Artificial Intelligence 1

Prof. Dr. Frank Hopfgartner
Dr. Matthias Horbach

Institute for Web Science and Technologies (WeST)
University of Koblenz

Overview

Symbolic AI vs. subsymbolic AI

- the area of "symbolic AI" is about formalisation of thought/reasoning through symbolic manipulation
  - Relationships are modelled through "rules"
  - The motivation behind this is not only "simulation" of intelligent processes but also "explanation" (white box approach)
- Another approach is "subsymbolic AI"
  - Motivation is mainly about simulation/imitation
  - Example: neural networks
  - Approaches are usually black boxes
  - but, in certain domains, quite accurate
- This course will focus solely on symbolic AI
- For subsymbolic AI see the course "Machine learning"

→ the foundation for symbolic AI is *logic*

Logic and formal systems

- Logics can be used to model reasoning processes
- A (logical) statement is an abstract construct that is either TRUE or FALSE
- Formal logic is about making statement about statements
- Example:

  Anna is a student

  All students are humans

  $\rightarrow$ Anna is a human
- Analysis:
  - Given that the statement "Anna is a student" is true
  - Given that the statement "All students are humans" is true
  - then the statement "Anna is a human" is a necessarily true statement

Logics and formal systems: structure

Every logic (=formal system) has the following components:

1. Syntax: What are the possible statements?
   1.1 Signature: What symbols are allowed?
   ($S = \{\mathrm{Anna}, \mathrm{human}, \mathrm{student}\}$)
   1.2 Grammar: how can symbols be combined in order to obtain complex statements? (student $\Rightarrow$ human)
2. Semantics: Which are the "true" statements? What is the relationship between true statements?
   2.1 Interpretations (or "possible worlds"): assign meaning to symbols of language.
   2.2 Models: What are the interpretations in which a statement is true?
   2.3 Entailment: when is one statement entailed (follows logically from) another?
3. Calculus: How can entailment be implemented?

Overview

Propositional logic - overview

- Propositional logic is one of the first formal systems for reasoning (Aristoteles, 384–322 BC)
- Propositional logic builds on atomic statements such as

$$\begin{aligned} A &= \text{“Paris is the capital of France”} \\ B &= \text{“A whale is a fish”} \\ C &= \text{“For every number } x \in \mathbb{R},\ x^2 \geq 0\text{”} \end{aligned}$$

- Atomic statements can be combined with $\wedge$ (AND), $\vee$ (OR), and $\neg$ (NOT) to obtain more complex statements
- A sentence is either true or false.
- For reasons of simplicity we always assume that our language always contains the following to special atomic statements:
  - $\top$: “The true statement” (is always true)
  - $\bot$: “The wrong statement” (is never true)

Propositional logic - syntax 1/4

Definition

A *propositional signature* At is a set of atoms (propositions).

Example

$\mathrm{At}_1 = \{a, b, c, d, e\}$
$\mathrm{At}_2 = \{$"Anna is a student", "Anna is a human"$\}$

Definition (Syntax 1)

Let At be a propositional signature. The *propositional language* $\mathcal{L}(\mathrm{At})$ is the minimal set $\mathcal{L}$ with

1. $\mathrm{At} \subseteq \mathcal{L}$
2. $\top, \bot \in \mathcal{L}$ (tautology and contradiction) and
3. for all $\phi, \psi \in \mathcal{L}$
   3.1 $\phi \land \psi \in \mathcal{L}$
   3.2 $\phi \lor \psi \in \mathcal{L}$
   3.3 $\neg\phi \in \mathcal{L}$

Propositional logic - syntax 2/4

Alternative definition of the syntax with grammar:

Definition (Syntax 2)

Let At be a propositional signature. The *propositional language* $\mathcal{L}(\mathrm{At})$ contains exactly those formula ϕ that can be constructed using the grammar

$$\phi \to a \mid \top \mid \bot \mid \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi$$

with $a \in \mathrm{At}$.

Remark

In order to structure formulas we also allow the use of parentheses "(" and ")" (which are formally also part of the syntax).

Example

Let $\mathrm{At}_1 = \{a, b, c, d, e\}$. Then

$$\phi_1 = a \wedge b$$
$$\phi_2 = a \vee b \wedge \neg c$$
$$\phi_3 = \neg\neg a \vee (\neg b \wedge e) \vee \neg\neg(d \vee c)$$

are formulas in $\mathcal{L}(\mathrm{At}_1)$.

