

IN3070/4070 – Logic – Autumn 2019

Lecture 4: First-order Logic

Martin Giese

11th September 2019



DEPARTMENT OF
INFORMATICS



UNIVERSITY OF
OSLO

Today's Plan

- ▶ Motivation
- ▶ Syntax
- ▶ Variables
- ▶ Semantics
- ▶ The Substitution Lemma
- ▶ Satisfiability & Validity
- ▶ LK for First-order Logic
- ▶ Summary

Outline

- ▶ Motivation
- ▶ Syntax
- ▶ Variables
- ▶ Semantics
- ▶ The Substitution Lemma
- ▶ Satisfiability & Validity
- ▶ LK for First-order Logic
- ▶ Summary

Limitations of Propositional Logic

Propositional logic: atomic formula (p, q, r) , \wedge , \vee , \neg , \rightarrow , $(,)$

Problem: How do we represent the following statements?

- ▶ "all men are mortal" $\forall x(\text{man}(x) \rightarrow \text{mortal}(x))$
- ▶ "there exist prime numbers that are even" $\exists y(\text{prime}(y) \wedge \text{even}(y))$
- ▶ "1 is smaller than 3" $1 < 3$ or $<(1, 3)$
- ▶ "transitivity of smaller" $\forall x \forall y \forall z (x < y \wedge y < z \rightarrow x < z)$
- ▶ $2 * 8 = 16$ $= (*(2, 8), 16)$
- ▶ "if x is even then $x + 2$ is even" $\forall x (\text{even}(x) \rightarrow \text{even}(x + 2))$
- ▶ "if x is prime then $x + 2$ is prime" $\forall x (\text{prime}(x) \rightarrow \text{prime}(x + 2))$

First-order logic: extension of propositional logic

First-Order Logic — Overview

Extending propositional logic by . . .

Syntax:

- ▶ constants (a, b, c), functions (f, g, h), variables (x, y, z)
- ▶ predicates (p, q, r)
- ▶ terms (t, u, v)
- ▶ quantifiers (\forall, \exists)
- ▶ scope of variables, free variables, variable assignment/substitution

Semantics:

- ▶ **interpretation** of constants, functions, variables
- ▶ **interpretation** of predicates
- ▶ **value** of terms
- ▶ **truth value** of (quantified) formulae
- ▶ satisfiability, validity, logical equivalence, . . .

Outline

- ▶ Motivation
- ▶ Syntax
- ▶ Variables
- ▶ Semantics
- ▶ The Substitution Lemma
- ▶ Satisfiability & Validity
- ▶ LK for First-order Logic
- ▶ Summary

Syntax — Terms

Terms are built up of constant (symbols), variable (symbols), and function (symbols).

Definition 2.1 (Terms).

Let $\mathcal{A} = \{a, b, \dots\}$ be a countable set of **constant symbols**, $\mathcal{V} = \{x, y, z, \dots\}$ be a countable set of **variable symbols**, and $\mathcal{F} = \{f, g, h, \dots\}$ be a countable set of **function symbols**.

Terms, denoted t, u, v , are inductively defined as follows:

- ▶ Every variable $x \in \mathcal{V}$ is a term.
- ▶ Every constant $a \in \mathcal{A}$ is a term.
- ▶ If $f \in \mathcal{F}$ is an n -ary function (symbol) $n > 0$ and t_1, \dots, t_n are terms, then $f(t_1, \dots, t_n)$ is a term.

Example: $a, x, f(a, x), f(g(x), b)$, and $g(f(a, g(y)))$ are terms.

Syntax — First-Order Formulae

Formulae are built up of **atomic formulae** and the **logical connectives** $\neg, \wedge, \vee, \rightarrow$, and \forall (universal quantifier), \exists (existential quantifier).

Definition 2.2 (Atomic Formulae).

Let $\mathcal{P} = \{p, q, r, \dots\}$ be a countable set of **predicate symbols**. If $p \in \mathcal{P}$ is an n -ary predicate (symbol) $n \geq 0$ and t_1, \dots, t_n are terms, then $p(t_1, \dots, t_n), \top$, and \perp are **atomic formulae** (or **atoms**).

Definition 2.3 ((First-Order) Formulae).

(First-order) formulae, denoted A, B, C, F, G, H , are inductively defined as follows:

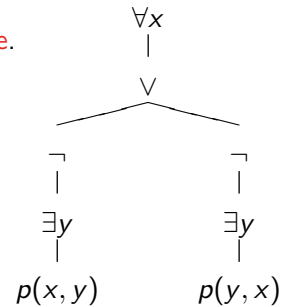
- ▶ Every atomic formula p is a formula.
- ▶ If A and B are formulae and $x \in \mathcal{V}$, then $(\neg A), (A \wedge B), (A \vee B), (A \rightarrow B), \forall x A$, and $\exists x A$ are formulae.

