Fakultät Informatik, Institut für Künstliche Intelligenz, Professur Computational Logic

SEMINAR LOGIC-BASED KNOWLEDGE REPRESENTATION

Logic Recap

Sebastian Rudolph

https://iccl.inf.tu-dresden.de/web/Seminar_Logic-Based_Knowledge_Representation_(SS2024)

TU Dresden, 24th April, 2023

Current Seminar Plan (Maybe Subject to Change)

Date	Topic	Presenter
22.04.	Modal Logic – Semantics	Aidan
29.04.	Temporal Reasoning	Hamza
27.05. (!)	Epistemic Logics	Carlos
03.06.	NMR Introduction	Sepideh
17.06. (!)	Autoepistemic Logic	Abdul
24.06.	free slot	Latecomer 1
01.07.	free slot	Latecomer 2

Note: Register seminar with examination office.

Outline

We review some key concepts of classical logic:

- Syntax and Semantics of Propositional Logic
- Model Theory vs. Proof Theory
- Soundness and Completeness
- Syntax and Semantics of First-Order Logic

Motivation

In **Knowledge Representation and Reasoning** we want to . . .

. . . formally represent a collection of **propositions** believed by some **agent**,

and to **derive** new information from these propositions by applying reasoning techniques.

Logic allows us to . . .

. . . **formally** represent information in various logical systems,

and to **draw logical inferences** from given information.

Quiz: Logic Basics

Quiz: For each of the following statements, decide whether it holds.

1. In propositional logic (PL), $P \to Q$ is equivalent to $P \vee \neg Q$.
2. In PL, if φ is a tautology and $\mathcal{I}$ is an interpretation, then $\varphi^{\mathcal{I}} = \mathit{true}$.
3. In PL, if Δ and Γ are sets of formulas and φ is a formula, then $\Delta \models \varphi$ implies $\Delta \cup \Gamma \models \varphi$.
4. In first-order logic, $\neg\forall x.(P(x) \vee Q(x))$ is equivalent to $(\exists x.\neg P(x)) \wedge (\exists x.\neg Q(x))$.
5. In first-order logic, there is a proof system with a derivation relation $\vdash$ that coincides with the entailment relation $\models$.

Propositional Logic

Propositional Logic – Overview

- It is one of the simplest logics
- It can be used to write simple representations of a domain
- There exist reasoning algorithms that exhibit excellent performance in practice
- (Most of) you are already familiar with it.

Syntax: Propositional Alphabet

1. Propositional variables (**PL**):

   basic statements that can be true or false

2. The symbols $\top$ (“truth”) and $\bot$ (“falsehood”)

3. Propositional connectives:

   $\neg$ negation (not)
   $\wedge$ conjunction (and)
   $\vee$ disjunction (or)
   $\rightarrow$ implication (if ... then)
   $\leftrightarrow$ bi-directional implication (if and only if)

4. Punctuation symbols “(” and “)” can be used to avoid ambiguity

Semantics: Interpretations

Definition 2.1 (Interpretation):
An **interpretation** $\mathcal{I}$ assigns truth values to propositional variables:

$$\mathcal{I} : \textbf{PL} \rightarrow \{true, false\}$$

An interpretation for a (set of) formulas X interprets the propositional variables occurring in X.

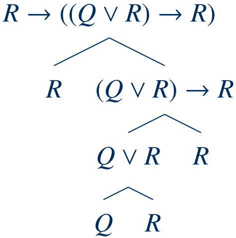
Example: An interpretation $\mathcal{I}$ for the formula $R \rightarrow ((Q \vee R) \rightarrow R)$:

$$\begin{aligned} R^{\mathcal{I}} &= true \\ Q^{\mathcal{I}} &= false \end{aligned}$$

A formula with n propositional variables has 2^n interpretations.

Semantics of Formulas

The truth value of the propositional variables in a formula α determines the truth value of α.



$$
\begin{aligned}
R^{\mathcal{I}} &= \mathit{true} \\
Q^{\mathcal{I}} &= \mathit{false} \\
(Q \vee R)^{\mathcal{I}} &= \mathit{true} \\
((Q \vee R) \rightarrow R)^{\mathcal{I}} &= \mathit{true} \\
(R \rightarrow ((Q \vee R) \rightarrow R))^{\mathcal{I}} &= \mathit{true}
\end{aligned}
$$

Definition 2.2 (Model): We say that $\mathcal{I}$ is a **model** of α iff $\mathcal{I}$ makes α true.