Remark

The symbols $\wedge$, $\vee$, and $\neg$ are expressive enough to define other common operators:

$$\phi \Rightarrow \psi := \neg\phi \vee \psi$$

$$\phi \Leftrightarrow \psi := (\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$$

Another remark

Actually, already $\wedge$ and $\neg$ are sufficient as $\phi \vee \psi = \neg(\neg\phi \wedge \neg\psi)$.

- Interpretations assign meaning to propositional formulas.
- An interpretation describes a *possible* world by assigning a truth value to each atom.

Definition

Let At be a propositional signature. A *propositional interpretation* I is a function $I : \mathrm{At} \to \{\mathrm{true}, \mathrm{false}\}$.

Let $\mathrm{Int}_{\mathrm{At}}$ be the set of all propositional interpretations.

Example

Let $\mathrm{At}_3 = \{a, b, c\}$. Then I defined through

$$I(a) = \mathrm{true} \qquad I(b) = \mathrm{false} \qquad I(c) = \mathrm{true}$$

is a propositional interpretation.

- Propositional interpretations can be abbreviated by *complete conjunctions*:

$$I(a) = \text{true} \qquad I(b) = \text{false} \qquad I(c) = \text{true}$$

  becomes $a\overline{b}c$ (negation is represented by overlining)

- All possible interpretations for $\text{At}_3 = \{a, b, c\}$:

Int_{At_3}	a	b	c	Abbreviation
I_1	true	true	true	abc
I_2	true	true	false	$ab\overline{c}$
I_3	true	false	true	$a\overline{b}c$
I_4	true	false	false	$a\overline{b}\overline{c}$
I_5	false	true	true	$\overline{a}bc$
I_6	false	true	false	$\overline{a}b\overline{c}$
I_7	false	false	true	$\overline{a}\overline{b}c$
I_8	false	false	false	$\overline{a}\overline{b}\overline{c}$

Propositional logic - semantics 3/9

- Interpretations tell which atoms are true statements wrt. this interpretation
- This is formalised through a *satisfaction relation* $\models$
- We say an interpretation I satisfies an atom $a \in \mathrm{At}$, written $I \models a$, iff $I(a) = \mathrm{true}$.
- What about arbitrary formulas $\phi \in \mathcal{L}(\mathrm{At})$?

Definition

Let At be a propositional signature, $I \in \mathrm{Int}_{\mathrm{At}}$ and $\phi, \psi \in \mathcal{L}(\mathrm{At})$. Define recursively

$$
\begin{array}{ll}
I \models \phi & \text{for } \phi = a \in \mathrm{At} \text{ iff. } I(a) = \mathrm{true} \\
I \models \phi \vee \psi & \text{iff } I \models \phi \text{ or } I \models \psi \\
I \models \phi \wedge \psi & \text{iff } I \models \phi \text{ and } I \models \psi \\
I \models \neg\phi & \text{iff } I \not\models \phi
\end{array}
$$

In addition, define $I \models \top$ and $I \not\models \bot$ for every interpretation.

Example

Let I be defined through

$$I(a) = \text{true} \qquad I(b) = \text{false} \qquad I(c) = \text{true}$$

and consider ϕ defined through

$$\phi = \neg a \vee (\neg b \wedge c)$$

Is it true that $I \models \phi$?

Propositional logic - semantics 5/9

$$I(a) = \text{true} \qquad I(b) = \text{false} \qquad I(c) = \text{true}$$

Is it true that $I \models \phi$?

$$\begin{aligned}
& I \models \phi \\
\text{iff} \quad & I \models \neg a \vee (\neg b \wedge c) \\
\text{iff} \quad & I \models \neg a \text{ or } I \models \neg b \wedge c \\
\text{iff} \quad & I \not\models a \text{ or } I \models \neg b \wedge c \\
\text{iff} \quad & \text{not } I(a) = \text{true or } I \models \neg b \wedge c \\
\text{iff} \quad & I(a) = \text{false or } I \models \neg b \wedge c \\
\text{iff} \quad & \mathit{FALSE} \textit{ or } I \models \neg b \wedge c \\
\text{iff} \quad & \mathit{FALSE} \textit{ or } I \models \neg b \text{ and } I \models c \\
\text{iff} \quad & \mathit{FALSE} \textit{ or } I \not\models b \text{ and } I \models c \\
\text{iff} \quad & \mathit{FALSE} \textit{ or } \text{not } I(b) = \text{true and } I \models c
\end{aligned}$$