Formula Trees

A formula can be presented as **formula tree**.

Example:

$\forall x (\neg \exists y p(x, y) \vee \neg \exists y p(y, x))$



Definition 2.4 (Subformula, Main Operator).

Formula A is a (*proper*) **subformula** of formula B iff A is a (*proper*) subtree of B . If the root of a formula tree of A is a logical connective/quantifier, then it is called the **main operator** of A .

Outline

- ▶ Motivation
- ▶ Syntax
- ▶ Variables
- ▶ Semantics
- ▶ The Substitution Lemma
- ▶ Satisfiability & Validity
- ▶ LK for First-order Logic
- ▶ Summary

Free Variables

A **free variable** is a variable that is not in the **scope** of a quantifier.

Definition 3.1 (Free/Bound Variables, Closed Formula/Term).

Free variables in a formula A are inductively defined:

- ▶ If A is an atomic formula, then all variables in A are free.
- ▶ If $A = \neg B$, then the free variables of A are exactly those of B .
- ▶ If $A = B \wedge C$, $A = B \vee C$, or $A = B \rightarrow C$, then the free variables of A are those of B together with those of C .
- ▶ If $A = \forall x B$ or $A = \exists x B$, then the free variables of A are those of B without the variable x .

A **bound variable** in a formula C is a variable that appears in $\forall x$ or $\exists x$ in some subformula of C . A formula/term is **closed** iff it has no free variables.

Scope, Universal and Existential Closure

Definition 3.2 (Scope of Variables).

Let $\forall x A$ or $\exists x A$ be a universally or existentially **quantified formula**. Then x is the **quantified variable** and its **scope** is the formula A .

Remark: It is not required that x actually appears in the scope of its quantification, e.g. $\forall x \exists y p(y, y)$.

Definition 3.3 (Universal and Existential Closure).

If $\{x_1, \dots, x_n\}$ are all the free variables of A , the **universal closure** of A is $\forall x_1 \dots \forall x_n A$ and the **existential closure** of A is $\exists x_1 \dots \exists x_n A$.

- ▶ $p(x, y)$ has the two free variables x and y . Its universal closure is $\forall x \forall y p(x, y)$ and its existential closure is $\exists x \exists y p(x, y)$; $\exists y p(x, y)$ has the only free variable x ; $\forall x \exists y p(x, y)$ is closed
- ▶ In $\forall x p(x) \wedge q(x)$, the x occurs bound and free. The universal closure is $\forall x (\forall x p(x) \wedge q(x))$; **renaming**: $\forall y (\forall x p(x) \wedge q(y))$

Substitutions

Free variables in a first-order formula can be substituted by terms.

Definition 3.4 (Substitution).

Let \mathcal{V} be a set of variables, \mathcal{T} be the set of terms. A **substitution** $\sigma : \mathcal{V} \rightarrow \mathcal{T}$ assigns each variable a term.

Remark: The substitution σ is often represented as set $\{x \setminus t \mid \sigma(x) = t\}$.

Example: For the variable set $\{x, y\}$, $\sigma(x) = a$, $\sigma(y) = f(z, b)$ is a substitution and can also be represented as $\{x \setminus a, y \setminus f(z, b)\}$.

Ben-Ari: $\{x \leftarrow a, y \leftarrow f(z, b)\}$.

Others: $[a/x, f(z, b)/y]$

Application of substitutions

Definition 3.5 (Application of Substitutions, informally).

Let σ be a substitution. The **application** of σ to a term t or formula A , written $\sigma(t)$ or $\sigma(A)$, replaces every free variable in t or A according to its image under σ . Short hand: $A[x \setminus t] = \sigma(A)$ with $\sigma = \{x \setminus t\}$.

Example: Let $\sigma = \{x \setminus a, y \setminus f(z, b)\}$ be a substitution.

Then $\sigma(g(y)) = g(f(z, b))$

and $\sigma(p(x) \wedge \forall x q(x, g(y))) = p(a) \wedge \forall x q(x, g(f(z, b)))$

Problem: $\sigma(\forall z p(z, y)) = \forall z p(z, f(z, b))$

The free variable z in σ is **captured** by the quantifier.

This is **bad** because the effect depends on the choice of variable names

Definition 3.6 (Capture-free substitution).

A substitution σ is **capture-free** for a formula A if for every free variable x in A , none of the variables in $\sigma(x)$ is bound in A .

Application of Substitutions

Definition 3.7 (Application of Substitutions, formally).

