Logic for Multiagent Systems

Master 1st Year, 1st Semester 201/2022

Laurențiu Leuștean

Web page: http://cs.unibuc.ro/~lleustean/

Propositional logic

Language

Definition 1.1

The language of propositional logic PL consists of:

- *a countable set $V = \{v_n \mid n \in \mathbb{N}\}$ of variables;*
- *the logic connectives $\neg$ (non), $\to$ (implies)*
- *parantheses: (,).*

- The set *Sym* of symbols of *PL* is

$$Sym := V \cup \{\neg, \to, (,)\}.$$

- We denote variables by $u, v, x, y, z \ldots$

Language

Definition 1.2

The set Expr of expressions of PL is the set of all finite sequences of symbols of PL.

Definition 1.3

Let $\theta = \theta_0\theta_1 \ldots \theta_{k-1}$ be an expression, where $\theta_i \in Sym$ for all $i = 0, \ldots, k-1$.

- *If $0 \leq i \leq j \leq k-1$, then the expression $\theta_i \ldots \theta_j$ is called the (i,j)-subexpression of θ.*
- *We say that an expression ψ appears in θ if there exists $0 \leq i \leq j \leq k-1$ such that ψ is the (i,j)-subexpression of θ.*
- *We denote by $Var(\theta)$ the set of variables appearing in θ.*

Language

The definition of formulas is an example of an inductive definition.

Definition 1.4

The formulas of PL are the expressions of PL defined as follows:

- *(F0)* *Any variable is a formula.*
- *(F1)* *If φ is a formula, then $(\neg\varphi)$ is a formula.*
- *(F2)* *If φ and ψ are formulas, then $(\varphi \to \psi)$ is a formula.*
- *(F3)* *Only the expressions obtained by applying rules (F0), (F1), (F2) are formulas.*

Notations

The set of formulas is denoted by *Form*. Formulas are denoted by $\varphi, \psi, \chi, \ldots$.

Proposition 1.5

The set Form is countable.

Language

Unique readability

If φ is a formula, then exactly one of the following hold:

- $\varphi = v$, where $v \in V$.
- $\varphi = (\neg\psi)$, where ψ is a formula.
- $\varphi = (\psi \to \chi)$, where ψ, χ are formulas.

Furthermore, φ can be written in a unique way in one of these forms.

Definition 1.6

Let φ be a formula. A subformula of φ is any formula ψ that appears in φ.

Language

Proposition 1.7 (Induction principle on formulas)

Let Γ be a set of formulas satisfying the following properties:

- *$V \subseteq \Gamma$.*
- *Γ is closed to $\neg$, that is: $\varphi \in \Gamma$ implies $(\neg\varphi) \in \Gamma$.*
- *Γ is closed to $\to$, that is: $\varphi, \psi \in \Gamma$ implies $(\varphi \to \psi) \in \Gamma$.*

Then $\Gamma = Form$.

It is used to prove that all formulas have a property $\mathcal{P}$: we define Γ as the set of all formulas satisfying $\mathcal{P}$ and apply induction on formulas to obtain that $\Gamma = Form$.

Language

The derived connectives $\vee$ (or), $\wedge$ (and), $\leftrightarrow$ (if and only if) are introduced by the following abbreviations:

$$\begin{aligned} \varphi \vee \psi &:= ((\neg\varphi) \to \psi) \\ \varphi \wedge \psi &:= \neg(\varphi \to (\neg\psi))) \\ \varphi \leftrightarrow \psi &:= ((\varphi \to \psi) \wedge (\psi \to \varphi)) \end{aligned}$$

Conventions and notations

- The external parantheses are omitted, we put them only when necessary. We write $\neg\varphi$, $\varphi \to \psi$, but we write $(\varphi \to \psi) \to \chi$.
- To reduce the use of parentheses, we assume that
  - $\neg$ has higher precedence than $\to, \wedge, \vee, \leftrightarrow$;
  - $\wedge, \vee$ have higher precedence than $\to, \leftrightarrow$.
- Hence, the formula $(((\varphi \to (\psi \vee \chi)) \wedge ((\neg\psi) \leftrightarrow (\psi \vee \chi)))$ is written as $(\varphi \to \psi \vee \chi) \wedge (\neg\psi \leftrightarrow \psi \vee \chi)$.

Semantics

Truth values

We use the following notations for the truth values:

1 for true and 0 for false.

Hence, the set of truth values is $\{0,1\}$.

Define the following operations on $\{0,1\}$ using truth tables.

$\neg : \{0,1\} \to \{0,1\}$,

p	$\neg p$
0	1
1	0

$\rightarrow : \{0,1\} \times \{0,1\} \to \{0,1\}$,

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Semantics

$\vee : \{0,1\} \times \{0,1\} \to \{0,1\}$,

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

$\wedge : \{0,1\} \times \{0,1\} \to \{0,1\}$,

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

$\leftrightarrow : \{0,1\} \times \{0,1\} \to \{0,1\}$,

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

Semantics

Definition 1.8

An evaluation (or interpretation) is a function $e : V \to \{0,1\}$.

Theorem 1.9

For any evaluation $e : V \to \{0,1\}$ there exists a unique function

$$e^+ : Form \to \{0,1\}$$

satisfying the following properties:

- $e^+(v) = e(v)$ *for all* $v \in V$.
- $e^+(\neg\varphi) = \neg e^+(\varphi)$ *for any formula* φ.
- $e^+(\varphi \to \psi) = e^+(\varphi) \rightarrow e^+(\psi)$ *for any formulas* φ, ψ.

Proposition 1.10

For any formula φ and all evaluations $e_1, e_2 : V \to \{0,1\}$,

if $e_1(v) = e_2(v)$ for all $v \in Var(\varphi)$, then $e_1^+(\varphi) = e_2^+(\varphi)$.

Semantics

Let φ be a formula.

Definition 1.11

- *An evaluation $e : V \to \{0,1\}$ is a model of φ if $e^+(\varphi) = 1$.*
  Notation: $e \vDash \varphi$.
- *φ is satisfiable if it has a model.*
- *If φ is not satisfiable, we also say that φ is unsatisfiable or contradictory.*
- *φ is a tautology if every evaluation is a model of φ.*
  Notation: $\vDash \varphi$.

Notation 1.12

The set of models of φ is denoted by $Mod(\varphi)$.

Remark 1.13

- φ is a tautology iff $\neg\varphi$ is unsatisfiable.
- φ is unsatisfiable iff $\neg\varphi$ is a tautology.

Proposition 1.14

Let $e : V \to \{0,1\}$ be an evaluation. Then for all formulas φ, ψ,

- $e \models \neg\varphi$ iff $e \not\models \varphi$.
- $e \models \varphi \to \psi$ iff ($e \models \varphi$ implies $e \models \psi$) iff ($e \not\models \varphi$ or $e \models \psi$).
- $e \models \varphi \vee \psi$ iff ($e \models \varphi$ or $e \models \psi$).
- $e \models \varphi \wedge \psi$ iff ($e \models \varphi$ and $e \models \psi$).
- $e \models \varphi \leftrightarrow \psi$ iff ($e \models \varphi$ iff $e \models \psi$).

Definition 1.15

Let φ, ψ be formulas. We say that

- φ is a semantic consequence of ψ if $Mod(\psi) \subseteq Mod(\varphi)$.
  Notation: $\psi \models \varphi$.
- φ and ψ are (logically) equivalent if $Mod(\psi) = Mod(\varphi)$.
  Notation: $\varphi \sim \psi$.

Remark 1.16

Let φ, ψ be formulas.

- $\psi \models \varphi$ iff $\models \psi \to \varphi$.
- $\psi \sim \varphi$ iff ($\psi \models \varphi$ and $\varphi \models \psi$) iff $\models \psi \leftrightarrow \varphi$.

