1 Motivation

Motivation

Syntax

Semantics

Normal forms

Herbrand interpretations

Further Theorems

Literature
In propositional logic, the only building blocks are atomic propositions.

We cannot talk about the internal structures of these propositions.

Example:

- All CS students know formal logic
- Peter is a CS student
- Therefore, Peter knows formal logic

...not possible in propositional logic

Idea: We introduce predicates, functions, object variables and quantifiers.
2 Syntax
Motivation
Syntax
Semantics
Normal forms
Herbrand interpretations
Further Theorems
Literature

Syntax

- **variable** symbols: \(x, y, z, \ldots\)
- **\(n\)-ary function** symbols: \(f, g, \ldots\)
- **constant** symbols: \(a, b, c, \ldots\)
- **\(n\)-ary predicate** symbols: \(P, Q, \ldots\)
- **logical** symbols: \(\forall, \exists, =, \neg, \wedge, \ldots\)

Terms

\[ t ::= x \quad \text{variable} \]
\[ | \quad f(t_1, \ldots, t_n) \quad \text{function application} \]
\[ | \quad a \quad \text{constant} \]

Formulae

\[ \varphi ::= P(t_1, \ldots, t_n) \quad \text{atomic formulae} \]
\[ | \quad t = t' \quad \text{identity formulae} \]
\[ | \quad \ldots \quad \text{propositional connectives} \]
\[ | \quad \forall x \varphi' \quad \text{universal quantification} \]
\[ | \quad \exists x \varphi' \quad \text{existential quantification} \]

**Ground term**, etc.: term, etc. without variable occurrences
3 Semantics

- Interpretations
- Variable Assignments
- Definition of Truth
- Terminology
- Free and Bound Variables
- Open and Closed Formulae
Semantics: idea

- In FOL, the universe of discourse consists of objects: we consider functions and relations over these objects.
- Function symbols are mapped to functions, predicate symbols are mapped to relations, and terms to objects.
- **Notation:** Instead of $I(x)$ we write $x^I$.
- **Note:** Usually one considers all possible non-empty universes. (However, sometimes the interpretations are restricted to particular domains, e.g. integers or real numbers.)
- Satisfiability and validity is then considered wrt. all these universes.
Formal semantics: interpretations

**Interpretations:** \( \mathcal{I} = \langle \mathcal{D}, \cdot^\mathcal{I} \rangle \) with \( \mathcal{D} \) being an arbitrary non-empty set and \( \cdot^\mathcal{I} \) being a function which maps

- \( n \)-ary function symbols \( f \) to \( n \)-ary functions \( f^\mathcal{I} \in [\mathcal{D}^n \rightarrow \mathcal{D}] \),
- constant symbols \( a \) to objects \( a^\mathcal{I} \in \mathcal{D} \), and
- \( n \)-ary predicates \( P \) to \( n \)-ary relations \( P^\mathcal{I} \subseteq \mathcal{D}^n \).

**Interpretation** of ground terms:

\[
(f(t_1, \ldots, t_n))^\mathcal{I} = f^\mathcal{I}(t_1^\mathcal{I}, \ldots, t_n^\mathcal{I}) \ (\in \mathcal{D})
\]

**Truth** of ground atoms:

\[
\mathcal{I} \models P(t_1, \ldots, t_n) \iff \langle t_1^\mathcal{I}, \ldots, t_n^\mathcal{I} \rangle \in P^\mathcal{I}
\]
Examples

\[ \mathcal{D} = \{d_1, \ldots, d_n\}, \ n \geq 2 \]
\[ \mathcal{D} = \{1, 2, 3, \ldots\} \]
\[ a^\mathcal{I} = d_1 \]
\[ 1^\mathcal{I} = 1 \]
\[ b^\mathcal{I} = d_2 \]
\[ 2^\mathcal{I} = 2 \]
\[ \text{Cat}^\mathcal{I} = \{d_1\} \]
\[ \text{Red}^\mathcal{I} = \mathcal{D} \]
\[ \text{even}^\mathcal{I} = \{2, 4, 6, \ldots\} \]
\[ \text{succ}^\mathcal{I} = \{(1 \mapsto 2), (2 \mapsto 3), \ldots\} \]
\[ \mathcal{I} \models \text{Red}(b) \]
\[ \mathcal{I} \not\models \text{Cat}(b) \]
\[ \mathcal{I} \not\models \text{even}(3) \]
\[ \mathcal{I} \models \text{even}(\text{succ}(3)) \]
Formal semantics: variable assignments