The application of a substitution σ to a term or formula is defined by structural induction:

- ▶ $\sigma(x) = \sigma(x)$ for variables x in the range of σ
- ▶ $\sigma(y) = y$ for variables y not in the range of σ
- ▶ $\sigma(a) = a$ for constants $a \in \mathcal{A}$
- ▶ $\sigma(f(t_1, \dots, t_n)) = f(\sigma(t_1), \dots, \sigma(t_n))$ for a function symbol $f \in \mathcal{F}$
- ▶ $\sigma(p(t_1, \dots, t_n)) = p(\sigma(t_1), \dots, \sigma(t_n))$ for a predicate symbol $p \in \mathcal{P}$
- ▶ $\sigma(A \wedge B) = \sigma(A) \wedge \sigma(B)$ for formulae A, B
- ▶ ... similarly for $\neg A, A \vee B, A \rightarrow B$...
- ▶ $\sigma(\exists x A) = \exists x \sigma_x(A)$, $\sigma(\forall x A) = \forall x \sigma_x(A)$

where we define σ_x by: $\sigma_x(x) = x$, and $\sigma_x(y) = \sigma(y)$ for all $y \neq x$

Outline

- ▶ Motivation
- ▶ Syntax
- ▶ Variables
- ▶ Semantics
- ▶ The Substitution Lemma
- ▶ Satisfiability & Validity
- ▶ LK for First-order Logic
- ▶ Summary

Semantics — Interpretation

An **interpretation** assigns concrete objects, functions and relations to constant symbols, function symbols, and predicate symbols.

Definition 4.1 (Interpretation/Structure).

An **interpretation** (or **structure**) $\mathcal{I} = (D, \iota)$ consists of the following elements:

- ▶ **Domain** D is a non-empty set
- ▶ **Interpretation of constant symbols** assigns each constant $a \in \mathcal{A}$ an element $a^\iota \in D$
- ▶ **Interpretation of function symbols** assigns each n -ary function symbol $f \in \mathcal{F}$ with $n > 0$ a function $f^\iota : D^n \rightarrow D$
- ▶ **Interpretation of propositional variables** assigns each 0-ary predicate symbol $p \in \mathcal{P}$ a truth value $p^\iota \in \{T, F\}$
- ▶ **Interpretation of predicate symbols** assigns each n -ary predicate symbol $p \in \mathcal{P}$ with $n > 0$ a relation $p^\iota \subseteq D^n$

Semantics — Examples

Example: $\forall x p(a, x)$ with the interpretations

- ▶ $\mathcal{I} = (\mathbb{N}, \iota)$ with $p^\iota = \leq$ and $a^\iota = 0$
- ▶ $\mathcal{I} = (\mathbb{N}, \iota)$ with $p^\iota = \leq$ and $a^\iota = 3$
- ▶ $\mathcal{I} = (\mathbb{Z}, \iota)$ with $p^\iota = \leq$ and $a^\iota = 0$
- ▶ $\mathcal{I} = (\{c, d, e, f\}, \iota)$ with $p^\iota = \leq_{lexi}$ and $a^\iota = c$

Remark: In Ben-Ari: $(\mathbb{N}, \{\leq\}, \{0\})$, $(\mathbb{N}, \{\leq\}, \{3\})$, $(\mathbb{Z}, \{\leq\}, \{0\})$

Example: $\forall x \forall y (p(x, y) \rightarrow p(f(x, a), f(y, a)))$ with interpretations

- ▶ $\mathcal{I} = (\mathbb{Z}, \iota)$ with $p^\iota = \leq$, $f^\iota = +$, and $a^\iota = 1$
- ▶ $\mathcal{I} = (\mathbb{Z}, \iota)$ with $p^\iota = >$, $f^\iota = *$, and $a^\iota = -1$

Remark: In Ben-Ari: $(\mathbb{Z}, \{\leq\}, \{+\}, \{1\})$, $(\mathbb{Z}, \{>\}, \{*\}, \{-1\})$.

Semantics — Value of Closed Terms

Terms are evaluated according to the interpretation of their constant and function symbols.

Definition 4.2 (Term Value for Closed Terms).

Let $\mathcal{I} = (D, \iota)$ be an interpretation. The **term value** $v_{\mathcal{I}}(t)$ of a closed term $t \in \mathcal{T}$ under the interpretation \mathcal{I} is inductively defined:

- ▶ For a constant symbol $a \in \mathcal{A}$ the term value is $v_{\mathcal{I}}(a) = a^\iota$;
- ▶ Let $f \in \mathcal{F}$ be an n -ary function, $n > 0$, and t_1, \dots, t_n be terms; the term value of $f(t_1, \dots, t_n)$ is $v_{\mathcal{I}}(f(t_1, \dots, t_n)) = f^\iota(v_{\mathcal{I}}(t_1), \dots, v_{\mathcal{I}}(t_n))$

Examples:

- ▶ $f(a, f(a, b))$ with $\mathcal{I} = (\mathbb{N}, \iota)$ with $f^\iota = +$, $a^\iota = 20$, $b^\iota = 2$; then $v_{\mathcal{I}}(f(a, f(a, b))) = 42$
- ▶ $+(1, *(4, 2))$ with $\mathcal{I} = (\mathbb{Z}, \iota)$ with $+^\iota = *$ (multiplication), $*^\iota = -$ (subtraction), $1^\iota = -20$, $2^\iota = 0$, $4^\iota = 10$; then $v_{\mathcal{I}}(+(1, *(4, 2))) = -200$

Semantics — Variable Assignments, Value of Terms

The interpretation doesn't tell what to do about variables.
We need something additional.

Definition 4.3 (Variable Assignment).

Given the set of variables \mathcal{V} , and an interpretation $\mathcal{I} = (D, \iota)$, a **variable assignment** α for \mathcal{I} is a function $\alpha : \mathcal{V} \rightarrow D$.

Ben-Ari (7.18) writes this $\sigma_{\mathcal{I}\mathcal{A}}$

Definition 4.4 (Term Value).

Let $\mathcal{I} = (D, \iota)$ be an interpretation, and α an variable assignment for \mathcal{I} . The **term value** $v_{\mathcal{I}}(\alpha, t)$ of a term $t \in \mathcal{T}$ under \mathcal{I} and α is inductively defined:

- ▶ $v_{\mathcal{I}}(\alpha, x) = \alpha(x)$ for a variable $v \in \mathcal{V}$
- ▶ $v_{\mathcal{I}}(\alpha, a) = a^\iota$ for a constant symbol $a \in \mathcal{A}$
- ▶ $v_{\mathcal{I}}(\alpha, f(t_1, \dots, t_n)) = f^\iota(v_{\mathcal{I}}(\alpha, t_1), \dots, v_{\mathcal{I}}(\alpha, t_n))$ for an n -ary $f \in \mathcal{F}$

Semantics — Term value Examples

- ▶ $\mathcal{I} = (\mathbb{N}, \iota)$ with $f^\iota = +$, $a^\iota = 10$
 - ▶ $\mathcal{V} = \{x, y\}$
 - ▶ $\alpha(x) = 3 \in \mathbb{N}$ and $\alpha(y) = 5 \in \mathbb{N}$ is an assignment for \mathcal{I}
 - ▶ $v_{\mathcal{I}}(\alpha, f(a, f(a, x))) = 23$
- ▶ $\mathcal{I} = (\text{Strings}, \iota)$ with $g^\iota = \text{concatenation}$, $a^\iota = \text{"Hello"}$
 - ▶ $\mathcal{V} = \{y\}$
 - ▶ $\alpha(y) = \text{"students"}$
 - ▶ $v_{\mathcal{I}}(\alpha, f(a, f(x, a))) = \text{"HelloworldstudentsHello"}$

Semantics — Modification of an assignment

Definition 4.5 (Modification of a variable assignment).

Given an interpretation $\mathcal{I} = (D, \iota)$ and a variable assignment α for \mathcal{I} .
 Given also a variable $y \in \mathcal{V}$ and a domain element $d \in D$.
 The modified variable assignment $\alpha\{y \leftarrow d\}$ is defined as

$$\alpha\{y \leftarrow d\}(x) = \begin{cases} d & \text{if } x = y \\ \alpha(x) & \text{otherwise} \end{cases}$$

- ▶ $\mathcal{I} = (\mathbb{N}, \iota)$
- ▶ $\mathcal{V} = \{x, y\}$
- ▶ $\alpha(x) = 3 \in \mathbb{N}$ and $\alpha(y) = 5 \in \mathbb{N}$ is an assignment for \mathcal{I}
- ▶ $\alpha\{y \leftarrow 7\}(x) = 3$ and $\alpha\{y \leftarrow 7\}(y) = 7$

Semantics — Truth Value

Definition 4.6 (Truth Value).

Let $\mathcal{I} = (D, \iota)$ be an interpretation and α an assignment for \mathcal{I} . The **truth value** $v_{\mathcal{I}}(\alpha, A) \in \{T, F\}$ of a formula A under \mathcal{I} and α is defined inductively as follows:

- ▶ $v_{\mathcal{I}}(\alpha, p) = T$ for 0-ary $p \in \mathcal{P}$ iff $p^\iota = T$, otherwise $v_{\mathcal{I}}(\alpha, p) = F$
- ▶ $v_{\mathcal{I}}(\alpha, p(t_1, \dots, t_n)) = T$ for $p \in \mathcal{P}$, $n > 0$, iff $(v_{\mathcal{I}}(\alpha, t_1), \dots, v_{\mathcal{I}}(\alpha, t_n)) \in p^\iota$, otherwise $v_{\mathcal{I}}(\alpha, p(t_1, \dots, t_n)) = F$
- ▶ $v_{\mathcal{I}}(\alpha, \neg A) = T$ iff $v_{\mathcal{I}}(\alpha, A) = F$, otherwise $v_{\mathcal{I}}(\alpha, \neg A) = F$
- ▶ $v_{\mathcal{I}}(\alpha, A \wedge B) = T$ iff $v_{\mathcal{I}}(\alpha, A) = T$ and $v_{\mathcal{I}}(\alpha, B) = T$, otherwise $v_{\mathcal{I}}(\alpha, A \wedge B) = F$
- ▶ $v_{\mathcal{I}}(\alpha, A \vee B) = T$ iff $v_{\mathcal{I}}(\alpha, A) = T$ or $v_{\mathcal{I}}(\alpha, B) = T$, otherwise $v_{\mathcal{I}}(\alpha, A \vee B) = F$
- ▶ $v_{\mathcal{I}}(\alpha, A \rightarrow B) = T$ iff $v_{\mathcal{I}}(\alpha, A) = F$ or $v_{\mathcal{I}}(\alpha, B) = T$, otherwise $v_{\mathcal{I}}(\alpha, A \rightarrow B) = F$
- ▶ $v_{\mathcal{I}}(\alpha, \forall x A) = T$ iff $v_{\mathcal{I}}(\alpha\{x \leftarrow d\}, A) = T$ for all $d \in D$, otherwise $v_{\mathcal{I}}(\alpha, \forall x A) = F$
- ▶ $v_{\mathcal{I}}(\alpha, \exists x A) = T$ iff $v_{\mathcal{I}}(\alpha\{x \leftarrow d\}, A) = T$ for some $d \in D$, otherwise $v_{\mathcal{I}}(\alpha, \exists x A) = F$
- ▶ $v_{\mathcal{I}}(\alpha, \top) = T$ and $v_{\mathcal{I}}(\alpha, \perp) = F$

Semantics — Truth Value

Theorem 4.1 (Value of closed formulae).

For a **closed** term or formula, the assignment has no influence on the term value or truth value. $v_{\mathcal{I}}$ for $v_{\mathcal{I}, \alpha}$.

Example: $A = \forall x p(a, x)$ with the interpretations

- ▶ $\mathcal{I} = (\mathbb{N}, \iota)$ with $p^\iota = \leq$ and $a^\iota = 0 \rightsquigarrow v_{\mathcal{I}}(A) = T$
- ▶ $\mathcal{I} = (\mathbb{N}, \iota)$ with $p^\iota = \leq$ and $a^\iota = 3 \rightsquigarrow v_{\mathcal{I}}(A) = F$
- ▶ $\mathcal{I} = (\mathbb{Z}, \iota)$ with $p^\iota = \leq$ and $a^\iota = 0 \rightsquigarrow v_{\mathcal{I}}(A) = F$
- ▶ $\mathcal{I} = (\{c, d, e, f\}, \iota)$ with $p^\iota = \leq_{\text{lexi}}$ and $a^\iota = c \rightsquigarrow v_{\mathcal{I}}(A) = T$

Example: $B = \forall x \forall y (p(x, y) \rightarrow p(f(x, a), f(y, a)))$ with interpretations

- ▶ $\mathcal{I} = (\mathbb{Z}, \iota)$ with $p^\iota = \leq$, $f^\iota = +$, and $a^\iota = 1 \rightsquigarrow v_{\mathcal{I}}(B) = T$
- ▶ $\mathcal{I} = (\mathbb{Z}, \iota)$ with $p^\iota = >$, $f^\iota = *$, and $a^\iota = -1 \rightsquigarrow v_{\mathcal{I}}(B) = F$

Outline

- ▶ Motivation
- ▶ Syntax
- ▶ Variables
- ▶ Semantics
- ▶ The Substitution Lemma
- ▶ Satisfiability & Validity
- ▶ LK for First-order Logic
- ▶ Summary

The Substitution Lemma for Terms

Theorem 5.1 (Substitution Lemma for Terms).

Given an interpretation $\mathcal{I} = (D, \iota)$ and a variable assignment α for \mathcal{I} .
Given also a variable $y \in \mathcal{V}$, and terms $t, s \in \mathcal{T}$

$$v_{\mathcal{I}}(\alpha, t[y \setminus s]) = v_{\mathcal{I}}(\alpha\{y \leftarrow v_{\mathcal{I}}(\alpha, s)\}, t)$$

Proof.