For all formulas φ, ψ, χ,

$$\begin{aligned}
&\models \varphi \vee \neg\varphi \\
&\models \neg(\varphi \wedge \neg\varphi) \\
&\models \varphi \wedge \psi \to \varphi \\
&\models \varphi \to \varphi \vee \psi \\
&\models \varphi \to (\psi \to \varphi) \\
&\models (\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \\
&\models (\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi)) \\
&\models (\varphi \to \psi) \vee (\neg\varphi \to \psi) \\
&\models (\varphi \to \psi) \vee (\varphi \to \neg\psi) \\
&\models \neg\varphi \to (\neg\psi \leftrightarrow (\psi \to \varphi)) \\
&\models (\varphi \to \psi) \to (((\varphi \to \chi) \to \psi) \to \psi) \\
&\models \neg\psi \to (\psi \to \varphi)
\end{aligned}$$

$$\begin{aligned}
&\models \psi \to (\neg\psi \to \varphi) \\
&\models (\varphi \to \neg\varphi) \to \neg\varphi \\
&\models (\neg\varphi \to \varphi) \to \varphi \\
\psi &\models \varphi \to \psi \\
\neg\varphi &\models \varphi \to \psi \\
\neg\psi \wedge (\varphi \to \psi) &\models \neg\varphi \\
(\varphi \to \psi) \wedge (\psi \to \chi) &\models \varphi \to \chi \\
\varphi \wedge (\varphi \to \psi) &\models \psi \\
\{\psi, \neg\psi\} &\models \varphi \\
\{\psi, \neg\varphi\} &\models \neg(\psi \to \varphi)
\end{aligned}$$

Semantics

$$
\begin{aligned}
\varphi &\sim \neg\neg\varphi \\
\varphi \to \psi &\sim \neg\psi \to \neg\varphi \\
\varphi \vee \psi &\sim \neg(\neg\varphi \wedge \neg\psi) \\
\varphi \wedge \psi &\sim \neg(\neg\varphi \vee \neg\psi) \\
\varphi \to (\psi \to \chi) &\sim \varphi \wedge \psi \to \chi \\
\varphi \sim \varphi \wedge \varphi &\sim \varphi \vee \varphi \\
\varphi \wedge \psi &\sim \psi \wedge \varphi \\
\varphi \vee \psi &\sim \psi \vee \varphi \\
\varphi \wedge (\psi \wedge \chi) &\sim (\varphi \wedge \psi) \wedge \chi \\
\varphi \vee (\psi \vee \chi) &\sim (\varphi \vee \psi) \vee \chi \\
\varphi \vee (\varphi \wedge \psi) &\sim \varphi \\
\varphi \wedge (\varphi \vee \psi) &\sim \varphi
\end{aligned}
$$

Semantics

$$
\begin{aligned}
\varphi \wedge (\psi \vee \chi) &\sim (\varphi \wedge \psi) \vee (\varphi \wedge \chi) \\
\varphi \vee (\psi \wedge \chi) &\sim (\varphi \vee \psi) \wedge (\varphi \vee \chi) \\
\varphi \to \psi \wedge \chi &\sim (\varphi \to \psi) \wedge (\varphi \to \chi) \\
\varphi \to \psi \vee \chi &\sim (\varphi \to \psi) \vee (\varphi \to \chi) \\
\varphi \wedge \psi \to \chi &\sim (\varphi \to \chi) \vee (\psi \to \chi) \\
\varphi \vee \psi \to \chi &\sim (\varphi \to \chi) \wedge (\psi \to \chi) \\
\varphi \to (\psi \to \chi) &\sim \psi \to (\varphi \to \chi) \\
&\sim (\varphi \to \psi) \to (\varphi \to \chi) \\
\neg\varphi \sim \varphi \to \neg\varphi &\sim (\varphi \to \psi) \wedge (\varphi \to \neg\psi) \\
\varphi \to \psi \sim \neg\varphi \vee \psi &\sim \neg(\varphi \wedge \neg\psi) \\
\varphi \vee \psi \sim \varphi \vee (\neg\varphi \wedge \psi) &\sim (\varphi \to \psi) \to \psi \\
\varphi \leftrightarrow (\psi \leftrightarrow \chi) &\sim (\varphi \leftrightarrow \psi) \leftrightarrow \chi
\end{aligned}
$$

Semantics

It is often useful to have a canonical tautology and a canonical unsatisfiable formula.

Remark 1.17

$v_0 \to v_0$ is a tautology and $\neg(v_0 \to v_0)$ is unsatisfiable.

Notation 1.18

Denote $v_0 \to v_0$ by $\top$ and call it the truth.
Denote $\neg(v_0 \to v_0)$ by $\bot$ and call it the false.

Remark 1.19

- *φ is a tautology iff $\varphi \sim \top$.*
- *φ is unsatisfiable iff $\varphi \sim \bot$.*

Semantics

Let Γ be a set of formulas.

Definition 1.20

An evaluation $e : V \to \{0, 1\}$ is a model of Γ if it is a model of every formula from Γ.
Notation: $e \vDash \Gamma$.

Notation 1.21

The set of models of Γ is denoted by $Mod(\Gamma)$.

Definition 1.22

A formula φ is a semantic consequence of Γ if $Mod(\Gamma) \subseteq Mod(\varphi)$.
Notation: $\Gamma \vDash \varphi$.

Semantics

Definition 1.23

- Γ *is satisfiable if it has a model.*
- Γ *is finitely satisfiable if every finite subset of* Γ *is satisfiable.*
- *If* Γ *is not satisfiable, we say also that* Γ *is unsatisfiable or contradictory.*

Proposition 1.24

The following are equivalent:

- Γ *is unsatisfiable.*
- $\Gamma \models \bot$.

Theorem 1.25 (Compactness Theorem)

Γ *is satisfiable iff* Γ *is finitely satisfiable.*

Syntax

We use a deductive system of Hilbert type for *LP*.

Logical axioms

The set *Axm* of (logical) axioms of *LP* consists of:

(A1) $\varphi \to (\psi \to \varphi)$

(A2) $(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$

(A3) $(\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$,

where φ, ψ and χ are formulas.

The deduction rule

For any formulas φ, ψ, from φ and $\varphi \to \psi$ infer ψ (modus ponens or (MP)):

$$\frac{\varphi,\ \varphi \to \psi}{\psi}$$

Syntax

Let Γ be a set of formulas. The definition of Γ-theorems is another example of an inductive definition.

Definition 1.26

The Γ*-theorems of PL are the formulas defined as follows:*

(T0) *Every logical axiom is a* Γ*-theorem.*

(T1) *Every formula of* Γ *is a* Γ*-theorem.*

(T2) *If* φ *and* $\varphi \to \psi$ *are* Γ*-theorems, then* ψ *is a* Γ*-theorem.*

(T3) *Only the formulas obtained by applying rules (T0), (T1), (T2) are* Γ*-theorems.*

If φ is a Γ-theorem, then we also say that φ is deduced from the hypotheses Γ.

Syntax

Notations

$\Gamma \vdash \varphi \quad :\Leftrightarrow \quad \varphi$ is a Γ-theorem

$\vdash \varphi \quad :\Leftrightarrow \quad \emptyset \vdash \varphi$.

Definition 1.27

A formula φ *is called a theorem of LP if* $\vdash \varphi$.

By a reformulation of the conditions (T0), (T1), (T2) using the notation $\vdash$, we get

Remark 1.28

- *If* φ *is an axiom, then* $\Gamma \vdash \varphi$.
- *If* $\varphi \in \Gamma$*, then* $\Gamma \vdash \varphi$.
- *If* $\Gamma \vdash \varphi$ *and* $\Gamma \vdash \varphi \to \psi$*, then* $\Gamma \vdash \psi$.

Syntax

Definition 1.29

A Γ-proof (or proof from the hypotheses Γ) is a sequence of formulas $\theta_1, \ldots, \theta_n$ such that for all $i \in \{1, \ldots, n\}$, one of the following holds:

- *θ_i is an axiom.*
- *$\theta_i \in \Gamma$.*
- *there exist $k, j < i$ such that $\theta_k = \theta_j \to \theta_i$.*

Definition 1.30

Let φ be a formula. A Γ-proof of φ or a proof of φ from the hypotheses Γ is a Γ-proof $\theta_1, \ldots, \theta_n$ such that $\theta_n = \varphi$.