$\mathcal{V}$ is the set of variables. Functions $\alpha : \mathcal{V} \rightarrow \mathcal{D}$ are called variable assignments.

Notation: $\alpha[x/d]$ is identical to $\alpha$ except for $x$ where $\alpha[x/d](x) = d$.

Interpretation of terms under $\mathcal{I}, \alpha$:

\[
\begin{align*}
\text{x}^{\mathcal{I}, \alpha} & = \alpha(x) \\
\text{a}^{\mathcal{I}, \alpha} & = a^\mathcal{I} \\
(f(t_1, \ldots, t_n))^{\mathcal{I}, \alpha} & = f^\mathcal{I}(t_1^{\mathcal{I}, \alpha}, \ldots, t_n^{\mathcal{I}, \alpha})
\end{align*}
\]

Truth of atomic formulae:

$\mathcal{I}, \alpha \models P(t_1, \ldots, t_n)$ iff $\langle t_1^{\mathcal{I}, \alpha}, \ldots, t_n^{\mathcal{I}, \alpha} \rangle \in P^\mathcal{I}$

Example (cont’d):

$\alpha = \{ x \mapsto d_1, y \mapsto d_2 \}$  \hspace{1cm} $\mathcal{I}, \alpha \models \text{Red}(x)$  \hspace{1cm} $\mathcal{I}, \alpha[y/d_1] \models \text{Cat}(y)$
Formal semantics: truth

Truth of $\varphi$ under $\mathcal{I}$ and $\alpha$ ($\mathcal{I}, \alpha \models \varphi$) is defined as follows.

- $\mathcal{I}, \alpha \models P(t_1, \ldots, t_n)$ iff $\langle t_1^{\mathcal{I}, \alpha}, \ldots, t_n^{\mathcal{I}, \alpha} \rangle \in P^\mathcal{I}$
- $\mathcal{I}, \alpha \models t_1 = t_2$ iff $t_1^{\mathcal{I}, \alpha} = t_2^{\mathcal{I}, \alpha}$
- $\mathcal{I}, \alpha \models \neg \varphi$ iff $\mathcal{I}, \alpha \not\models \varphi$
- $\mathcal{I}, \alpha \models \varphi \land \psi$ iff $\mathcal{I}, \alpha \models \varphi$ and $\mathcal{I}, \alpha \models \psi$
- $\mathcal{I}, \alpha \models \varphi \lor \psi$ iff $\mathcal{I}, \alpha \models \varphi$ or $\mathcal{I}, \alpha \models \psi$
- $\mathcal{I}, \alpha \models \varphi \rightarrow \psi$ iff if $\mathcal{I}, \alpha \models \varphi$, then $\mathcal{I}, \alpha \models \psi$
- $\mathcal{I}, \alpha \models \varphi \leftrightarrow \psi$ iff $\mathcal{I}, \alpha \models \varphi$ iff $\mathcal{I}, \alpha \models \psi$
- $\mathcal{I}, \alpha \models \forall x \varphi$ iff $\mathcal{I}, \alpha[x/d] \models \varphi$ for all $d \in \mathcal{D}$
- $\mathcal{I}, \alpha \models \exists x \varphi$ iff $\mathcal{I}, \alpha[x/d] \models \varphi$ for some $d \in \mathcal{D}$
Examples

\[ D = \{d_1, \ldots, d_n\}, \quad n > 1 \]
\[ a^I = d_1 \]
\[ b^I = d_1 \]
\[ \text{Cat}^I = \{d_1\} \]
\[ \text{Red}^I = D \]
\[ \alpha = \{(x \mapsto d_1), (y \mapsto d_2)\} \]
\[ \Theta = \{\text{Cat}(a), \text{Cat}(b), \forall x(\text{Cat}(x) \rightarrow \text{Red}(x))\} \]