Propositional logic - semantics 6/9

$I(a) = \text{true} \qquad I(b) = \text{false} \qquad I(c) = \text{true}$

$$
\begin{array}{ll}
\text{iff} & \mathit{FALSE}\ \mathit{or}\ \text{not}\ I(b) = \text{true and}\ I \models c \\
\text{iff} & \mathit{FALSE}\ \mathit{or}\ I(b) = \text{false and}\ I \models c \\
\text{iff} & \mathit{FALSE}\ \mathit{or}\ \mathit{TRUE}\ \text{and}\ I \models c \\
\text{iff} & \mathit{FALSE}\ \mathit{or}\ \mathit{TRUE}\ \text{and}\ I(c) = \text{true} \\
\text{iff} & \mathit{FALSE}\ \mathit{or}\ \mathit{TRUE}\ \text{and}\ \mathit{TRUE} \\
\text{iff} & \mathit{TRUE}
\end{array}
$$

Therefore $I \models \phi$.

Propositional logic - semantics 7/9

An interpretation I with $I \models \phi$ is also called *model* of ϕ.

Definition

Let At be a propositional signature and $\phi \in \mathcal{L}(\mathrm{At})$. The set of *models* $\mathrm{Mod}(\phi)$ of ϕ is defined through

$$\mathrm{Mod}(\phi) = \{I \in \mathrm{Int}_{\mathrm{At}} \mid I \models \phi\}$$

Example

Let $\mathrm{At}_3 = \{a, b, c\}$.

$$\begin{aligned}
\mathrm{Mod}(a) &= \{abc, ab\overline{c}, a\overline{b}c, a\overline{b}\overline{c}\} \\
\mathrm{Mod}(a \wedge b) &= \{abc, ab\overline{c}\} \\
\mathrm{Mod}(\neg c) &= \{ab\overline{c}, a\overline{b}\overline{c}, \overline{a}b\overline{c}, \overline{a}\overline{b}\overline{c}\} \\
\mathrm{Mod}(a \vee \neg b) &= \{abc, ab\overline{c}, a\overline{b}c, a\overline{b}\overline{c}, \overline{a}\overline{b}c, \overline{a}\overline{b}\overline{c}\}
\end{aligned}$$

Proposition

Let At be a propositional signature and $\phi, \psi \in \mathcal{L}(\mathsf{At})$. Then

1. $\mathsf{Mod}(\phi \vee \psi) = \mathsf{Mod}(\phi) \cup \mathsf{Mod}(\psi)$
2. $\mathsf{Mod}(\phi \wedge \psi) = \mathsf{Mod}(\phi) \cap \mathsf{Mod}(\psi)$
3. $\mathsf{Mod}(\neg\phi) = \mathsf{Int}_{\mathsf{At}} \setminus \mathsf{Mod}(\phi)$

Proof.

1.:

$$\begin{aligned} \mathsf{Mod}(\phi \vee \psi) &= \{I \in \mathsf{Int}_{\mathsf{At}} \mid I \models \phi \vee \psi\} \\ &= \{I \in \mathsf{Int}_{\mathsf{At}} \mid I \models \phi \text{ or } I \models \psi\} \\ &= \{I \in \mathsf{Int}_{\mathsf{At}} \mid I \models \phi\} \cup \{I \in \mathsf{Int}_{\mathsf{At}} \mid I \models \psi\} \\ &= \mathsf{Mod}(\phi) \cup \mathsf{Mod}(\psi) \end{aligned}$$

Analogous for 2. and 3. □

Propositional logic - semantics 9/9

Proposition

It holds

1. $\mathrm{Mod}(\top) = \mathrm{Int}_{\mathrm{At}}$
2. $\mathrm{Mod}(\bot) = \emptyset$

Proof.

$$
\begin{aligned}
\mathrm{Mod}(\top) &= \{I \in \mathrm{Int}_{\mathrm{At}} \mid I \models \top\} \\
&= \{I \in \mathrm{Int}_{\mathrm{At}}\} \\
&= \mathrm{Int}_{\mathrm{At}} \\
\mathrm{Mod}(\bot) &= \{I \in \mathrm{Int}_{\mathrm{At}} \mid I \models \bot\} \\
&= \emptyset
\end{aligned}
$$

□

Propositional logic - semantics and inference 1/3

- Interpretations and models only define meaning (semantics) of formulas.
- We are mainly interested in *inference*, i. e., when does a formula ψ follow logically from another formula ϕ?