Proposition 1.31

For any formula φ,

$$\Gamma \vdash \varphi \quad \text{iff} \quad \text{there exists a } \Gamma\text{-proof of } \varphi.$$

Syntax

Theorem 1.32 (Deduction Theorem)

Let $\Gamma \cup \{\varphi, \psi\}$ be a set of formulas. Then

$$\Gamma \cup \{\varphi\} \vdash \psi \quad \text{iff} \quad \Gamma \vdash \varphi \to \psi.$$

Proposition 1.33

For any formulas φ, ψ, χ,

$$\vdash (\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$$
$$\vdash (\varphi \to (\psi \to \chi)) \to (\psi \to (\varphi \to \chi))$$

Proposition 1.34

Let $\Gamma \cup \{\varphi, \psi, \chi\}$ be a set of formulas.

$$\Gamma \vdash \varphi \to \psi \text{ and } \Gamma \vdash \psi \to \chi \quad \Rightarrow \quad \Gamma \vdash \varphi \to \chi$$
$$\Gamma \cup \{\neg\psi\} \vdash \neg(\varphi \to \varphi) \quad \Rightarrow \quad \Gamma \vdash \psi$$
$$\Gamma \cup \{\psi\} \vdash \varphi \text{ and } \Gamma \cup \{\neg\psi\} \vdash \varphi \quad \Rightarrow \quad \Gamma \vdash \varphi.$$

Consistent sets

Let Γ be a set of formulas.

Definition 1.35

Γ is called consistent if there exists a formula φ such that $\Gamma \nvdash \varphi$. Γ is said to be inconsistent if it is not consistent, that is $\Gamma \vdash \varphi$ for any formula φ.

Proposition 1.36

- *$\emptyset$ is consistent.*
- *The set of theorems is consistent.*

Proposition 1.37

The following are equivalent:

- *Γ is inconsistent.*
- *$\Gamma \vdash \bot$.*

Completeness Theorem

Theorem 1.38 (Completeness Theorem (version 1))

Let Γ be a set of formulas. Then

$$\Gamma \text{ is consistent} \quad \Longleftrightarrow \quad \Gamma \text{ is satisfiable.}$$

Theorem 1.39 (Completeness Theorem (version 2))

Let Γ be a set of formulas. For any formula φ,

$$\Gamma \vdash \varphi \quad \Longleftrightarrow \quad \Gamma \vDash \varphi.$$

First-order logic

First-order languages

Definition 2.1

A first-order language $\mathcal{L}$ consists of:

- *a countable set $V = \{v_n \mid n \in \mathbb{N}\}$ of variables;*
- *the connectives $\neg$ and $\rightarrow$;*
- *parantheses (,);*
- *the equality symbol $=$;*
- *the universal quantifier $\forall$;*
- *a set $\mathcal{R}$ of relation symbols;*
- *a set $\mathcal{F}$ of function symbols;*
- *a set $\mathcal{C}$ of constant symbols;*
- *an arity function $ari : \mathcal{F} \cup \mathcal{R} \rightarrow \mathbb{N}^*$.*

- $\mathcal{L}$ is uniquely determined by the quadruple $\tau := (\mathcal{R}, \mathcal{F}, \mathcal{C}, \mathrm{ari})$.
- τ is called the signature of $\mathcal{L}$ or the similaritaty type of $\mathcal{L}$.

First-order languages

Let $\mathcal{L}$ be a first-order language.

- The set $Sym_{\mathcal{L}}$ of symbols of $\mathcal{L}$ is

$$Sym_{\mathcal{L}} := V \cup \{\neg, \rightarrow, (,), =, \forall\} \cup \mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$$

- The elements of $\mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$ are called non-logical symbols.
- The elements of $V \cup \{\neg, \rightarrow, (,), =, \forall\}$ are called logical symbols.

- We denote variables by $x, y, z, v, \ldots$, relation symbols by $P, Q, R \ldots$, function symbols by $f, g, h, \ldots$ and constant symbols by $c, d, e, \ldots$.

- For every $m \in \mathbb{N}^*$ we denote:

$\mathcal{F}_m$:= the set of function symbols of arity m;

$\mathcal{R}_m$:= the set of relation symbols of arity m.

First-order languages

Definition 2.2

The set $Expr_{\mathcal{L}}$ of expressions of $\mathcal{L}$ is the set of all finite sequences of symbols of $\mathcal{L}$.

Definition 2.3

Let $\theta = \theta_0\theta_1 \ldots \theta_{k-1}$ be an expression of $\mathcal{L}$, where $\theta_i \in Sym_{\mathcal{L}}$ for all $i = 0, \ldots, k-1$.

- *If $0 \le i \le j \le k-1$, then the expression $\theta_i \ldots \theta_j$ is called the (i, j)-subexpression of θ.*
- *We say that an expression ψ appears in θ if there exists $0 \le i \le j \le k-1$ such that ψ is the (i, j)-subexpression of θ.*
- *We denote by $Var(\theta)$ the set of variables appearing in θ.*

Definition 2.4

The terms *of $\mathcal{L}$ are the expressions defined as follows:*

(T0) *Every variable is a term.*

(T1) *Every constant symbol is a term.*

(T2) *If $m \geq 1$, $f \in \mathcal{F}_m$ and $t_1, \ldots, t_m$ are terms, then $ft_1 \ldots t_m$ is a term.*

(T3) *Only the expressions obtained by applying rules (T0), (T1), (T2) are terms.*

Notations:

- The set of terms is denoted by $Term_{\mathcal{L}}$.
- Terms are denoted by $t, s, t_1, t_2, s_1, s_2, \ldots$.
- $Var(t)$ is the set of variables that appear in the term t.

Definition 2.5

A term t is called closed *if $Var(t) = \emptyset$.*

Proposition 2.6 (Induction on terms)

Let Γ be a set of terms satisfying the following properties:

- *Γ contains the variables and the constant symbols.*
- *If $m \geq 1$, $f \in \mathcal{F}_m$ and $t_1, \ldots, t_m \in \Gamma$, then $ft_1 \ldots t_m \in \Gamma$.*

Then $\Gamma = Term_{\mathcal{L}}$.

It is used to prove that all terms have a property $\mathcal{P}$: we define Γ as the set of all terms satisfying $\mathcal{P}$ and apply induction on terms to obtain that $\Gamma = Term_{\mathcal{L}}$.

Definition 2.7

The atomic formulas *of $\mathcal{L}$ are the expressions having one of the following forms:*

- *$(s = t)$, where s, t are terms;*
- *$(Rt_1 \ldots t_m)$, where $R \in \mathcal{R}_m$ and $t_1, \ldots, t_m$ are terms.*

Definition 2.8

The formulas *of $\mathcal{L}$ are the expressions defined as follows:*

(F0) *Every atomic formula is a formula.*

(F1) *If φ is a formula, then $(\neg\varphi)$ is a formula.*

(F2) *If φ and ψ are formulas, then $(\varphi \rightarrow \psi)$ is a formula.*

(F3) *If φ is a formula, then $(\forall x \varphi)$ is a formula for every variable x.*

(F4) *Only the expressions obtained by applying rules (F0), (F1), (F2), (F3) are formulas.*

Notations

- The set of formulas is denoted by $Form_{\mathcal{L}}$.
- Formulas are denoted by $\varphi, \psi, \chi, \ldots$.
- $Var(\varphi)$ is the set of variables that appear in the formula φ.

Unique readability

If φ is a formula, then exactly one of the following hold:

- $\varphi = (s = t)$, where s, t are terms.
- $\varphi = (Rt_1 \ldots t_m)$, where $R \in \mathcal{R}_m$ and $t_1, \ldots, t_m$ are terms.
- $\varphi = (\neg\psi)$, where ψ is a formula.
- $\varphi = (\psi \rightarrow \chi)$, where ψ, χ are formulas.
- $\varphi = (\forall x \psi)$, where x is a variable and ψ is a formula.