Questions:

\[ \mathcal{I}, \alpha \models \text{Cat}(b) \lor \neg \text{Cat}(b)? \]
Yes

\[ \mathcal{I}, \alpha \models \text{Cat}(x) \rightarrow \text{Cat}(y)? \]
Yes

\[ \mathcal{I}, \alpha \models \forall x(\text{Cat}(x) \rightarrow \text{Red}(x))? \]
Yes

\[ \mathcal{I}, \alpha \models \Theta? \quad \text{Yes} \]
Terminology

$I, \alpha$ is a **model** of $\varphi$ iff

$$I, \alpha \models \varphi.$$ 

A formula can be **satisfiable**, **unsatisfiable**, **falsifiable**, **valid**, $\ldots$

Formulae $\varphi$ and $\psi$ are **logically equivalent** (symb.: $\varphi \equiv \psi$) iff for all $I, \alpha$:

$$I, \alpha \models \varphi \text{ iff } I, \alpha \models \psi.$$ 

**Note**: $P(x) \not\equiv P(y)$!

**Logical implication** is also analogous to propositional logic:

$$\Theta \models \varphi \text{ iff for all } I, \alpha \text{ s.t. } I, \alpha \models \Theta \text{ also } I, \alpha \models \varphi.$$
Free and bound variables

Variables can be free or bound (by a quantifier) in a formula:

\[
\begin{align*}
\text{free}(x) &= \{x\} \\
\text{free}(f(t_1, \ldots, t_n)) &= \text{free}(t_1) \cup \cdots \cup \text{free}(t_n) \\
\text{free}(t_1 = t_2) &= \text{free}(t_1) \cup \text{free}(t_2) \\
\text{free}(P(t_1, \ldots, t_n)) &= \text{free}(t_1) \cup \cdots \cup \text{free}(t_n) \\
\text{free}(\neg \varphi) &= \text{free}(\varphi) \\
\text{free}(\varphi \ast \psi) &= \text{free}(\varphi) \cup \text{free}(\psi), \text{ for } \ast = \lor, \land, \rightarrow, \leftrightarrow \\
\text{free}(Qx \varphi) &= \text{free}(\varphi) \setminus \{x\}, \text{ for } Q = \forall, \exists
\end{align*}
\]

Example: \( \forall x (R(y, z) \land \exists y (\neg P(y, x) \lor R(y, z))) \)

Which occurrences are free, which are not free?
Open & closed formulae

- Formulae without free variables are called **closed formulae** or **sentences**. Formulae with free variables are called **open formulae**.

- Closed formulae are all we need when we want to state something about the world. Open formulae (and variable assignments) are only necessary for technical reasons (semantics of $\forall$ and $\exists$).

- Note that **logical equivalence**, **satisfiability**, and **entailment** are independent from variable assignments if we consider only closed formulae.

- For closed formulae, we omit $\alpha$ in connection with $\models$:

$$\mathcal{I} \models \phi.$$
4 Normal forms
Prenex Normal Form

The **prenex normal form** of a FOL formula has the following form:

\[
\text{quantifier prefix} + (\text{quantifier free}) \text{ matrix}
\]

Generate prenex normal form:

1. Eliminate → and ↔.
3. Moving quantifiers out (using a number of equivalences).

**Theorem**

*For each FOL formula, an equivalent formula in prenex normal form exists and can be effectively computed.*
Skolemization

We can further simplify formulae by eliminating existential quantifiers using fresh function symbols (Skolem functions).

**Theorem (Skolem normal form)**

Let $\varphi$ be a closed formula in prenex normal form with all variables pairwise distinct of the form $\varphi = \forall x_1 \ldots \forall x_i \exists y \psi$. Let $g_i$ be an $i$-ary function symbols not appearing in $\varphi$. Then $\varphi$ is satisfiable iff

$$\varphi' = \forall x_1 \ldots \forall x_i \psi[y/g_i(x_1, \ldots, x_i)]$$

is satisfiable.