By structural induction on t . We abbreviate: $\alpha' := \alpha\{y \leftarrow v_{\mathcal{I}}(\alpha, s)\}$

For a constant a , $a[y \setminus s] = a$, so $v_{\mathcal{I}}(\alpha, a[y \setminus s]) = v_{\mathcal{I}}(\alpha, a) = a' = v_{\mathcal{I}}(\alpha', a)$

For a variable $x \neq y$, $x[y \setminus s] = x$, so

$$v_{\mathcal{I}}(\alpha, x[y \setminus s]) = v_{\mathcal{I}}(\alpha, x) = \alpha(x) = \alpha'(x) = v_{\mathcal{I}}(\alpha', x) \quad \square$$

Proof of substitution lemma, continued

Proof.

For the variable y , $y[y \setminus s] = s$, so

$$v_{\mathcal{I}}(\alpha, y[y \setminus s]) = v_{\mathcal{I}}(\alpha, s) = v_{\mathcal{I}}(\alpha\{y \leftarrow v_{\mathcal{I}}(\alpha, s)\}, y)$$

For a complex term, $f(\dots t_i \dots)[y \setminus s] = f(\dots t_i[y \setminus s] \dots)$, so

$$\begin{aligned} & v_{\mathcal{I}}(\alpha, f(\dots t_i \dots)[y \setminus s]) \\ &= v_{\mathcal{I}}(\alpha, f(\dots t_i[y \setminus s] \dots)) \quad \text{by def. of substitution} \\ &= f^{\iota}(\dots v_{\mathcal{I}}(\alpha, t_i[y \setminus s]) \dots) \quad \text{by model semantics} \\ &= f^{\iota}(\dots v_{\mathcal{I}}(\alpha', t_i) \dots) \quad \text{by the induction hypothesis} \\ &= v_{\mathcal{I}}(\alpha', f(\dots t_i \dots)) \quad \text{by model semantics} \end{aligned}$$

□

The Substitution Lemma for Formulae

Theorem 5.2 (Substitution Lemma for Terms).

Given an interpretation $\mathcal{I} = (D, \iota)$ and a variable assignment α for \mathcal{I} .
Given also a variable $y \in \mathcal{V}$, a formula A and a term $s \in \mathcal{T}$, such that $\{y \setminus s\}$ is collision-free for A .

$$v_{\mathcal{I}}(\alpha, A[y \setminus s]) = v_{\mathcal{I}}(\alpha', A)$$

Outline

- ▶ Motivation
- ▶ Syntax
- ▶ Variables
- ▶ Semantics
- ▶ The Substitution Lemma
- ▶ Satisfiability & Validity
- ▶ LK for First-order Logic
- ▶ Summary

Satisfiability and Validity

Definition 6.1 (Satisfiable, Model, Unsatisfiable, Valid, Invalid).

Let A be a **closed** (first-order) formula and $U = \{A_1, \dots\}$ be a set of **closed** (first-order) formulae A_i .

- ▶ A is **satisfiable** iff $v_{\mathcal{I}}(A) = T$ for some interpretation \mathcal{I} .
- ▶ A satisfying interpretation \mathcal{I} for A is called a **model** for A .
- ▶ $U = \{A_1, \dots\}$ is **satisfiable** iff there is (common) model for all A_i .
- ▶ A/U is **unsatisfiable** iff A/U is **not** satisfiable.
- ▶ A is **valid**, denoted $\models A$, iff $v_{\mathcal{I}}(A) = T$ for all interpretations \mathcal{I} .
- ▶ A is **invalid/falsifiable** iff A is **not** valid.

Theorem 6.1 (Satisfiable, Valid, Unsatisfiable, Invalid).

A is **valid** iff $\neg A$ is **unsatisfiable**. A is **satisfiable** iff $\neg A$ is **invalid**.

Examples for Satisfiable and Invalid Formulae

Example: $A = \forall x p(a, x)$

- ▶ $\mathcal{I} = (\mathbb{N}, \iota)$ with $p^{\iota} = \leq$ and $a^{\iota} = 3 \rightsquigarrow v_{\mathcal{I}}(A) = F$
 $\rightsquigarrow A$ is invalid
- ▶ $\mathcal{I} = (\{c, d, e, f\}, \iota)$ with $p^{\iota} = \leq_{lexi}$ and $a^{\iota} = c \rightsquigarrow v_{\mathcal{I}}(A) = T$
 $\rightsquigarrow A$ is satisfiable (\mathcal{I} is a model)

Example: $B = \forall x \forall y (p(x, y) \rightarrow p(f(x, a), f(y, a)))$

- ▶ $\mathcal{I} = (\mathbb{Z}, \iota)$ with $p^{\iota} = \leq$, $f^{\iota} = +$, and $a^{\iota} = 1 \rightsquigarrow v_{\mathcal{I}}(B) = T$
 \rightsquigarrow satisfiable (\mathcal{I} is a model)
- ▶ $\mathcal{I} = (\mathbb{Z}, \iota)$ with $p^{\iota} = >$, $f^{\iota} = *$, and $a^{\iota} = -1 \rightsquigarrow v_{\mathcal{I}}(B) = F$
 \rightsquigarrow invalid (\mathcal{I} is a “counter-model”)