Furthermore, φ can be written in a unique way in one of these forms.

Proposition 2.9 (Induction principle on formulas)

Let Γ be a set of formulas satisfying the following properties:

- *Γ contains all atomic formulas.*
- *Γ is closed to $\neg, \rightarrow$ and $\forall x$ (for any variable x), that is:*

$$\text{if } \varphi, \psi \in \Gamma, \text{ then } (\neg\varphi), (\varphi \rightarrow \psi), (\forall x \varphi) \in \Gamma.$$

Then $\Gamma = Form_{\mathcal{L}}$.

It is used to prove that all formulas have a property $\mathcal{P}$: we define Γ as the set of all formulas satisfying $\mathcal{P}$ and apply induction on formulas to obtain that $\Gamma = Form_{\mathcal{L}}$.

Derived connectives

Connectives $\vee$, $\wedge$, $\leftrightarrow$ and the existential quantifier $\exists$ are introduced by the following abbreviations:

$$\begin{array}{lcl} \varphi \vee \psi & := & ((\neg\varphi) \rightarrow \psi) \\ \varphi \wedge \psi & := & \neg(\varphi \rightarrow (\neg\psi))) \\ \varphi \leftrightarrow \psi & := & ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)) \\ \exists x \varphi & := & (\neg \forall x (\neg\varphi)) \end{array}$$

Usually the external parantheses are omitted, we write them only when necessary. We write $s = t$, $Rt_1 \ldots t_m$, $ft_1 \ldots t_m$, $\neg\varphi$, $\varphi \rightarrow \psi$, $\forall x \varphi$. On the other hand, we write $(\varphi \rightarrow \psi) \rightarrow \chi$.

To reduce the use of parentheses, we assume that

- $\neg$ has higher precedence than $\rightarrow, \wedge, \vee, \leftrightarrow$;
- $\wedge, \vee$ have higher precedence than $\rightarrow, \leftrightarrow$;
- quantifiers $\forall$, $\exists$ have higher precedence than the other connectives. Thus, $\forall x \varphi \rightarrow \psi$ is $(\forall x \varphi) \rightarrow \psi$ and not $\forall x (\varphi \rightarrow \psi)$.

- We write sometimes $f(t_1, \ldots, t_m)$ instead of $ft_1 \ldots t_m$ and $R(t_1, \ldots, t_m)$ instead of $Rt_1 \ldots t_m$.
- Function/relation symbols of arity 1 are called unary. Function/relation symbols of arity 2 are called binary.
- If f is a binary function symbol, we write $t_1 f t_2$ instead of $f t_1 t_2$.
- If R is a binary relation symbol, we write $t_1 R t_2$ instead of $R t_1 t_2$.

We identify often a language $\mathcal{L}$ with the set of its non-logical symbols and write $\mathcal{L} = (\mathcal{R}, \mathcal{F}, \mathcal{C})$.

First-order languages

Definition 2.10

Let $\varphi = \varphi_0\varphi_1 \dots \varphi_{n-1}$ be a formula of $\mathcal{L}$ and x be a variable.

- *We say that x occurs bound on position k in φ if $x = \varphi_k$ and there exists $0 \le i \le k \le j \le n-1$ such that the (i,j)-subexpression of φ has the form $\forall x\psi$.*
- *We say that x occurs free on position k in φ if $x = \varphi_k$, but x does not occur bound on position k in φ.*
- *x is a bound variable of φ if there exists k such that x occurs bound on position k in φ.*
- *x is a free variable of φ if there exists k such that x occurs free on position k in φ.*

Example

Let $\varphi = \forall x(x = y) \to x = z$. Free variables: x, y, z. Bound variables: x.

First-order languages

Notation: $FV(\varphi) :=$ the set of free variables of φ.

Alternative definition

The set $FV(\varphi)$ of free variables of a formula φ can be also defined by induction on formulas:

$$\begin{aligned}
FV(\varphi) &= Var(\varphi), \quad \text{if } \varphi \text{ is an atomic formula}\\
FV(\neg\varphi) &= FV(\varphi)\\
FV(\varphi \to \psi) &= FV(\varphi) \cup FV(\psi)\\
FV(\forall x\varphi) &= FV(\varphi) \setminus \{x\}.
\end{aligned}$$

$\mathcal{L}$-structures

Definition 2.11

An $\mathcal{L}$-structure is a quadruple

$$\mathcal{A} = (A, \mathcal{F}^{\mathcal{A}}, \mathcal{R}^{\mathcal{A}}, \mathcal{C}^{\mathcal{A}}),$$

where

- *A is a nonempty set.*
- *$\mathcal{F}^{\mathcal{A}} = \{f^{\mathcal{A}} \mid f \in \mathcal{F}\}$ is a set of functions on A; if f has arity m, then $f^{\mathcal{A}} : A^m \to A$.*
- *$\mathcal{R}^{\mathcal{A}} = \{R^{\mathcal{A}} \mid R \in \mathcal{R}\}$ is a set of relations on A; if R has arity m, then $R^{\mathcal{A}} \subseteq A^m$.*
- *$\mathcal{C}^{\mathcal{A}} = \{c^{\mathcal{A}} \in A \mid c \in \mathcal{C}\}$.*
- *A is called the universe of the structure $\mathcal{A}$. Notation: $A = |\mathcal{A}|$*
- *$f^{\mathcal{A}}$ ($R^{\mathcal{A}}$, $c^{\mathcal{A}}$, respectively) is called the interpretation of f (R, c, respectively) in $\mathcal{A}$.*

Examples - The language of equality $\mathcal{L}_=$

$\mathcal{L}_= = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, where

- $\mathcal{R} = \mathcal{F} = \mathcal{C} = \emptyset$;
- this language is proper for expressing the properties of equality;
- $\mathcal{L}_=$-structures are the nonempty sets.

Examples of formulas:

- equality is symmetric:

$$\forall x \forall y(x = y \to y = x)$$

- the universe has at least three elements:

$$\exists x \exists y \exists z(\neg(x = y) \land \neg(y = z) \land \neg(z = x))$$

Examples - The language of arithmetics $\mathcal{L}_{ar}$

$\mathcal{L}_{ar} = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, where

- $\mathcal{R} = \{\dot{<}\}$; $\dot{<}$ is a binary relation symbol;
- $\mathcal{F} = \{\dot{+}, \dot{\times}, \dot{S}\}$; $\dot{+}, \dot{\times}$ are binary function symbols and $\dot{S}$ is a unary function symbol;
- $\mathcal{C} = \{\dot{0}\}$.

We write $\mathcal{L}_{ar} = (\dot{<}; \dot{+}, \dot{\times}, \dot{S}; \dot{0})$ or $\mathcal{L}_{ar} = (\dot{<}, \dot{+}, \dot{\times}, \dot{S}, \dot{0})$.

The natural example of $\mathcal{L}_{ar}$-structure:

$$\mathcal{N} := (\mathbb{N}, <, +, \cdot, S, 0),$$

where $S : \mathbb{N} \to \mathbb{N}, S(m) = m + 1$ is the successor function. Thus,

$$\dot{<}^{\mathcal{N}} = <, \; \dot{+}^{\mathcal{N}} = +, \; \dot{\times}^{\mathcal{N}} = \cdot, \; \dot{S}^{\mathcal{N}} = S, \; \dot{0}^{\mathcal{N}} = 0.$$

- Another example of $\mathcal{L}_{ar}$-structure: $\mathcal{A} = (\{0, 1\}, <, \vee, \wedge, \neg, 1)$.

Examples - The language with a binary relation symbol

$\mathcal{L}_R = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, where

- $\mathcal{R} = \{R\}$; R is a binary relation symbol;
- $\mathcal{F} = \mathcal{C} = \emptyset$;
- $\mathcal{L}$-structures are nonempty sets together with a binary relation.