**Proof idea.**

For each assignment to $x_1 \ldots x_i$, there is a value of $y [= g(x_1, \ldots, x_i)]$ and vice versa.
Skolem normal form

Skolem Normal Form

Prenex normal form without existential quantifiers.

Notation: $\varphi^*$ is SNF of $\varphi$

Theorem

For each closed formula $\varphi$, a corresponding SNF $\varphi^*$ can be effectively computed.

Example

$\exists x ((\forall x \ p(x)) \land \neg q(x))$
$\exists y ((\forall x \ p(x)) \land \neg q(y))$
$\exists y (\forall x \ (p(x) \land \neg q(y)))$
$\forall x (p(x) \land \neg q(g_0))$
5 Herbrand interpretations
Reducing FOL satisfiability to propositional satisfiability …

Idea 1: We use one particular interpretation which has as the universe of discourse all possible ground terms – and we add one constant if we do not have already one $\implies$ Herbrand universe

Example: $\forall x \forall y (\neg P(x, y) \lor R(g_2(x, y), x))$

$\mathcal{D}^H = \{a_0, g_2(a_0, a_0), g_2(a_0, g_2(a_0, a_0)), \ldots \}$

Idea 2: Function symbols are interpreted syntactically, predicate symbols are interpreted arbitrarily over this universe (each ground atom gets a truth value): $\implies$ Herbrand interpretation

\[ a^\mathcal{I} = a \]

\[ (f(t_1, \ldots, t_n))^\mathcal{I} = f(t_1, \ldots, t_n) \]

$\mathcal{I}$ could then be defined such that, e.g., $\mathcal{I} \not\models P(a_0, a_0)$, $\mathcal{I} \not\models P(a_0, g_2(a_0, a_0))$, etc.
Herbrand models and Herbrand expansions

**Theorem**

A formula $\varphi$ has a model iff it has a Herbrand model.

**Idea 3:** We expand each SNF-formula by substituting all variables by all possible terms $\leadsto$ Herbrand expansion $(E(\varphi))$

**Example:**

\[ \neg P(a_0, a_0) \lor R(g_2(a_0, a_0), a_0), \neg P(a_0, g_2(a_0, a_0)) \lor R(g_2(a_0, g_2(a_0, a_0)), a_0), \ldots \]

**Theorem**

A formula $\varphi$ is satisfiable if $E(\varphi)$ is satisfiable.
A reduction to a satisfiability problem with infinitely many formulae

- Note that the Herbrand universe can be infinite, therefore $E(\varphi)$ can be infinite!
- If the Herbrand base is finite there is no problem (well, . . .)
- Use $E(\varphi)$ in a “lazy” way, expand only as needed
- Semi-decision method for unsatisfiability
- In fact, unsatisfiability (and validity) in FOL is only semi-decidable (use e.g. PCP to prove)!
6 Further Theorems
Further theorems

Some corollaries from the previous theorems:

**Theorem (Compactness)**

Let $\Phi \cup \{\psi\}$ be a set of closed formulae.

(a) $\Phi \models \psi$ iff there exists a finite subset $\Phi' \subseteq \Phi$ s.t. $\Phi' \models \psi$.

(b) $\Phi$ is satisfiable iff each finite subset $\Phi' \subseteq \Phi$ is satisfiable.

**Theorem (Löwenheim-Skolem)**

Each countable set of closed formulae that is satisfiable is satisfiable on a countable domain.
7 Literature
Harry R. Lewis and Christos H. Papadimitriou.  
Elements of the Theory of Computation.  

Volker Sperschneider and Grigorios Antoniou.  
Logic – A Foundation for Computer Science.  
Addison-Wesley, Reading, MA, 1991 (Chapters 1–3).

Einführung in die mathematische Logik.  

U. Schöning.  
Logik für Informatiker.  
Spektrum-Verlag.