Using Propositional Logic for KR

Propositional Logic provides a simple KR language.

To write down a representation of our domain do the following:

1. Identify the relevant propositions:

Benign	The tumour is benign
$\mathit{Metastasis}$	The tumour has metastasis
$\mathit{Stage}4$	The tumour is in Stage 4
$\ldots$	

2. Express our knowledge using a set of formulas (knowledge base):

$$\mathit{Benign}$$
$$\mathit{Benign} \leftrightarrow \neg \mathit{Metastasis}$$
$$\mathit{Stage}4 \rightarrow \mathit{Metastasis}$$
$$\ldots$$

Reasoning with a Knowledge Base

Knowledge Base $\mathcal{K}_1$:

$$\begin{aligned} \mathit{Benign} \land \mathit{Stage}4 \\ \mathit{Benign} \leftrightarrow \neg \mathit{Metastasis} \\ \mathit{Stage}4 \rightarrow \mathit{Metastasis} \\ \ldots \end{aligned}$$

Knowledge Base $\mathcal{K}_2$:

$$\begin{aligned} \mathit{Benign} \\ \mathit{Benign} \leftrightarrow \neg \mathit{Metastasis} \\ \mathit{Stage}4 \rightarrow \mathit{Metastasis} \\ \ldots \end{aligned}$$

We would like to answer the following questions:

1. Do our KBs make sense?
   $\mathcal{K}_1$ seems contradictory
2. What is the implicit knowledge we can derive from our KBs?
   $\mathcal{K}_2$ seems to imply the formula $\neg \mathit{Stage}4$

Model Theory – Reasoning

Definition 2.3 (Semantic Consequence): Let Γ be a set of formulas and α a formula. We write $\Gamma \models \alpha$ if and only if every model of Γ is also a model of α.

Definition 2.4 (Tautology): Let α be some formula. We write $\models \alpha$ if and only if α is true in every interpretation.

Example from $\mathcal{K}_2$:

$$\{B, B \leftrightarrow \neg M, S4 \rightarrow M\} \models \neg S4$$

- Let $\mathcal{I}$ be a model of $\{B, B \leftrightarrow \neg M, S4 \rightarrow M\}$.
- Then $(B)^{\mathcal{I}} = \mathit{true}$, $(M)^{\mathcal{I}} = \mathit{false}$.
- Since $(S4 \rightarrow M)^{\mathcal{I}} = \mathit{true}$ and $(M)^{\mathcal{I}} = \mathit{false}$, it must hold that $(S4)^{\mathcal{I}} = \mathit{false}$.
- Thus $(\neg S4)^{\mathcal{I}} = \mathit{true}$.

Proof Theory

In proof theory:

- We do not consider the semantical interpretation of logical formulas.
- Rather, we are concerned with a syntactic description of our logical system . . .
- that allows to put our logic on an axiomatic foundations and to
- syntactically derive formulas from a set of axioms and inference rules.
- Some well-known systems include: Hilbert Systems, Natural Deduction and Tableau Calculi.

Definition 2.5 (Syntactic Consequence): Let Γ be a set of formulas and α a formula. We write $\Gamma \vdash \alpha$ if and only if there is a derivation with conclusion α from Γ.

Definition 2.6 (Theorem): If $\Gamma = \emptyset$, we write $\vdash \alpha$ and we say that α is a **theorem**.

Proof Theory – Hilbert System

Axioms:

Axiom 1. $\phi \rightarrow (\psi \rightarrow \phi)$
Axiom 2. $(\phi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \chi))$
Axiom 3. $(\neg\phi \rightarrow \neg\psi) \rightarrow ((\neg\phi \rightarrow \psi) \rightarrow \phi)$

Inference Rule: (Modus Ponens): From $\phi \rightarrow \psi$ and ϕ, infer ψ.

Example: Show $\phi \rightarrow \psi, \psi \rightarrow \chi \vdash \phi \rightarrow \chi$

$$\begin{array}{rl}
(\psi \rightarrow \chi) \rightarrow (\phi \rightarrow (\psi \rightarrow \chi)) & Ax1[\phi/\psi \rightarrow \chi; \psi/\phi] \\
(\phi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \chi)) & Ax2 \\
\psi \rightarrow \chi & Premise \\
\phi \rightarrow (\psi \rightarrow \chi) & MP(1,3) \\
(\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \chi) & MP(2,4) \\
\phi \rightarrow \psi & Premise \\
\phi \rightarrow \chi & MP(5,6)
\end{array}$$

Soundness and Completeness

For propositional logic (and other logical systems) we can show that the **semantic** and **syntactic entailment** coincide. That is: the relations "$\models$" and "$\vdash$" coincide.

