Knowledge Representation and Reasoning

Lecture 7:
Classical Logic I

dr Anna Maria Radzikowska
Warsaw University of Technology
Faculty of Mathematics and Information Science
E–mail: annrad@mini.pw.edu.pl
The notion of logic

By a *logic* we mean a triple $\text{Log} = (\mathcal{L}, \Sigma, \models)$, where

- $\mathcal{L}$ is the *language* of $\text{Log}$, i.e., the set of all formulas in $\text{Log}$
- $\Sigma$ is the class of all frames used for interpretation of formulas
- $\models: 2^\Sigma \rightarrow 2^\mathcal{L}$ is the consequence mapping which for each set $\mathcal{M}$ of frames determines the set of formulas *satisfied* in every frame from $\mathcal{M}$.

Classical logic:

- Propositional Calculus
- First Order Predicate Calculus.
Propositional Calculus
Propositional Calculus – Syntax

A language of a propositional calculus (PC) is determined by the following disjoint sets of symbols:

- a set $Var$ of propositional variables
- the truth constant $\top$
- logical connectives $\neg$ and $\rightarrow$
- parentheses ( and ).

The **language** of PC is the smallest set of the following expressions:

- $Var \subseteq \mathcal{L}$
- $\top \in \mathcal{L}$
- if $\alpha, \beta$ are formulas, then so are $\neg \alpha$ and $\alpha \rightarrow \beta$. 
Propositional Calculus – Syntax (cont.)

The remaining symbols are defines as:

- the truth constant \( \bot : \bot \overset{def}{=} \neg \top \)

- logical connectives:
  - disjunction: \( \alpha \lor \beta \overset{def}{=} \neg \alpha \rightarrow \beta \)
  - conjunction: \( \alpha \land \beta \overset{def}{=} \neg(\neg \alpha \lor \neg \beta) \)
  - equivalence: \( \alpha \equiv \beta \overset{def}{=} (\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha) \).
Semantics of $PC$

Let $\mathcal{L}$ be a language of propositional logic. An *interpretation* of $\mathcal{L}$ is a mapping $m : Var \rightarrow \{0, 1\}$.

The mapping $m$ are easily extended for the set $\mathcal{L}$ of all formulas.

A formula $\alpha \in \mathcal{L}$ is *true in* $m$ ($m$ is a *model* of $\alpha$), in symbols $m \models \alpha$, iff $m(\alpha) = 1$.

Let $\alpha \in \mathcal{L}$ be a formula. We say that $\alpha$ is

- **satisfiable** iff it has a model; otherwise it is called **unsatisfiable**
- **tautology**, written $\models \alpha$, iff every interpretation of $\mathcal{L}$ is a model of $\alpha$
## Most famous tautologies

| |= \( \alpha \lor \neg \alpha \) | **Excluded Middle Law** |
| |= \( \neg (\alpha \land \beta) \equiv \neg \alpha \lor \neg \beta \) | **De Morgan Law** |
| |= \( \neg (\alpha \lor \beta) \equiv \neg \alpha \land \neg \beta \) | **De Morgan Law** |
| |= \( \neg \neg \alpha \equiv \alpha \) | **Double Negation Law** |
| |= \( \alpha \lor (\beta \land \gamma) \equiv (\alpha \lor \beta) \land (\alpha \lor \gamma) \) | **Distributive Law** |
| |= \( \alpha \land (\beta \lor \beta) \equiv (\alpha \land \beta) \lor (\alpha \land \gamma) \) | **Distributive Law** |
| |= \( \alpha \land \beta \equiv \beta \land \alpha \) | **Commutative Law** |
| |= \( \alpha \lor \beta \equiv \beta \lor \alpha \) | **Commutative Law** |
| |= \( \alpha \rightarrow \beta \equiv \neg \beta \rightarrow \neg \alpha \) | **Contraposition Law** |
The validity problem is the task to determine whether or not a given formula is a tautology.

In virtue of the truth-table method we have the following fact.

**Theorem 7.1** The validity problem for classical propositional calculus is decidable.
Let $\mathcal{L}$ be a language of (classical) propositional logic. Any subset $T \subseteq \mathcal{L}$ is called a \textit{theory}.

For a set $T \subseteq \mathcal{L}$ of formulas and the set $\mathcal{M}$ of interpretations of $\mathcal{L}$, $\mathcal{M} \models T$ means that every formula $\alpha \in T$ is true in every interpretation $m \in \mathcal{M}$.

Two theories $T_1, T_1 \subseteq \mathcal{L}$ are called \textit{equivalent}, written $T_1 \Leftrightarrow T_2$, iff $\text{Mod}(T_1) = \text{Mod}(T_2)$, where $\text{Mod}(T)$ stands for the set of all models of $T$. 