Example: $\forall x \forall y (p(x, y) \rightarrow p(y, x))$

\rightsquigarrow satisfiable (e.g. $p^{\iota} = =$), but invalid (e.g. $p^{\iota} = <$)

Example: $\exists x \exists y (p(x) \wedge \neg p(y))$

\rightsquigarrow only satisfiable for $|D| \geq 2$, invalid (e.g. $D = \mathbb{N}$, $p^{\iota} = \text{even}$)

Logical Equivalence

The concept of **logical equivalence** can be adapted to first-order logic, i.e. to closed first-order formulae.

Definition 6.2 (Logical Equivalence).

Let A_1, A_2 be two closed formulae. A_1 is **logically equivalent** to A_2 , denoted $A_1 \equiv A_2$ iff $v_{\mathcal{I}}(A_1) = v_{\mathcal{I}}(A_2)$ for all interpretations \mathcal{I} .

Theorem 6.2 (Relation \equiv and \leftrightarrow).

Let A, B be two closed formulae and $U = \{A_1, \dots, A_n\}$ be a set of closed formulas. Then $A \equiv B$ iff $\models A \leftrightarrow B$.

Remark: $A \leftrightarrow B := (A \rightarrow B) \wedge (B \rightarrow A)$

Important: even though \equiv and \leftrightarrow are closely related, they are different relations. Whereas \leftrightarrow is part of the object language (i.e. the definition of formulae), \equiv is used in the meta-language to talk about or relate formulae.

Logically Equivalent Formulae

Duality: \forall can be expressed with \exists , and vice versa

- ▶ $\models \forall x A(x) \leftrightarrow \neg \exists x \neg A(x)$
- ▶ $\models \exists x A(x) \leftrightarrow \neg \forall x \neg A(x)$

Commutativity:

- ▶ $\models \forall x \forall y A(x, y) \leftrightarrow \forall y \forall x A(x, y)$
- ▶ $\models \exists x \exists y A(x, y) \leftrightarrow \exists y \exists x A(x, y)$
- ▶ $\models \exists x \forall y A(x, y) \rightarrow \forall y \exists x A(x, y)$ (other direction is not valid!)

Distributivity:

- ▶ $\models \exists x (A(x) \vee B(x)) \leftrightarrow \exists x A(x) \vee \exists x B(x)$
- ▶ $\models \forall x (A(x) \wedge B(x)) \leftrightarrow \forall x A(x) \wedge \forall x B(x)$
- ▶ $\models \forall x A(x) \vee \forall x B(x) \rightarrow \forall x (A(x) \vee B(x))$ (other direction not valid!)
- ▶ $\models \exists x (A(x) \wedge B(x)) \rightarrow \exists x A(x) \wedge \exists x B(x)$ (other direction not valid!)

See [Ben-Ari 2012] for **more equivalences** involving quantifiers.

Logical Consequence

Definition 6.3 (Logical Consequence).

Let A be a closed formula and U be a set of closed formulae. A is a **logical consequence** of U , denoted $U \models A$, iff every model of U is a model of A , i.e. $v_{\mathcal{I}}(A_i) = T$ for all $A_i \in U$ implies $v_{\mathcal{I}}(A) = T$.

Theorem 6.3 (Logical Consequence and Validity).

Let A be a closed formula and $U = \{A_1, \dots, A_n\}$ be a set of closed formulae. Then $U \models A$ iff $\models (A_1 \wedge \dots \wedge A_n) \rightarrow A$.

- ▶ again, we can **reduce** the problem of “logical consequence” to the problem of determining if a formula is **valid**
- ▶ hence, we need methods or **proof search calculi** that can deal with **first-order formulae**

Outline

- ▶ Motivation
- ▶ Syntax
- ▶ Variables
- ▶ Semantics
- ▶ The Substitution Lemma
- ▶ Satisfiability & Validity
- ▶ LK for First-order Logic
- ▶ Summary

LK — Axiom and Propositional Rules

- ▶ **axiom**
$$\frac{}{\Gamma, A \Rightarrow A, \Delta} \text{ axiom}$$
- ▶ **rules for \wedge (conjunction)**

$$\frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta} \wedge\text{-left} \quad \frac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta} \wedge\text{-right}$$
- ▶ **rules for \vee (disjunction)**

$$\frac{\Gamma, A \Rightarrow \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \vee B \Rightarrow \Delta} \vee\text{-left} \quad \frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \vee B, \Delta} \vee\text{-right}$$
- ▶ **rules for \rightarrow (implication)**