We distinguish both directions:

Theorem 2.7 (Soundness): $\Gamma \vdash \alpha \Rightarrow \Gamma \models \alpha$

Theorem 2.8 (Completeness): $\Gamma \models \alpha \Rightarrow \Gamma \vdash \alpha$

[For proofs, see for instance: Dirk van Dalen, Logic and Structure (2008)]

Monotonicity

What does it mean for a logic to be monotone? Monotonicity is a property of the consequence relation:

Definition 2.9 (Monotonicity): Let Σ and Δ be sets of formulas and H be a formula. If $\Sigma \models H$ and $\Sigma \subseteq \Delta$ then $\Delta \models H$.

Example: Let $\Sigma = \{p, q\}, H = p, \Delta = \{p, q, r\}$. What if $\Delta = \{p, q, \neg p\}$?

What would we have to do to show that some entailment relation is non-monotonic? Find an example where:

- $\Sigma \models H$
- $\Sigma \subseteq \Delta$
- **But**: $\Delta \not\models H$

Limitations of Propositional Logic

Consider the following argument:

$$\begin{array}{rl} & \text{All men are mortal} \\ & \text{Socrates is a man} \\ \hline \therefore & \text{Socrates is mortal} \end{array}$$

The argument seems to be valid.

However, in propositional logic:

$$\begin{array}{rl} & p \\ & q \\ \hline \not\therefore & r \end{array}$$

First-Order Logic

FOL Syntax: Symbols

A first-order alphabet consists of

- Predicate Symbols, each with a fixed arity

$Arthritis$	Unary Predicate
$Affects$	Binary Predicate

- Function symbols, each with a fixed arity

$ssnOf$	Unary Function Symbol

- Constants: JohnSmith, MaryJones, JRA
- Variables: x, y, z
- Propositional connectives $\{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$
- Symbols $\top$ and $\bot$
- The universal and existential quantifiers: $\forall$, $\exists$

FOL Syntax: Terms

Terms stand for specific objects:

- Variables are terms
- Constants are terms
- The application of a function symbol to terms leads to a term

$\mathit{JohnSmith}$	stands for	the person named John Smith
$\mathit{ssnOf}(\mathit{JohnSmith})$	stands for	the ssn number of John Smith
x	stands for	some object (undetermined)
$\mathit{ssnOf}(x)$	stands for	some ssn number (undetermined)

FOL Syntax: Formulas

An atomic formula (atom) is of the form

$$P(t_1, \ldots, t_n) \quad P \text{ is an } n\text{-ary predicate}, t_i \text{ are terms}$$

Examples:

$\mathit{Child}(\mathit{JohnSmith})$	John Smith is a child
$\mathit{JuvenileArthritis}(\mathit{JRA})$	JRA is a juvenile arthritis
$\mathit{Affects}(\mathit{JRA}, \mathit{JohnSmith})$	John Smith is affected by JRA

An atom represents a simple statement:

- similar to atoms in propositional logic,
- but first-order atoms have finer-grained structure.

FOL Syntax: Formulas

Complex formulas:

- Every atom is a formula

$$Child(JohnSmith), Affects(x, JohnSmith)$$

- $\top$ and $\bot$ are formulas
- If α is a formula, then $\neg\alpha$ is a formula

$$\neg Affects(JRA, JohnSmith), \quad \neg Child(y)$$

- If α, β are formulas, $(\alpha \circ \beta)$ is a formula for $\{\circ \in \wedge, \vee, \rightarrow, \leftrightarrow\}$

$$Affects(JRA, y) \rightarrow Child(y) \vee Teenager(y)$$

- If α a formula and x a variable, $(\forall x.\alpha)$, $(\exists x.\alpha)$ are formulas

$$\forall y.(Affects(JRA, y)) \rightarrow Child(y) \vee Teenager(y))$$

$$\neg(\exists x.\exists y(JuvArthritis(x) \wedge Affects(x, y) \wedge Adult(y)))$$

FOL Syntax: Formulas

Intuitively, a free variable occurrence in a formula is one that does not appear in the scope of a quantifier:

$$Affects(JRA, \underline{y}) \rightarrow Child(\underline{y}) \vee Teenager(\underline{y})$$

$$\exists x.\big(JuvArthritis(x) \wedge Affects(x, \underline{y}) \wedge Adult(\underline{y})\big)$$

$$\exists x.\big(JuvArthritis(x)\big) \wedge Affects(\underline{x}, \underline{y}) \wedge Adult(\underline{y})$$

A variable occurrence is bound if it is not free.