A reasoning rule

A reasoning rule

\[ r = \frac{\alpha_1, \ldots, \alpha_n}{\gamma} \]

is a partial mapping \( r : \mathcal{L}^n \to \mathcal{L} \). For \( \alpha_1, \ldots, \alpha_n \) from the domain of \( r \), \( \alpha_1, \ldots, \alpha_n \) are premises of \( r \) and \( \gamma = r(\alpha_1, \ldots, \alpha_n) \) is the consequence of \( r \).

A reasoning rule \( \frac{\alpha_1, \ldots, \alpha_n}{\gamma} \) is called sound iff \( \{\alpha_1, \ldots, \alpha_n\} \models \gamma \).
A deduction system (axiomatization) is a triple \( DS = (L, A, R) \), where

- \( L \) is a language of propositional logic,
- \( A \subseteq L \) is the set of logical axioms, and
- \( R \) is the set of reasoning rules.

DS is called sound iff

- each formula \( \alpha \in A \) is a tautology
- each reasoning rule \( r \in R \) is sound.
Derivability

Let $DS = (\mathcal{L}, \mathcal{A}, \mathcal{R})$ be an axiomatization, $T \subseteq \mathcal{L}$, and let $\alpha \in \mathcal{L}$.

- A **formal proof** of $\alpha$ in $DS$ from $T$ is a sequence $(\alpha_0, \ldots, \alpha_k)$ of formulas such that:
  - $\alpha_0 \in \mathcal{A} \cup T$
  - $\alpha_n = \alpha$
  - for every $i = 1, \ldots, k$, either $\alpha_i \in \mathcal{A} \cup T$ or $\alpha_i$ is a direct consequence of $\alpha_0, \ldots, \alpha_{i-1}$ wrt some reasoning rule $r \in \mathcal{R}$.

- $\alpha$ is called **derivable** from $T$ wrt $DS$, written $T \vdash_{DS} \alpha$, iff there exists a formal proof of $\alpha$ from $T$ in $DS$.

- Derivability operator $Th$: for any $T \subseteq \mathcal{L}$, $Th(T) = \{ \alpha : T \vdash_{DS} \alpha \}$.

- $T$ is **consistent** iff $T \not\vdash \alpha$ for some formula $\alpha$. 
Soundness and Completeness

Let $DS = (\mathcal{L}, \mathcal{A}, \mathcal{R})$ be an axiomatization.

- $DS$ is **sound** iff for every theory $T \subseteq \mathcal{L}$ and for every $\alpha \in \mathcal{L}$,
  \[ T \vdash_{DS} \alpha \implies T \models \alpha \]

- $DS$ is called **complete** iff for every theory $T \subseteq \mathcal{L}$ and for every formula $\alpha \in \mathcal{L}$,
  \[ T \models \alpha \implies T \vdash_{DS} \alpha. \]
The most popular (sound and complete) axiomatization of propositional logic:

- **Logical axioms:**
  - $\top$
  - $\alpha \to (\beta \to \alpha)$
  - $(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$
  - $(\alpha \to \beta) \to ((\alpha \to \neg \beta) \to \neg \alpha)$

- **Reasoning rule:** *Modus Ponens* \[ \frac{\alpha, \alpha \to \beta}{\beta} \]
First Order Predicate Calculus
First–Order Predicate Calculus FOPC

An alphabet $\mathcal{A}$ of FOPC consists of:

- a denumerable set of variables $\text{Var}$
- a denumerable set of *function symbols* $\mathcal{F}$; for each $f \in \mathcal{F}$ there is uniquely assigned a non–negative integer called *arity* (number of arguments)
- a denumerable set $\mathcal{R}$ of *predicate (relation) symbols*; to each predicate $R \in \mathcal{R}$, there is uniquely assigned a non–negative integer called *arity* (number of arguments)
- the truth–constant $\top$
- logical connectives $\neg$ and $\rightarrow$
- a quantifier $\forall$
- parentheses $(, )$. 

dr Anna M. Radzikowska, Knowledge Representation and Reasoning 7, – p.16/30
Abbreviations

- Another truth–constant \( \textit{falsity} \): \( \bot \overset{\text{def}}{=} \neg \top \)

- The remaining logical connectives:

  \[
  \begin{align*}
  \alpha \lor \beta & \overset{\text{def}}{=} \neg \alpha \rightarrow \beta \\
  \alpha \lor \beta & \overset{\text{def}}{=} \neg (\neg \alpha \land \neg \beta) \\
  \alpha \equiv \beta & \overset{\text{def}}{=} (\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha)
  \end{align*}
  \]

- **Existential quantifier**: \( \exists x. \alpha(x) \overset{\text{def}}{=} \neg \forall x. \neg \alpha(x) \).
A term over $\mathcal{A}$ is an expression of the form:

- each variable $x \in \text{Var}$ is a term
- if $\tau_1, \ldots, \tau_n$ are terms, $n \geq 0$, and $f^{(n)} \in \mathcal{F}$ is a function symbol of the arity $n$, then $f^{(n)}(\tau_1, \ldots, \tau_n)$ is a term
- no other symbols are terms.