Definition

Let At be a propositional signature and $\phi, \psi \in \mathcal{L}(\mathrm{At})$. We say that ψ follows logically from ϕ (or ϕ entails ψ), written $\phi \vdash \psi$, iff

$$\mathsf{Mod}(\phi) \subseteq \mathsf{Mod}(\psi)$$

Propositional logic - semantics and inference 2/3

Example

Let $\mathrm{At}_3 = \{a, b, c\}$. Is it true that $a \wedge \neg b \vdash a$?

- We already know that $\mathrm{Mod}(\phi \wedge \psi) = \mathrm{Mod}(\phi) \cap \mathrm{Mod}(\psi)$, so $\mathrm{Mod}(a \wedge \neg b) = \mathrm{Mod}(a) \cap \mathrm{Mod}(\neg b)$
- Hence $\mathrm{Mod}(a \wedge \neg b) \subseteq \mathrm{Mod}(a)$ and therefore $a \wedge \neg b \vdash a$

Example

Let $\mathrm{At}_3 = \{a, b, c\}$. It is true that $a \vee b \vdash a$?

- No, because there is a model of $a \vee b$ that is not a model of a:

$$\bar{a}bc \models a \vee b$$
$$\bar{a}bc \not\models a$$

- Hence $\mathrm{Mod}(a \vee b) \not\subseteq \mathrm{Mod}(a)$

Propositional logic - semantics and inference 3/3

Proposition

Let At be a propositional signature and $\phi, \psi \in \mathcal{L}(\mathsf{At})$. Then $\phi \vdash \psi$ iff

$$\text{For all } I \in \mathsf{Int}_{\mathsf{At}},\ \text{if } I \models \phi \text{ then } I \models \psi$$

Proof.

Let $\phi \vdash \psi$. Then $\mathsf{Mod}(\phi) \subseteq \mathsf{Mod}(\psi)$. This is equivalent to: for all $I \in \mathsf{Int}_{\mathsf{At}}$, if $I \models \phi$ then $I \models \psi$. □

Definition

Let At be a propositional signature and $\phi, \psi \in \mathcal{L}(\mathrm{At})$. We say that ϕ and ψ are *equivalent*, written $\phi \equiv \psi$, iff

$$\phi \vdash \psi \qquad \text{and} \qquad \psi \vdash \phi$$

Remark

$\phi \equiv \psi$ iff

- $\phi \vdash \psi$ and $\psi \vdash \phi$ iff
- $\mathrm{Mod}(\phi) = \mathrm{Mod}(\psi)$ iff
- For all $I \in \mathrm{Int}_{\mathrm{At}}$, $I \models \phi \leftrightarrow I \models \psi$ iff
- $\top \vdash \phi \Leftrightarrow \psi$

Propositional logic - equivalence 2/2

Example

Let $\mathrm{At}_3 = \{a, b, c\}$. Then

$$\neg\neg a \equiv a$$
$$\neg(a \wedge b) \equiv \neg a \vee \neg b$$
$$a \wedge \neg a \equiv \bot$$
$$a \vee \neg a \equiv \top$$

Propositional logic - inference

Proposition

Let At be a propositional signature and $\phi, \psi \in \mathcal{L}(\text{At})$. Then

$$\phi \vdash \psi \qquad \text{iff} \qquad \phi \wedge \neg\psi \vdash \bot$$

Proof.