- If we are interested in partially ordered sets $(A, \leq)$, we use the symbol $\dot{\leq}$ instead of R and we denote the language by $\mathcal{L}_{\leq}$.
- If we are interested in strictly ordered sets $(A, <)$, we use the symbol $\dot{<}$ instead of R and we denote the language by $\mathcal{L}_{<}$.
- If we are interested in graphs $G = (V, E)$, we use the symbol $\dot{E}$ instead of R and we denote the language by $\mathcal{L}_{Graf}$.
- If we are interested in structures $(A, \in)$, we use the symbol $\dot{\in}$ instead of R and we denote the language by $\mathcal{L}_{\in}$.

Semantics

Let $\mathcal{L}$ be a first-order language and $\mathcal{A}$ be an $\mathcal{L}$-structure.

Definition 2.12

An $\mathcal{A}$-assignment or $\mathcal{A}$-evaluation is a function $e : V \to A$.

When the $\mathcal{L}$-structure $\mathcal{A}$ is clear from the context, we also write simply e is an assignment.

In the following, $e : V \to A$ is an $\mathcal{A}$-assignment.

Definition 2.13 (Interpretation of terms)

The interpretation $t^{\mathcal{A}}(e) \in A$ of a term t under the $\mathcal{A}$-assignment *e is defined by induction on terms :*

- *if $t = x \in V$, then $t^{\mathcal{A}}(e) := e(x)$;*
- *if $t = c \in \mathcal{C}$, then $t^{\mathcal{A}}(e) := c^{\mathcal{A}}$;*
- *if $t = ft_1 \dots t_m$, then $t^{\mathcal{A}}(e) := f^{\mathcal{A}}(t_1^{\mathcal{A}}(e), \dots, t_m^{\mathcal{A}}(e))$.*

Semantics

The interpretation

$$\varphi^{\mathcal{A}}(e) \in \{0, 1\}$$

of a *formula φ under the $\mathcal{A}$-assignment e* is defined by induction on formulas.

$$(s = t)^{\mathcal{A}}(e) = \begin{cases} 1 & \text{if } s^{\mathcal{A}}(e) = t^{\mathcal{A}}(e) \\ 0 & \text{otherwise.} \end{cases}$$

$$(Rt_1 \dots t_m)^{\mathcal{A}}(e) = \begin{cases} 1 & \text{if } R^{\mathcal{A}}(t_1^{\mathcal{A}}(e), \dots, t_m^{\mathcal{A}}(e)) \\ 0 & \text{otherwise.} \end{cases}$$

Negation and implication

- $(\neg\varphi)^{\mathcal{A}}(e) = 1 - \varphi^{\mathcal{A}}(e)$;
- $(\varphi \to \psi)^{\mathcal{A}}(e) = \varphi^{\mathcal{A}}(e) \to \psi^{\mathcal{A}}(e)$, where,

$\to: \{0,1\} \times \{0,1\} \to \{0,1\}$,

p	q	$p \to q$
0	0	1
0	1	1
1	0	0
1	1	1

Hence,

- $(\neg\varphi)^{\mathcal{A}}(e) = 1$ iff $\varphi^{\mathcal{A}}(e) = 0$.
- $(\varphi \to \psi)^{\mathcal{A}}(e) = 1$ iff $\big(\ \varphi^{\mathcal{A}}(e) = 0$ or $\psi^{\mathcal{A}}(e) = 1\big)$.

Notation

For any variable $x \in V$ and any $a \in A$, we define a new $\mathcal{A}$-assignment $e_{x\leftarrow a} : V \to A$ by

$$e_{x\leftarrow a}(v) = \begin{cases} e(v) & \text{if } v \neq x \\ a & \text{if } v = x. \end{cases}$$

Universal quantifier

$$(\forall x\varphi)^{\mathcal{A}}(e) \quad = \quad \begin{cases} 1 & \text{if } \varphi^{\mathcal{A}}(e_{x\leftarrow a}) = 1 \text{ for all } a \in A \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{A}$ be an $\mathcal{L}$-structure and $e : V \to A$ be an $\mathcal{A}$-assignment.

Definition 2.14

Let φ be a formula. We say that:

- *e satisfies φ in $\mathcal{A}$ if $\varphi^{\mathcal{A}}(e) = 1$. Notation: $\mathcal{A} \models \varphi[e]$.*
- *e does not satisfy φ in $\mathcal{A}$ if $\varphi^{\mathcal{A}}(e) = 0$. Notation: $\mathcal{A} \not\models \varphi[e]$.*

Proposition 2.15

For all formulas φ, ψ and any variable x,

(i) $\mathcal{A} \models \neg\varphi[e]$ iff $\mathcal{A} \not\models \varphi[e]$.

(ii) $\mathcal{A} \models (\varphi \to \psi)[e]$ iff ($\mathcal{A} \models \varphi[e]$ implies $\mathcal{A} \models \psi[e]$)
iff ($\mathcal{A} \not\models \varphi[e]$ or $\mathcal{A} \models \psi[e]$).

(iii) $\mathcal{A} \models (\forall x\varphi)[e]$ iff for all $a \in A$, $\mathcal{A} \models \varphi[e_{x\leftarrow a}]$.

Proposition 2.16

For all formulas φ, ψ and any variable x,

(i) $\mathcal{A} \models (\varphi \wedge \psi)[e]$ iff ($\mathcal{A} \models \varphi[e]$ and $\mathcal{A} \models \psi[e]$).

(ii) $\mathcal{A} \models (\varphi \vee \psi)[e]$ iff ($\mathcal{A} \models \varphi[e]$ or $\mathcal{A} \models \psi[e]$).

(iii) $\mathcal{A} \models (\varphi \leftrightarrow \psi)[e]$ iff ($\mathcal{A} \models \varphi[e]$ iff $\mathcal{A} \models \psi[e]$).

(iv) $\mathcal{A} \models (\exists x\varphi)[e]$ iff there exists $a \in A$ s.t. $\mathcal{A} \models \varphi[e_{x\leftarrow a}]$.

Semantics

Let φ be a formula of $\mathcal{L}$.

Definition 2.17

φ is satisfiable if there exists an $\mathcal{L}$-structure $\mathcal{A}$ and an $\mathcal{A}$-assignment e such that $\mathcal{A} \vDash \varphi[e]$.
We also say that $(\mathcal{A}, e)$ is a model of φ.

Definition 2.18

φ is true in an $\mathcal{L}$-structure $\mathcal{A}$ if $\mathcal{A} \vDash \varphi[e]$ for all $\mathcal{A}$-assignments e.
We also say that $\mathcal{A}$ satisfies φ or that $\mathcal{A}$ is a model of φ.
Notation: $\mathcal{A} \vDash \varphi$

Definition 2.19

φ is universally true (or logically valid or, simply, valid) if $\mathcal{A} \vDash \varphi$ for all $\mathcal{L}$-structures $\mathcal{A}$.
Notation: $\vDash \varphi$

Semantics

Let φ, ψ be formulas of $\mathcal{L}$.

Definition 2.20

ψ is a logical consequence of φ if for all $\mathcal{L}$-structures $\mathcal{A}$ and all $\mathcal{A}$-assignments e,

$$\mathcal{A} \vDash \varphi[e] \quad \textit{implies} \quad \mathcal{A} \vDash \psi[e].$$

Notation: $\varphi \vDash \psi$

Definition 2.21

φ and ψ are logically equivalent or, simply, equivalent if for all $\mathcal{L}$-structures $\mathcal{A}$ and all $\mathcal{A}$-assignments e,

$$\mathcal{A} \vDash \varphi[e] \textit{ iff } \mathcal{A} \vDash \psi[e].$$

Notation: $\varphi \;\text{⫤⊨}\; \psi$

Remark

- $\varphi \vDash \psi$ iff $\vDash \varphi \to \psi$.
- $\varphi \;\text{⫤⊨}\; \psi$ iff ($\psi \vDash \varphi$ and $\varphi \vDash \psi$) iff $\vDash \psi \leftrightarrow \varphi$.

