Foundations of Artificial Intelligence

29. Propositional Logic: Basics

Malte Helmert

University of Basel

April 15, 2019

Classification

classification:

> Propositional Logic
> environment:
> - ▶ static vs. dynamic
> - ▶ deterministic vs. non-deterministic vs. stochastic
> - ▶ fully vs. partially vs. not observable
> - ▶ discrete vs. continuous
> - ▶ single-agent vs. multi-agent
>
> problem solving method:
> - ▶ problem-specific vs. general vs. learning

(applications also in more complex environments)

Propositional Logic: Overview

Chapter overview: propositional logic
- ▶ 29. Basics
- ▶ 30. Reasoning and Resolution
- ▶ 31. DPLL Algorithm
- ▶ 32. Local Search and Outlook

29.1 Motivation

Propositional Logic: Motivation

propositional logic

- ▶ modeling and representing problems and knowledge
- ▶ basics for general problem descriptions and solving strategies
  (⤳ automated planning ⤳ later in this course)
- ▶ allows for automated reasoning

German: Aussagenlogik, automatisches Schliessen

Relationship to CSPs

- ▶ previous topic: constraint satisfaction problems
- ▶ satisfiability problem in propositional logic can be viewed as
  non-binary CSP over $\{\mathbf{F}, \mathbf{T}\}$
- ▶ formula encodes constraints
- ▶ solution: satisfying assignment of values to variables
- ▶ SAT algorithms for this problem: ⤳ DPLL (Wednesday)

Propositional Logic: Description of State Spaces

propositional variables for missionaries and cannibals problem:

```
two-missionaries-on-left-shore
one-cannibal-on-left-shore
boat-on-left-shore
...
```

- ▶ problem description for general problem solvers
- ▶ states represented as truth values of atomic propositions

German: Aussagenvariablen

Propositional Logic: Intuition

propositions: atomic statements over the world
that cannot be divided further

Propositions with logical connectives like
"and", "or" and "not" form the propositional formulas.

German: logische Verknüpfungen

29.2 Syntax

Syntax

Σ alphabet of propositions
(e.g., $\{P, Q, R, \dots\}$ or $\{X_1, X_2, X_3, \dots\}$).

Definition (propositional formula)
- ▶ $\top$ and $\bot$ are formulas.
- ▶ Every proposition in Σ is an (atomic) formula.
- ▶ If φ is a formula, then $\neg\varphi$ is a formula (negation).
- ▶ If φ and ψ are formulas, then so are
  - ▶ $(\varphi \wedge \psi)$ (conjunction)
  - ▶ $(\varphi \vee \psi)$ (disjunction)
  - ▶ $(\varphi \rightarrow \psi)$ (implication)

German: aussagenlogische Formel, atomare Formel, Konjunktion,
Disjunktion, Implikation

binding strength: $(\neg) > (\wedge) > (\vee) > (\rightarrow)$
(may omit redundant parentheses)

29.3 Semantics

Semantics

A formula is true or false,
depending on the interpretation of the propositions.

Semantics: Intuition
▶ A proposition p is either true or false.
The truth value of p is determined by an interpretation.

▶ The truth value of a formula follows from
the truth values of the propositions.

Example
$\varphi = (P \lor Q) \land R$

▶ If P and Q are false, then φ is false
(independent of the truth value of R).

▶ If P and R are true, then φ is true
(independent of the truth value of Q).

Semantics: Formally

▶ defined over interpretation $I : \Sigma \to \{\mathbf{T}, \mathbf{F}\}$

▶ interpretation I: assignment of propositions in Σ

▶ When is a formula φ true under interpretation I?
symbolically: When does $I \models \varphi$ hold?

German: Interpretation, Belegung

Semantics: Formally

Definition ($I \models \varphi$)
▶ $I \models \top$ and $I \not\models \bot$
▶ $I \models P$ iff $I(P) = \mathbf{T}$ for $P \in \Sigma$
▶ $I \models \neg\varphi$ iff $I \not\models \varphi$
▶ $I \models (\varphi \land \psi)$ iff $I \models \varphi$ and $I \models \psi$
▶ $I \models (\varphi \lor \psi)$ iff $I \models \varphi$ or $I \models \psi$
▶ $I \models (\varphi \to \psi)$ iff $I \not\models \varphi$ or $I \models \psi$
▶ $I \models \Phi$ for a set of formulas Φ iff $I \models \varphi$ for all $\varphi \in \Phi$

German: I erfüllt φ, φ gilt unter I

Examples

Example (Interpretation I)
$I = \{P \mapsto \mathbf{T}, Q \mapsto \mathbf{T}, R \mapsto \mathbf{F}, S \mapsto \mathbf{F}\}$

Which formulas are true under I?
▶ $\varphi_1 = \neg(P \land Q) \land (R \land \neg S)$. Does $I \models \varphi_1$ hold?
▶ $\varphi_2 = (P \land Q) \land \neg(R \land \neg S)$. Does $I \models \varphi_2$ hold?
▶ $\varphi_3 = (R \to P)$. Does $I \models \varphi_3$ hold?