Assumption: $\phi \vdash \psi$

$$\begin{aligned}
&\iff && \text{Mod}(\phi) \subseteq \text{Mod}(\psi) \\
&\iff && \text{Mod}(\phi) \cap (\text{Int}_{\text{At}} \setminus \text{Mod}(\psi)) = \emptyset \\
&\iff && \text{Mod}(\phi) \cap \text{Mod}(\neg\psi) = \emptyset \\
&\iff && \text{Mod}(\phi \wedge \neg\psi) = \emptyset \\
&\iff && \text{Mod}(\phi \wedge \neg\psi) \subseteq \text{Mod}(\bot) \\
&\iff && \phi \wedge \neg\psi \vdash \bot
\end{aligned}$$

□

Propositional logic - monotony

Propositional logic is *monotonous* in the sense that inferences are maintained when information is added.

Proposition

Let At be a propositional signature and $\alpha, \beta, \phi \in \mathcal{L}(\mathrm{At})$. Then

$$\text{If } \alpha \vdash \phi \qquad \text{then} \qquad \alpha \wedge \beta \vdash \phi$$

Proof.

If $\alpha \vdash \phi$ then $\mathrm{Mod}(\alpha) \subseteq \mathrm{Mod}(\phi)$. It also holds $\mathrm{Mod}(\alpha \wedge \beta) \subseteq \mathrm{Mod}(\alpha)$. Together we get $\mathrm{Mod}(\alpha \wedge \beta) \subseteq \mathrm{Mod}(\phi)$ and $\alpha \wedge \beta \vdash \phi$. $\square$

Propositional logic - trivialisation

- Propositional logic is *explosive* in the sense that inconsistency allows entailment of everything
- This phenomenon is also called *ex falso quodlibet* (lat. from falsehood, anything follows)

Proposition

Let At be a propositional signature and $\phi, \psi \in \mathcal{L}(\mathrm{At})$. Then

$$\text{If } \phi \equiv \perp \qquad \text{then} \qquad \phi \vdash \psi$$

Proof.

If $\phi \equiv \perp$ then $\mathrm{Mod}(\phi) = \emptyset$ and therefore $\mathrm{Mod}(\phi) \subseteq \mathrm{Mod}(\psi)$ for every $\psi \in \mathcal{L}(\mathrm{At})$. □

Example

$a \wedge \neg a \vdash b$

Propositional logic - further concepts

Let At be a propositional signature and $\phi \in \mathcal{L}(\mathrm{At})$.

- ϕ is *satisfiable* (consistent) if there is $I \in \mathrm{Int}_{\mathrm{At}}$ with $I \models \phi$
- ϕ is *unsatisfiable* (inconsistent) if there is no $I \in \mathrm{Int}_{\mathrm{At}}$ with $I \models \phi$
- ϕ is *valid* (tautological), if for all $I \in \mathrm{Int}_{\mathrm{At}}$ we have $I \models \phi$

Propositional logic - conjunctive normal form

- A *literal* is either an atom $a \in \text{At}$ or a negated atom $\neg a$
- A *clause* is a disjunction $\phi_1 \vee \ldots \vee \phi_n$ with literals $\phi_1, \ldots, \phi_n$
- A formula ϕ is in *conjunctive normal form* (CNF) if $\phi = \psi_1 \wedge \ldots \wedge \psi_m$ and every ψ_i is a clause

Proposition

Every formula $\phi \in \mathcal{L}(\text{At})$ can be rewritten in CNF, i. e., there is a formula ϕ' in CNF with $\phi \equiv \phi'$.

Without proof

Propositional logic - SAT

- The decision problem

  Given ϕ *in CNF, decide whether there is some* $I \in \mathrm{Int}_{\mathrm{At}}$ *with* $I \models \phi$

  is called the *satisfiability problem* (SAT) and *the* pivotal decision problem in theoretical computer science
- SAT is NP-complete: Solutions can be efficiently checked but not, as far as we know, efficiently found. Furthermore, every NP problem can be efficiently reduced to SAT.
- SAT solver can be used for reasoning; given a query "$\phi \vdash \psi$?"
  1. Transform $\phi \wedge \neg\psi$ into CNF
  2. Is $\phi \wedge \neg\psi$ not satisfiable then $\phi \vdash \psi$
- Implementations/systems:
  - SAT competition: `http://satcompetition.org`
  - SAT solver MapleSAT (high performant prover): `https://sites.google.com/a/gsd.uwaterloo.ca/maplesat/`
  - Tweety: `http://tweetyproject.org`