Semantics

For all formulas φ, ψ and all variables x, y,

$$
\begin{array}{rcll}
\neg\exists x\varphi & \text{⫤⊨} & \forall x\neg\varphi & (1)\\
\neg\forall x\varphi & \text{⫤⊨} & \exists x\neg\varphi & (2)\\
\forall x(\varphi\wedge\psi) & \text{⫤⊨} & \forall x\varphi\wedge\forall x\psi & (3)\\
\forall x\varphi\vee\forall x\psi & \vDash & \forall x(\varphi\vee\psi) & (4)\\
\exists x(\varphi\wedge\psi) & \vDash & \exists x\varphi\wedge\exists x\psi & (5)\\
\exists x(\varphi\vee\psi) & \text{⫤⊨} & \exists x\varphi\vee\exists x\psi & (6)\\
\forall x(\varphi\to\psi) & \vDash & \forall x\varphi\to\forall x\psi & (7)\\
\forall x(\varphi\to\psi) & \vDash & \exists x\varphi\to\exists x\psi & (8)\\
\forall x\varphi & \vDash & \exists x\varphi & (9)
\end{array}
$$

Semantics

$$
\begin{array}{rcll}
\varphi & \vDash & \exists x\varphi & (10)\\
\forall x\varphi & \vDash & \varphi & (11)\\
\forall x\forall y\varphi & \text{⫤⊨} & \forall y\forall x\varphi & (12)\\
\exists x\exists y\varphi & \text{⫤⊨} & \exists y\exists x\varphi & (13)\\
\exists y\forall x\varphi & \vDash & \forall x\exists y\varphi. & (14)
\end{array}
$$

Proposition 2.22

For all terms s, t, u,

(i) $\models t = t$;

(ii) $\models s = t \to t = s$;

(iii) $\models s = t \wedge t = u \to s = u$.

Proposition 2.23

For all $m \geq 1$, $f \in \mathcal{F}_m, R \in \mathcal{R}_m$ and all terms $t_i, u_i, i = 1, \ldots, m$,

$$\models (t_1 = u_1) \wedge \ldots \wedge (t_m = u_m) \to ft_1 \ldots t_m = fu_1 \ldots u_m$$
$$\models (t_1 = u_1) \wedge \ldots \wedge (t_m = u_m) \to (Rt_1 \ldots t_m \leftrightarrow Ru_1 \ldots u_m)$$

Proposition 2.24

For any $\mathcal{L}$-structure $\mathcal{A}$ and any $\mathcal{A}$-assignments e_1, e_2,

(i) for any term t,

if $e_1(v) = e_2(v)$ for all variables $v \in Var(t)$, then $t^{\mathcal{A}}(e_1) = t^{\mathcal{A}}(e_2)$.

(ii) for any formula φ,

if $e_1(v) = e_2(v)$ for all variables $v \in FV(\varphi)$, then $\mathcal{A} \models \varphi[e_1]$ iff $\mathcal{A} \models \varphi[e_2]$.

Proposition 2.25

For all formulas φ, ψ and any variable $x \notin FV(\varphi)$,

$$\varphi \mathrel{\models\!\!\!\dashv} \exists x\varphi \tag{15}$$
$$\varphi \mathrel{\models\!\!\!\dashv} \forall x\varphi \tag{16}$$
$$\forall x(\varphi \wedge \psi) \mathrel{\models\!\!\!\dashv} \varphi \wedge \forall x\psi \tag{17}$$
$$\forall x(\varphi \vee \psi) \mathrel{\models\!\!\!\dashv} \varphi \vee \forall x\psi \tag{18}$$
$$\exists x(\varphi \wedge \psi) \mathrel{\models\!\!\!\dashv} \varphi \wedge \exists x\psi \tag{19}$$
$$\exists x(\varphi \vee \psi) \mathrel{\models\!\!\!\dashv} \varphi \vee \exists x\psi \tag{20}$$
$$\forall x(\varphi \to \psi) \mathrel{\models\!\!\!\dashv} \varphi \to \forall x\psi \tag{21}$$
$$\exists x(\varphi \to \psi) \mathrel{\models\!\!\!\dashv} \varphi \to \exists x\psi \tag{22}$$
$$\forall x(\psi \to \varphi) \mathrel{\models\!\!\!\dashv} \exists x\psi \to \varphi \tag{23}$$
$$\exists x(\psi \to \varphi) \mathrel{\models\!\!\!\dashv} \forall x\psi \to \varphi \tag{24}$$

Definition 2.26

A formula φ is called a sentence if $FV(\varphi) = \emptyset$, that is φ does not have free variables.
Notation: $Sent_{\mathcal{L}}$:= the set of sentences of $\mathcal{L}$.

Proposition 2.27

Let φ be a sentence. For all $\mathcal{A}$-assignments e_1, e_2,

$$\mathcal{A} \models \varphi[e_1] \iff \mathcal{A} \models \varphi[e_2]$$

Definition 2.28

Let φ be a sentence. An $\mathcal{L}$-structure $\mathcal{A}$ is a model of φ if $\mathcal{A} \models \varphi[e]$ for an (any) $\mathcal{A}$-assignment e. Notation: $\mathcal{A} \models \varphi$

Semantics

Let φ be a formula and Γ be a set of formulas of $\mathcal{L}$.

Definition 2.29

We say that Γ is satisfiable if there exists an $\mathcal{L}$-structure $\mathcal{A}$ and an $\mathcal{A}$-assignment e such that

$$\mathcal{A} \models \gamma[e] \text{ for all } \gamma \in \Gamma.$$

$(\mathcal{A}, e)$ is called a model of Γ.

Definition 2.30

We say that φ is a logical consequence of Γ if for all $\mathcal{L}$-structures $\mathcal{A}$ and all $\mathcal{A}$-assignments $e : V \to A$,

$$(\mathcal{A}, e) \text{ model of } \Gamma \implies (\mathcal{A}, e) \text{ model of } \varphi.$$

Notation: $\Gamma \models \varphi$

Semantics

Let φ be a sentence and Γ be a set of sentences of $\mathcal{L}$.

Definition 2.31

We say that Γ is satisfiable if there exists an $\mathcal{L}$-structure $\mathcal{A}$ such that

$$\mathcal{A} \models \gamma \text{ for all } \gamma \in \Gamma.$$

$\mathcal{A}$ is called a model of Γ. Notation: $\mathcal{A} \models \Gamma$

Definition 2.32

We say that φ is a logical consequence of Γ if for all $\mathcal{L}$-structures $\mathcal{A}$,

$$\mathcal{A} \models \Gamma \implies \mathcal{A} \models \varphi.$$

Notation: $\Gamma \models \varphi$

Tautologies

The notions of tautology and tautological consequence from propositional logic can also be applied to a first-order language $\mathcal{L}$. Intuitively, a tautology is a formula which is "true" based only on the interpretations of the connectives $\neg, \to$.

Definition 2.33

An $\mathcal{L}$-truth assignment is a function $F : Form_{\mathcal{L}} \to \{0, 1\}$ satisfying, for all formulas φ, ψ,

- $F(\neg\varphi) = 1 - F(\varphi)$;
- $F(\varphi \to \psi) = F(\varphi) \to F(\psi)$.

Definition 2.34

φ is a tautology if $F(\varphi) = 1$ for any $\mathcal{L}$-truth assignment F.

Examples of tautologies: $\varphi \to (\psi \to \varphi)$, $(\varphi \to \psi) \leftrightarrow (\neg\psi \to \neg\varphi)$

Tautologies

Proposition 2.35

If φ is a tautology, then φ is valid.

Example

$x = x$ is valid, but $x = x$ is not a tautology.

Definition 2.36

We say that the formulas φ and ψ are tautologically equivalent if $F(\varphi) = F(\psi)$ for any $\mathcal{L}$-truth assignment F.

Example 2.37

$\varphi_1 \to (\varphi_2 \to \ldots \to (\varphi_n \to \psi) \ldots)$ and $(\varphi_1 \land \ldots \land \varphi_n) \to \psi$ are tautologically equivalent.