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \rightarrow B \Rightarrow \Delta} \rightarrow\text{-left} \quad \frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \rightarrow B, \Delta} \rightarrow\text{-right}$$
- ▶ **rules for \neg (negation)**

$$\frac{\Gamma \Rightarrow A, \Delta}{\Gamma, \neg A \Rightarrow \Delta} \neg\text{-left} \quad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \neg A, \Delta} \neg\text{-right}$$

LK — Rules for Universal and Existential Quantifier

► rules for \forall (universal quantifier)

$$\frac{\Gamma, A[x \setminus t], \forall x A \implies \Delta}{\Gamma, \forall x A \implies \Delta} \forall\text{-left} \quad \frac{\Gamma \implies A[x \setminus a], \Delta}{\Gamma \implies \forall x A, \Delta} \forall\text{-right}^*$$

- t is an arbitrary term
- **Eigenvariable condition** for the rule $\forall\text{-right}^*$: a must not occur in the conclusion, i.e. in Γ , Δ , or A
- the formula $\forall x A$ is preserved in the premise of the rule $\forall\text{-left}$

► rules for \exists (existential quantifier)

$$\frac{\Gamma, A[x \setminus a] \implies \Delta}{\Gamma, \exists x A \implies \Delta} \exists\text{-left}^* \quad \frac{\Gamma \implies \exists x A, A[x \setminus t], \Delta}{\Gamma \implies \exists x A, \Delta} \exists\text{-right}$$

- t is an arbitrary term
- **Eigenvariable condition** for the rule $\exists\text{-left}^*$: a must not occur in the conclusion, i.e. in Γ , Δ , or A
- the formula $\exists x A$ is preserved in the premise of the rule $\exists\text{-right}$

Soundness and Completeness

Theorem 7.1 (Soundness and Completeness of LK).

The calculus of natural deduction LK is sound and complete, i.e.

- if A is provable in LK, then A is valid (if $\vdash A$ then $\models A$)
- if A is valid, then A is provable in LK (if $\models A$ then $\vdash A$)

Proof.

Next week. □

Examples of LK Proofs

Example: $\forall x p(x) \rightarrow \exists x p(x)$

$$\frac{\frac{\frac{\overline{p(c), \forall x p(x) \implies p(c), \exists x p(x)} \text{ axiom}}{\overline{p(c), \forall x p(x) \implies \exists x p(x)}} \exists\text{-right}}{\forall x p(x) \implies \exists x p(x)} \forall\text{-left}}{\implies \forall x p(x) \rightarrow \exists x p(x)} \rightarrow\text{-right}$$

Example: $p(a) \rightarrow p(b)$

$$\frac{}{\implies p(a) \rightarrow p(b)} (?)$$

Example: $p(a) \rightarrow \exists x p(x)$

$$\frac{\frac{\frac{\overline{p(a) \implies p(a), \exists x p(x)} \text{ axiom}}{\overline{p(a) \implies \exists x p(x)}} \exists\text{-right}}{\implies p(a) \rightarrow \exists x p(x)} \rightarrow\text{-right}}$$

Example: $\exists x p(x) \rightarrow p(a)$

$$\frac{\frac{\overline{\exists x p(x) \implies p(a)} \exists\text{-left}^*}{\implies \exists x p(x) \rightarrow p(a)} \rightarrow\text{-right}}$$

rule $\exists\text{-left}^*$ with $p(x)[x \setminus a]$ **cannot** be applied as a occurs in the premise (Eigenvariable condition!)

Outline

- Motivation
- Syntax
- Variables
- Semantics
- The Substitution Lemma
- Satisfiability & Validity
- LK for First-order Logic
- Summary

Summary

- ▶ **first-order logic** extends the **syntax** of propositional logic by: **constants**, **functions**, **variables**, **predicates**, and the **quantifiers** \forall/\exists
- ▶ the **semantics** consists of a **domain** D and an **interpretation** ι
- ▶ the **interpretation** ι relates constants to elements of the domain, function symbols to actual functions, and predicates to relations
- ▶ variables are interpreted by a **variable assignment** α
- ▶ the formula $\forall x p(x)/\exists x p(x)$ evaluates to T iff $p(x)$ evaluates to T for **all/some** element(s) in D
- ▶ the **truth value** of formulae is inductively evaluated, and takes the **value of terms** into account
- ▶ most concepts from propositional logic can be adapted
- ▶ four **semantical concepts**: satisfiable, valid, unsatisfiable, invalid
- ▶ **logical consequence** in first-order logic can be reduced to **validity**
- ▶ **Next week**: Soundness and completeness