists y. \text{suc } x \equiv \text{suc } y$, supply x as the witness, and invoke *reflexivity* again.

Theories

Definition. A formula φ is called a *sentence* if $FV \varphi = \emptyset$.

Definition. A list of sentences is called a *theory*, whose elements are called *axioms*.

Definition. A sentence derivable from a theory \mathcal{T} is called a *theorem* of \mathcal{T} .

Example. **HA** is a theory;
 $\text{suc zero} + \text{suc zero} \equiv \text{suc} (\text{suc zero})$ and
 $\forall x. x \equiv \text{zero} \vee \exists y. x \equiv \text{suc } y$ are theorems of **HA**.

Syntactic consistency and completeness

Definition. A theory \mathcal{T} is (syntactically) *inconsistent* if $\mathcal{T} \vdash_{\text{NJ}} \perp$; otherwise it is (syntactically) *consistent*.

Theorem. Let \mathcal{T} be a theory. The following statements are equivalent:

- \mathcal{T} is inconsistent;
- there is a sentence φ such that $\mathcal{T} \vdash_{\text{NJ}} \varphi$ and $\mathcal{T} \vdash_{\text{NJ}} \neg\varphi$;
- $\mathcal{T} \vdash_{\text{NJ}} \varphi$ for every sentence φ .

Definition. A theory \mathcal{T} is (syntactically) *complete* if, for every sentence φ , either $\mathcal{T} \vdash_{\text{NJ}} \varphi$ or $\mathcal{T} \vdash_{\text{NJ}} \neg\varphi$.