Tautologies

Definition 2.38

Let φ be a formula and Γ be a set of formulas. We say that φ is a tautological consequence *of Γ if for any $\mathcal{L}$-truth assignment F,*

$$F(\gamma) = 1 \text{ for all } \gamma \in \Gamma \quad \Rightarrow \quad F(\varphi) = 1.$$

Proposition 2.39

If φ is a tautological consequence of Γ, then $\Gamma \vDash \varphi$.

Substitution

Let x be a variable of $\mathcal{L}$ and u be a term of $\mathcal{L}$.

Definition 2.40

For any term t of $\mathcal{L}$, we define

$t_x(u)$:= *the expression obtained from t by replacing all occurences of x with u.*

Proposition 2.41

For any term t of $\mathcal{L}$, $t_x(u)$ is a term of $\mathcal{L}$.

Substitution

- We would like to define, similarly, $\varphi_x(u)$ as the expression obtained from φ by replacing all free occurences of x in φ with u.
- We expect that the following natural properties of substitution are true:

$$\vDash \forall x\varphi \rightarrow \varphi_x(u) \quad \text{and} \quad \vDash \varphi_x(u) \rightarrow \exists x\varphi.$$

As the following example shows, there are problems with this definition.

Let $\varphi := \exists y \neg(x = y)$ and $u := y$. Then $\varphi_x(u) = \exists y \neg(y = y)$.
Avem

- For any $\mathcal{L}$-structure $\mathcal{A}$ with $|A| \geq 2$, $\mathcal{A} \vDash \forall x\varphi$.
- $\varphi_x(u)$ is not satisfiable.

Substitution

Let x be a variable, u a term and φ a formula.

Definition 2.42

We say that x is free for u *in φ or that u is* substitutable for x *in φ if for any variable y that occurs in u, no subformula of φ of the form $\forall y\psi$ contains free occurences of x.*

Remark

x is free for u in φ in any of the following cases:

- u does not contain variables;
- φ does not contain variables that occur in u;
- no variable from u occurs bound in φ;
- x does not occur in φ;
- φ does not contain free occurences of x.

Let x be a variable, u a term and φ be a formula such that x is free for u in φ.

Definition 2.43

$\varphi_x(u) \quad :=$ *the expression obtained from φ by replacing all free occurences of x in φ with u.*

We say that $\varphi_x(u)$ is a free substitution.

Proposition 2.44

$\varphi_x(u)$ is a formula of $\mathcal{L}$.

Free substitution rules out the problems mentioned above, it behaves as expected.

Proposition 2.45

Let φ be a formula and x be a variable.

(i) *For any term u substitutable for x in φ,*
$$\vDash \forall x\varphi \to \varphi_x(u) \quad \text{and} \quad \vDash \varphi_x(u) \to \exists x\varphi.$$

(ii) $\vDash \forall x\varphi \to \varphi$ *and* $\vDash \varphi \to \exists x\varphi$.

(iii) *For any constant symbol c,*
$$\vDash \forall x\varphi \to \varphi_x(c) \text{ and } \vDash \varphi_x(c) \to \exists x\varphi.$$

Proposition 2.46

For any formula φ, distinct variables x and y such that $y \notin FV(\varphi)$ and y is substitutable for x in φ,
$$\exists x\varphi \mathrel{\text{⫤}\!\vDash} \exists y\varphi_x(y) \quad \text{and} \quad \forall x\varphi \mathrel{\text{⫤}\!\vDash} \forall y\varphi_x(y).$$
In particular, this holds if y is a new variable, that does not occur in φ.

We use Proposition 2.46 as follows: if $\varphi_x(u)$ is not a free substitution (that is x is not free for u in φ), then we replace φ with a logically equivalent formula φ' such that $\varphi'_x(u)$ is a free substitution .

Definition 2.47

For any formula φ and any variables $y_1, \dots, y_k$, the $y_1, \dots, y_k$-free variant φ' of φ is inductively defined as follows:

- *if φ is an atomic formula, then φ' is φ;*
- *if $\varphi = \neg\psi$, then φ' is $\neg\psi'$;*
- *if $\varphi = \psi \to \chi$, then φ' is $\psi' \to \chi'$;*
- *if $\varphi = \forall z\psi$, then*
$$\varphi' = \begin{cases} \forall w\psi'_z(w) & \text{if } z \in \{y_1, \dots, y_k\} \\ \forall z\psi' & \text{altfel;} \end{cases}$$
where w is the first variable in the sequence $v_0, v_1, \dots,$ which does not occur in ψ' and is not among $y_1, \dots, y_k$.

Substitution

Definition 2.48

φ' is a variant of φ if it is the $y_1, \ldots, y_k$-free variant of φ for some variables $y_1, \ldots, y_k$.

Proposition 2.49

(i) For any formulas φ and φ', if φ' is a variant of φ, then $\varphi \vDash\!\dashv \varphi'$;

(ii) For any formula φ and any term u, if the variables of u are among $y_1, \ldots, y_k$ and φ' is the $y_1, \ldots, y_k$-free variant of φ, then $\varphi'_x(u)$ is a free substitution.

Syntax

Definition 2.50

The set $LogAx_{\mathcal{L}} \subseteq Form_{\mathcal{L}}$ of logical axioms of $\mathcal{L}$ consists of:

(i) all tautologies.

(ii) formulas of the form

$$t = t, \quad s = t \to t = s, \quad s = t \wedge t = u \to s = u,$$

for any terms s, t, u.

(iii) formulas of the form

$$t_1 = u_1 \wedge \ldots \wedge t_m = u_m \to ft_1 \ldots t_m = fu_1 \ldots u_m,$$

$$t_1 = u_1 \wedge \ldots \wedge t_m = u_m \to (Rt_1 \ldots t_m \leftrightarrow Ru_1 \ldots u_m),$$

for any $m \geq 1$, $f \in \mathcal{F}_m$, $R \in \mathcal{R}_m$ and any terms s_i, t_i $(i = 1, \ldots, m)$.

(iv) formulas of the form

$$\varphi_x(t) \to \exists x \varphi,$$

where $\varphi_x(t)$ is a free substitution ($\exists$-axioms).

Syntax

Definition 2.51

The deduction rules (or inference rules) are the following: for any formulas φ, ψ,

(i) from φ and $\varphi \to \psi$ infer ψ (modus ponens or (MP)):

$$\frac{\varphi, \ \varphi \to \psi}{\psi}$$

(ii) if $x \notin FV(\psi)$, then from $\varphi \to \psi$ infer $\exists x \varphi \to \psi$ ($\exists$-introduction):

$$\frac{\varphi \to \psi}{\exists x \varphi \to \psi} \quad \text{if } x \notin FV(\psi).$$

Syntax

Let Γ be a set of formulas of $\mathcal{L}$.

Definition 2.52

The Γ-theorems of $\mathcal{L}$ are the formulas defined as follows:

(Γ0) Every logical axiom is a Γ-theorem.

(Γ1) Every formula of Γ is a Γ-theorem.

(Γ2) If φ and $\varphi \to \psi$ are Γ-theorems, then ψ is a Γ-theorem.

(Γ3) If $\varphi \to \psi$ is a Γ-theorem and $x \notin FV(\psi)$, then $\exists x \varphi \to \psi$ is a Γ-theorem.

(Γ4) Only the formulas obtained by applying rules (Γ0), (Γ1), (Γ2) and (Γ3) are Γ-theorems.

If φ is a Γ-theorem, then we also say that φ is deduced from the hypotheses Γ.

Notations

$\Gamma \vdash_{\mathcal{L}} \varphi \quad := \quad \varphi$ is a Γ-theorem

$\vdash_{\mathcal{L}} \varphi \quad := \quad \emptyset \vdash_{\mathcal{L}} \varphi$

Definition 2.53

A formula φ is called a (logical) theorem of $\mathcal{L}$ if $\vdash_{\mathcal{L}} \varphi$.

Convention

When $\mathcal{L}$ is clear from the context, we write $\Gamma \vdash \varphi$, $\vdash \varphi$, etc..