A sentence is a formula with no free variable occurrences.

Example FOL Sentences

A juvenile disease affects only children or teenagers:

$$\forall x.\forall y.((\mathit{JuvDisease}(x) \land \mathit{Affects}(x, y)) \to \mathit{Child}(y) \lor \mathit{Teenager}(y))$$

Children and teenagers are not adults:

$$\forall x.((\mathit{Child}(x) \lor \mathit{Teenager}(x)) \to \neg \mathit{Adult}(x))$$

FOL Interpretations

As in PL, the meaning of sentences is given by interpretations.

An interpretation is a pair $\mathcal{I} = \langle D, \cdot^{\mathcal{I}} \rangle$ where:

- D is a non-empty set, called the interpretation domain.

$$D = \{u, v, w, s\}$$

- $\cdot^{\mathcal{I}}$ is the interpretation function and it associates:
  - With each constant c an object $c^{\mathcal{I}} \in D$.

$$JohnSmith^{\mathcal{I}} = u \quad MaryWilliams^{\mathcal{I}} = v \quad JRA^{\mathcal{I}} = w \quad \ldots$$

  - With each n-ary function symbol f, a function $f^{\mathcal{I}} : D^n \to D$.

$$ssnOf^{\mathcal{I}} = \{u \mapsto s, \ldots\}$$

  - With each n-ary predicate symbol P, a relation $P^{\mathcal{I}} \subseteq D^n$.

$$Child^{\mathcal{I}} = \{u, v\} \quad Adult^{\mathcal{I}} = \emptyset \quad Affects^{\mathcal{I}} = \{\langle w, u\rangle, \ldots\}$$

Evaluation of Terms

Terms are interpreted as elements of the interpretation domain.

We have already seen how to interpret constants

$$JohnSmith^{\mathcal{I}} = u \quad MaryWilliams^{\mathcal{I}} = v \quad JRA^{\mathcal{I}} = w \quad \ldots$$

To interpret terms, we need to interpret (free) variables by means of a mapping from variables to domain elements (an assignment)

Given $\mathcal{I}$ and assignment **a**, we can interpret any term.
Let $\mathcal{I}$ be as before and **a** map x to u:

$$\begin{aligned} JohnSmith^{\mathcal{I},\mathbf{a}} &= u \\ x^{\mathcal{I},\mathbf{a}} &= u \\ (ssnOf(x))^{\mathcal{I},\mathbf{a}} &= ssnOf^{\mathcal{I}}(u) = s \end{aligned}$$

Evaluation of Formulas

Given $\mathcal{I}$ and **a**, a formula is interpreted as either **true** or **false.**

Atomic formulas:

$$P(t_i, \ldots, t_n)^{\mathcal{I},\mathbf{a}} = \textbf{true} \quad \text{iff} \quad \langle t_i^{\mathcal{I},\mathbf{a}}, \ldots, t_n^{\mathcal{I},\mathbf{a}} \rangle \in P^{\mathcal{I}}$$

Examples:

$$\begin{aligned} \mathit{Child}(\mathit{JohnSmith})^{\mathcal{I},\mathbf{a}} &= \textbf{true} \quad \text{since} \quad \mathit{JohnSmith}^{\mathcal{I},\mathbf{a}} = u \text{ and } \mathit{Child}^{\mathcal{I}} = \{u, v\} \\ \mathit{Affects}(\mathit{JRA}, x)^{\mathcal{I},\mathbf{a}} &= \textbf{true} \quad \text{since} \quad \mathit{JRA}^{\mathcal{I},\mathbf{a}} = w, \quad x^{\mathcal{I},\mathbf{a}} = u \text{ and } \mathit{Affects}^{\mathcal{I}} = \{\langle w, u \rangle\} \end{aligned}$$