E.g. $x, x + y, \text{father}(x), \text{Child}(\text{John}, \text{Mary})$. 
Formulas of FOPC

- An **atomic formula** (atom) is an expression of the form:
  - $\top$ is an atom
  - if $\tau_1, \ldots, \tau_n$ are terms and $R^{(n)} \in \mathcal{R}$ is a predicate symbol of arity $n$, then $R^{(n)}(\tau_1, \ldots, \tau_n)$ is an atom
  - no other expressions are atoms.

- A **formula** is an expression of the form:
  - each atom is a formula
  - if $\alpha$ and $\beta$ are formulas, then so are $\neg\alpha$, $\alpha \land \beta$, $\alpha \lor \beta$, $\alpha \rightarrow \beta$, $\alpha \equiv \beta$, $\forall x.\alpha$
  - no other expressions are formulas.
For example,

\[ \text{Student(Brother}(x)\text{)} \]
\[ \forall x, y. \text{Likes}(x, y) \land \text{HasBirthday}(y) \rightarrow \text{GivesPresent}(x, y). \]

A formula is called **closed** (a sentence) iff all variables occurring in \( \alpha \) are bound by some quantifier in \( \alpha \); otherwise it is called **open**.

A variable \( x \) is **free** in a formula \( \alpha \) iff \( x \) is not bound by any quantifier in \( \alpha \).

For example,

\[ \forall x. \exists y. \text{Likes}(x, y) \quad \text{closed formula} \]
\[ \forall x. \text{IsFriend}(x, John) \rightarrow \text{Likes}(x, y) \quad \text{open formula}. \]

A **theory** of FOPC is a set of closed formulas.
Examples: Representation issues

Consider sentences:

- All students are adults.
- Some children do not learn English
- Each student of Computer Science is examined by a computer science researcher.

Representation in FOPC:

- $\forall x. \text{Student}(x) \rightarrow \text{Adult}(x)$
- $\exists x. \text{Child}(x) \land \text{LearnsEnglish}(x)$
- $\forall x. \text{StudentCS}(x) \rightarrow (\exists y. \text{Examines}(x, y) \land \text{ReasercherCS}(y))$. 
Semantics of FOPC

Let $\mathcal{L}$ be a language of FOPC. A *frame* for $\mathcal{L}$ is a structure $\Phi = (D, m)$ where

- $D$ is a nonempty domain
- $m$ is a mapping which assigns:
  - to each function symbol $f \in \mathcal{F}$ of arity $n$, a mapping $m(f) : D^n \to D$
  - to each predicate symbol $R \in \mathcal{R}$ of arity $n$, a relation $m(R) \subseteq D^n$. 
Valuation of terms

Let $\Phi = (D, m)$ be a frame for $\mathcal{L}$.

- An assignment over $\Phi$ is a mapping $a : Var \rightarrow D$.
  The set of all assignments over $\Phi$ will be denoted by $As(\Phi)$.

- Given a term $\tau$, a valuation of $\tau$ over $\Phi$ wrt $a \in As(\Phi)$, written $val^\Phi_a(\tau)$, is defined as:
  - if $\tau$ is a variable $x$, then $val^\Phi_a(x) = a(x)$
  - if $\tau_1, \ldots, \tau_n$ are terms, $f \in \mathcal{F}$ is a function symbol of arity $n$, and $\tau = f(\tau_1, \ldots, \tau_n)$, then
    $$val^\Phi_a(f(\tau_1, \ldots, \tau_n)) = m(f)(val^\Phi_a(\tau_1), \ldots, val^\Phi_a(\tau_n))$$

E.g., for $D = \mathbb{N}$, $+$ interpreted as usual addition, and $a$ given by $a(x) = 5$, $a(y) = 7$, we have:

$$val^\Phi_a(x + y) = val^\Phi_a(x) + val^\Phi_a(y) = 5 + 7 = 12.$$
Valuation of formulas

Let $\Phi = (D, m)$ be a frame for $\mathcal{L}$, $a \in As(\phi)$, and let $\alpha \in \mathcal{L}$.

Denote: $a_x = \{a' \in As(\Phi) : a(y) = a'(y) \text{ for any } y \neq x\}$.