Terminology

> **Definition (model)**
> An interpretation I is called a model of φ if $I \models \varphi$.

German: Modell

> **Definition (satisfiable etc.)**
> A formula φ is called
> - satisfiable if there is an interpretation I such that $I \models \varphi$.
> - unsatisfiable if φ is not satisfiable.
> - falsifiable if there is an interpretation I such that $I \not\models \varphi$.
> - valid (= a tautology) if $I \models \varphi$ for all interpretations I.

German: erfüllbar, unerfüllbar, falsifizierbar,
allgemeingültig (gültig, Tautologie)

Terminology

> **Definition (logical equivalence)**
> Formulas φ and ψ are called logically equivalent $(\varphi \equiv \psi)$
> if for all interpretations I: $I \models \varphi$ iff $I \models \psi$.

German: logisch äquivalent

Truth Tables

> **Truth Tables**
> How to determine automatically if a given formula is
> (un)satisfiable, falsifiable, valid?
> ⤳ simple method: truth tables

example: Is $\varphi = ((P \vee H) \wedge \neg H) \to P$ valid?

P	H	$P \vee H$	$((P \vee H) \wedge \neg H)$	$((P \vee H) \wedge \neg H) \to P$
F	F	F	F	T
F	T	T	F	T
T	F	T	T	T
T	T	T	F	T

$I \models \varphi$ for all interpretations I ⤳ φ is valid.
- satisfiability, falsifiability, unsatisfiability?

29.4 Normal Forms

Normal Forms: Terminology

> **Definition (literal)**
> If $P \in \Sigma$, then the formulas P and $\neg P$ are called literals.
>
> P is called positive literal, $\neg P$ is called negative literal.
>
> The complementary literal to P is $\neg P$ and vice versa.
> For a literal ℓ, the complementary literal to ℓ is denoted with $\bar{\ell}$.

German: Literal, positives/negatives/komplementäres Literal

Normal Forms: Terminology

> **Definition (clause)**
> A disjunction of 0 or more literals is called a clause.
> The empty clause $\perp$ is also written as $\square$.
> Clauses consisting of only one literal are called unit clauses.

German: Klausel

> **Definition (monomial)**
> A conjunction of 0 or more literals is called a monomial.

German: Monom

Normal Forms

> **Definition (normal forms)**
> A formula φ is in conjunctive normal form (CNF, clause form)
> if φ is a conjunction of 0 or more clauses:
>
> $$\varphi = \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} \ell_{i,j} \right)$$
>
> A formula φ is in disjunctive normal form (DNF)
> if φ is a disjunction of 0 or more monomials:
>
> $$\varphi = \bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} \ell_{i,j} \right)$$

German: konjunktive Normalform, disjunktive Normalform

Normal Forms

For every propositional formula, there exists
a logically equivalent propositional formula in CNF and in DNF.

> **Conversion to CNF**
> important rules for conversion to CNF:
> - $(\varphi \rightarrow \psi) \equiv (\neg\varphi \vee \psi)$　　　　　　　　$((\rightarrow)\text{-elimination})$
> - $\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$　　　　　　　　(De Morgan)
> - $\neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)$　　　　　　　　(De Morgan)
> - $\neg\neg\varphi \equiv \varphi$　　　　　　　　(double negation)
> - $((\varphi \wedge \psi) \vee \eta) \equiv ((\varphi \vee \eta) \wedge (\psi \vee \eta))$　　　(distributivity)

There are formulas φ for which every logically equivalent formula
in CNF and DNF is exponentially longer than φ.

29.5 Summary

Summary (1)

- **Propositional logic** forms the basis for a general representation of problems and knowledge.
- **Propositions** (atomic formulas) are statements over the world which cannot be divided further.
- **Propositional formulas** combine atomic formulas with ¬, ∧, ∨, → to more complex statements.
- **Interpretations** determine which atomic formulas are true and which ones are false.

Summary (2)

- important terminology:
  - **model**
  - **satisfiable**, **unsatisfiable**, **falsifiable**, **valid**
  - **logically equivalent**
- different kinds of formulas:
  - **atomic formulas** and **literals**
  - **clauses** and **monomials**
  - **conjunctive normal form** and **disjunctive normal form**