Propositional connectives are interpreted as usual:

$$\begin{aligned} (\neg \mathit{Child}(\mathit{JohnSmith}))^{\mathcal{I},\mathbf{a}} &= \textbf{false} \\ (\mathit{Affects}(\mathit{JRA}, x) \wedge \mathit{Child}(\mathit{JohnSmith}))^{\mathcal{I},\mathbf{a}} &= \textbf{true} \\ (\mathit{Child}(\mathit{JohnSmith}) \rightarrow \neg \mathit{Child}(\mathit{JohnSmith}))^{\mathcal{I},\mathbf{a}} &= \textbf{false} \end{aligned}$$

Evaluation of Formulas

Given $\mathcal{I}$ and **a**, a formula is interpreted as either **true** or **false.**

Existential quantifiers:

$$(\exists x.\mathit{Affects}(\mathit{JRA}, x))^{\mathcal{I},\mathbf{a}_\emptyset} = \textbf{true}$$

since there exists an assignment **a** extending $\mathbf{a}_\emptyset$ such that $\mathit{Affects}(\mathit{JRA}, x)^{\mathcal{I},\mathbf{a}} = \textbf{true}$

Universal quantifiers:

$$(\forall x.\mathit{Affects}(\mathit{JRA}, x))^{\mathcal{I},\mathbf{a}_\emptyset} = \textbf{false}$$

since it is not true that, for any assignment **a** extending $\mathbf{a}_\emptyset$, $\mathit{Affects}(\mathit{JRA}, x)^{\mathcal{I},\mathbf{a}} = \textbf{true}$.

Evaluation of Sentences

For interpreting a sentence φ under $\mathcal{I}$, **a**, the top-level assignment **a** is irrelevant.

Theorem 2.10: For any sentence φ and assignments **a**, **a**′, we have $\varphi^{\mathcal{I},\mathbf{a}} = \varphi^{\mathcal{I},\mathbf{a}'}$.

Example: Consider the sentence

$$\forall x \forall y.((\mathit{JuvDisease}(x) \land \mathit{Affects}(x,y)) \to (\mathit{Child}(y) \lor \mathit{Teenager}(y)))$$

Assume the interpretation $\mathcal{I}$ with $\mathbf{D} = \{u, v, w\}$ given as follows:

$$\mathit{JuvDisease}^{\mathcal{I}} = \{u\} \qquad \mathit{Child}^{\mathcal{I}} = \{w\} \qquad \mathit{Teenager}^{\mathcal{I}} = \emptyset \qquad \mathit{Affects}^{\mathcal{I}} = \{\langle u, w\rangle\}$$

φ without quantifiers must evaluate to true in $\mathcal{I}$ for all valuations $\mathbf{a} : \{x, y\} \to \mathbf{D}$.
Example for $\mathbf{a}_1 = \{x \mapsto u, y \mapsto v\}$:

$$(\mathit{JuvDisease}(x)^{\mathcal{I},\mathbf{a}_1} \land \mathit{Affects}(x,y)^{\mathcal{I},\mathbf{a}_1}) \to (\mathit{Child}(y)^{\mathcal{I},\mathbf{a}_1} \lor \mathit{Teenager}(y)^{\mathcal{I},\mathbf{a}_1})$$

$$(\mathbf{true} \land \mathbf{false}) \to (\mathbf{true} \lor \mathbf{false})$$

$$\mathbf{true}$$

Propositional vs. FOL Interpretations

More complicated to give meaning to FOL than to PL formulas:

$$JuvDisease \rightarrow AffectsChild \vee AffectsTeenager \quad (PL)$$

$$\forall x.\forall y.((JuvDisease(x) \wedge Affects(x, y)) \rightarrow (Child(y) \vee Teenager(y))) \quad (FOL)$$

PL Interpretations

- Assigns truth values to atoms
- The truth value of complex formulas determined by induction

Example formula has 8 possible interpretations and 7 models

FOL interpretations

- Specify the domain for quantifiers to quantify over
- Interpret constants, predicates, functions
- Assign objects to variables

Example formula has ∞ possible interpretations and ∞ models

Summary and Outlook

We reviewed syntax and semantics of PL and FOL.

Logical systems can be described from two points of view:

- model theory
- proof theory

For PL, FOL, and many other logics these points of view coincide (soundness and completeness).

PL, FOL, and many other logics are monotonic.

Open questions:

- How can we define systems other than PL and FOL? (Next session)
- What do non-monotonic logics look like? (In a few weeks)