A **valuation of $\alpha$ in $\Phi$ wrt $a$** is defined as follows:

- $\text{val}_a^\Phi(\top) = 1$
- $\text{val}_a^\Phi(R(\tau_1, \ldots, \tau_n)) = 1$ iff $(\text{val}_a^\Phi(\tau_1), \ldots, \text{val}_a^\Phi(\tau_n)) \in m(R)$
- $\text{val}_a^\Phi(\forall x.\alpha) = \min_{a_x \in As(\Phi)} \text{val}_{a_x}^\Phi(\alpha)$
For the remaining formulas their valuations are defined by:

- \( \text{val}_a^\Phi (\neg \alpha) = 1 - \text{val}_a^\Phi (\alpha) \)
- \( \text{val}_a^\Phi (\alpha \land \beta) = \min(\text{val}_a^\Phi (\alpha), \text{val}_a^\Phi (\beta)) \)
- \( \text{val}_a^\Phi (\alpha \lor \beta) = \max(\text{val}_a^\Phi (\alpha), \text{val}_a^\Phi (\beta)) \)
- \( \text{val}_a^\Phi (\alpha \rightarrow \beta) = 1 \text{ iff } \text{val}_a^\Phi (\alpha) = 0 \text{ or } \text{val}_a^\Phi (\beta) = 1 \)
- \( \text{val}_a^\Phi (\alpha \equiv \beta) = 1 \text{ iff } \text{val}_a^\Phi (\alpha) = \text{val}_a^\Phi (\beta) \)
- \( \text{val}_a^\Phi (\exists x. \alpha) = \max_{a, x \in As(\Phi)} \text{val}_a^\Phi (\alpha) \).
Satisfiability

Let $\Phi = (D, m)$ be a frame for $\mathcal{L}$ and let $a \in As(\Phi)$.

A formula $\alpha \in \mathcal{L}$ is called

- **satisfiable in $\Phi$ wrt $a$** iff $val^\Phi_a(\alpha) = 1$
- **satisfiable in $\Phi$** iff $val^\Phi_a(\alpha) = 1$ for every $a \in As(\Phi)$; if $\alpha$ is satisfiable in $\Phi$, then $\Phi$ is called a **model of $\alpha$**
- **tautology** iff for every frame $\Phi$, $\alpha$ is satisfiable in $\Phi$.

Accordingly, $\alpha$ is **unsatisfiable** iff it has no model.
Axiomatization of FOPC

Logical axioms:

- \( \top \)
- \( \alpha \rightarrow (\beta \rightarrow \alpha) \)
- \( (\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow (\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma) \)
- \( (\alpha \rightarrow \beta) \rightarrow ((\alpha \rightarrow \neg \beta) \rightarrow \neg \beta) \)
- \( \forall x. (\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \forall x. \beta) \), provided that there is no free occurrences of \( x \) in \( \alpha \)
- \( \forall x. \alpha(x) \rightarrow \alpha(\tau) \), where \( \tau \) is free for \( x \) in \( \alpha(x) \).

E.g., \( y \) is free for \( x \) in \( P(x) \), but if not free for \( x \) in \( \exists y. P(x, y) \).

Reasoning rules:

- (MP) \( \frac{\alpha, \alpha \rightarrow \beta}{\beta} \)
- (GEN) \( \frac{\alpha}{\forall x. \alpha} \), where \( x \) is not free in \( \alpha \).
Main theorems

- **Soundness:** \( T \vdash \alpha \iff T \models \alpha. \)
- **Completeness:** \( T \models \alpha \iff T \vdash \alpha. \)
- **Deduction:** \( T, \alpha \vdash \beta \iff T \vdash (\alpha \rightarrow \beta). \)
- **Monotonicity:** \( T_1 \subseteq T_2 \iff Th(T_1) \subseteq Th(T_2). \)
- **Compactness:** \( T \) is consistent iff each finite subtheory of \( T \) is consistent.

- \( \{\alpha_1, \ldots, \alpha_n\} \models \beta \iff \models \alpha_1 \land \ldots \land \alpha_n \rightarrow \beta \)
- \( T \models \alpha \iff T \cup \{\neg \alpha\} \) is unsatisfiable.
Recall:

*The validity problem is the task to determine whether or not a given formula is a tautology.*

**Theorem 7.2** The validity problem for FOPC is not decidable, but partly decidable.

In view of the above theorem, there exists an algorithm such that

- for every tautology it returns the answer **YES**,
- for some formula, which is not a tautology, it runs into infinite loop.
Thank you for your attention!

WARSZAWSKA UNIWERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Project is co-financed by European Union within European Social Fund