Definition 2.54

A Γ-proof (or proof from the hypotheses Γ) of $\mathcal{L}$ is a sequence of formulas $\theta_1, \ldots, \theta_n$ such that for all $i \in \{1, \ldots, n\}$, one of the following holds:

(i) θ_i *is an axiom;*

(ii) $\theta_i \in \Gamma$;

(iii) *there exist $k, j < i$ such that $\theta_k = \theta_j \to \theta_i$;*

(iv) *there exists $j < i$ such that*

$$\theta_j = \varphi \to \psi \text{ and } \theta_i = \exists x \varphi \to \psi,$$

where φ, ψ are formulas and $x \notin FV(\psi)$.

A $\emptyset$-proof is called simply a proof.

Definition 2.55

Let φ be a formula. A Γ-proof of φ or a proof of φ from the hypotheses Γ is a Γ-proof $\theta_1, \ldots, \theta_n$ such that $\theta_n = \varphi$.

Proposition 2.56

Let Γ be a set of formulas. For any formula φ,

$$\Gamma \vdash \varphi \quad \text{iff} \quad \text{there exists a } \Gamma\text{-proof of } \varphi.$$

Let Γ be a set of formulas.

Theorem 2.57 (Tautology Theorem (Post))

If ψ is a tautological consequence of $\{\varphi_1, \ldots, \varphi_n\}$ and $\Gamma \vdash \varphi_1, \ldots, \Gamma \vdash \varphi_n$, then $\Gamma \vdash \psi$.

Theorem 2.58 (Deduction Theorem)

Let $\Gamma \cup \{\psi\}$ be a set of formulas and φ be a sentence. Then

$$\Gamma \cup \{\varphi\} \vdash \psi \quad \text{iff} \quad \Gamma \vdash \varphi \to \psi.$$

Definition 2.59

Γ is called consistent if there exists a formula φ such that $\Gamma \not\vdash \varphi$. Γ is said to be inconsistent if it is not consistent, that is $\Gamma \vdash \varphi$ for any formula φ.

Syntax

Proposition 2.60

For any formula φ and variable x,

$$\Gamma \vdash \varphi \quad \Longleftrightarrow \quad \Gamma \vdash \forall x \varphi.$$

Definition 2.61

Let φ be a formula with $FV(\varphi) = \{x_1, \ldots, x_n\}$. The universal closure of φ is the sentence

$$\overline{\forall \varphi} := \forall x_1 \ldots \forall x_n \varphi.$$

Notation 2.62

$\overline{\forall \Gamma} := \{\overline{\forall \psi} \mid \psi \in \Gamma\}$.

Proposition 2.63

For any formula φ,

$$\Gamma \vdash \varphi \quad \Longleftrightarrow \quad \Gamma \vdash \overline{\forall \varphi} \quad \Longleftrightarrow \quad \overline{\forall \Gamma} \vdash \varphi \quad \Longleftrightarrow \quad \overline{\forall \Gamma} \vdash \overline{\forall \varphi}.$$

Completeness Theorem

Theorem 2.64 (Completeness Theorem (version 1))

Let Γ be a set of sentences. Then

$$\Gamma \text{ is consistent} \quad \Longleftrightarrow \quad \Gamma \text{ is satisfiable.}$$

Theorem 2.65 (Completeness Theorem (version 2))

For any set of sentences Γ and any sentence φ,

$$\Gamma \vdash \varphi \quad \Longleftrightarrow \quad \Gamma \vDash \varphi.$$

- The Completeness Theorem was proved by Gödel in 1929 in his PhD thesis.
- Henkin gave in 1949 a simplified proof.

Prenex normal form

Definition 2.66

A formula that does not contain quantifiers is called quantifier-free.

Definition 2.67

A formula φ is in prenex normal form if

$$\varphi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \, \psi,$$

where $n \in \mathbb{N}$, $Q_1, \ldots, Q_n \in \{\forall, \exists\}$, $x_1, \ldots, x_n$ are variables and ψ is a quantifier-free formula. $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n$ is the prefix of φ and ψ is called the matrix of φ.

Any quantifier-free formula is in prenex normal form, as one can take $n = 0$ in the above definition.

Prenex normal form

Examples of formulas in prenex normal form:

- universal formulas: $\varphi = \forall x_1 \forall x_2 \ldots \forall x_n \, \psi$, where ψ is quantifier-free
- existential formulas: $\varphi = \exists x_1 \exists x_2 \ldots \exists x_n \, \psi$, where ψ is quantifier-free
- $\forall\exists$-formulas: $\varphi = \forall x_1 \forall x_2 \ldots \forall x_n \exists y_1 \exists y_2 \ldots \exists y_k \psi$, where ψ is quantifier-free
- $\forall\exists\forall$-formulas: $\varphi = \forall x_1 \ldots \forall x_n \exists y_1 \ldots \exists y_k \forall z_1 \ldots \forall z_p \psi$, where ψ is quantifier-free

Theorem 2.68 (Prenex normal form theorem)

For any formula φ there exists a formula φ^* in prenex normal form such that $\varphi \mathrel{\text{⫤⊨}} \varphi^*$ and $FV(\varphi) = FV(\varphi^*)$. φ^* is called a prenex normal form of φ.

Prenex normal form

Let $\mathcal{L}$ be a first-order language containing

- two unary relation symbols R, S and two binary relation symbols P, Q;
- a unary function symbol f and a binary function symbol g;
- two constant symbols c, d.

Example

Find a prenex normal form of the formula

$$\varphi := \exists y(g(y,z) = c) \wedge \neg\exists x(f(x) = d)$$

We have that

$$\begin{aligned} \varphi &\models\!\!\mid \exists y\big(g(y,z) = c \wedge \neg\exists x(f(x) = d)\big) \\ &\models\!\!\mid \exists y\big(g(y,z) = c \wedge \forall x \neg(f(x) = d)\big) \\ &\models\!\!\mid \exists y \forall x\big(g(y,z) = c \wedge \neg(f(x) = d)\big) \end{aligned}$$

Thus, $\varphi^* = \exists y \forall x\big(g(y,z) = c \wedge \neg(f(x) = d)\big)$ is a prenex normal form of φ.

Prenex normal form

Example

Find a prenex normal form of the formula

$$\varphi := \neg\forall y(S(y) \to \exists z R(z)) \wedge \forall x(\forall y P(x,y) \to f(x) = d).$$

$$\begin{aligned} \varphi &\models\!\!\mid \exists y \neg(S(y) \to \exists z R(z)) \wedge \forall x(\forall y P(x,y) \to f(x) = d) \\ &\models\!\!\mid \exists y \neg\exists z(S(y) \to R(z)) \wedge \forall x(\forall y P(x,y) \to f(x) = d) \\ &\models\!\!\mid \exists y \neg\exists z(S(y) \to R(z)) \wedge \forall x \exists y(P(x,y) \to f(x) = d) \\ &\models\!\!\mid \exists y \forall z \neg(S(y) \to R(z)) \wedge \forall x \exists y(P(x,y) \to f(x) = d) \\ &\models\!\!\mid \exists y \forall z\big(\neg(S(y) \to R(z)) \wedge \forall x \exists y(P(x,y) \to f(x) = d)\big) \\ &\models\!\!\mid \exists y \forall z \forall x\big(\neg(S(y) \to R(z)) \wedge \exists y(P(x,y) \to f(x) = d)\big) \\ &\models\!\!\mid \exists y \forall z \forall x\big(\neg(S(y) \to R(z)) \wedge \exists v(P(x,v) \to f(x) = d)\big) \\ &\models\!\!\mid \exists y \forall z \forall x \exists v\big(\neg(S(y) \to R(z)) \wedge (P(x,v) \to f(x) = d)\big) \end{aligned}$$

$\varphi^* = \exists y \forall z \forall x \exists v\big(\neg(S(y) \to R(z)) \wedge (P(x,v) \to f(x) = d)\big)$ is a prenex normal form of φ.