Foundations of Artificial Intelligence

E3. Propositional Logic: Reasoning and Resolution

Malte Helmert

University of Basel

April 23, 2025

Propositional Logic: Overview

Chapter overview: propositional logic

- E1. Syntax and Semantics
- E2. Equivalence and Normal Forms
- E3. Reasoning and Resolution
- E4. DPLL Algorithm
- E5. Local Search and Outlook

E3.1 Reasoning

Reasoning: Intuition

Reasoning: Intuition

- Generally, formulas only represent an incomplete description of the world.
- In many cases, we want to know if a formula logically follows from (a set of) other formulas.
- What does this mean?

Reasoning: Intuition

- example: $\varphi = (P \lor Q) \land (R \lor \neg P) \land S$
- S holds in every model of φ.
  What about P, Q and R?

$\rightsquigarrow$ consider all models of φ:

  - $I_1 = \{P \mapsto \mathbf{F}, Q \mapsto \mathbf{T}, R \mapsto \mathbf{F}, S \mapsto \mathbf{T}\}$
  - $I_2 = \{P \mapsto \mathbf{F}, Q \mapsto \mathbf{T}, R \mapsto \mathbf{T}, S \mapsto \mathbf{T}\}$
  - $I_3 = \{P \mapsto \mathbf{T}, Q \mapsto \mathbf{F}, R \mapsto \mathbf{T}, S \mapsto \mathbf{T}\}$
  - $I_4 = \{P \mapsto \mathbf{T}, Q \mapsto \mathbf{T}, R \mapsto \mathbf{T}, S \mapsto \mathbf{T}\}$

Observation

- In all models of φ, the formula $Q \lor R$ holds as well.
- We say: "$Q \lor R$ logically follows from φ."

Reasoning: Formally

Definition (logical consequence)

Let Φ be a set of formulas. A formula ψ **logically follows** from Φ (in symbols: $\Phi \models \psi$) if all models of Φ are also models of ψ.

German: logische Konsequenz, folgt logisch

In other words: for each interpretation I,
if $I \models \varphi$ for all $\varphi \in \Phi$, then also $I \models \psi$.

Question

How can we automatically compute if $\Phi \models \psi$?

- One possibility: Build a truth table. (How?)
- Are there "better" possibilities that (potentially) avoid generating the whole truth table?

Reasoning: Deduction Theorem

Proposition (deduction theorem)

Let Φ be a finite set of formulas and let ψ be a formula. Then

$$\Phi \models \psi \quad \textit{iff} \quad \Big(\bigwedge_{\varphi \in \Phi} \varphi\Big) \rightarrow \psi \textit{ is a tautology.}$$

German: Deduktionssatz

Proof.

$\Phi \models \psi$

iff for each interpretation I: if $I \models \varphi$ for all $\varphi \in \Phi$, then $I \models \psi$

iff for each interpretation I: if $I \models \bigwedge_{\varphi \in \Phi} \varphi$, then $I \models \psi$

iff for each interpretation I: $I \not\models \bigwedge_{\varphi \in \Phi} \varphi$ or $I \models \psi$

iff for each interpretation I: $I \models (\bigwedge_{\varphi \in \Phi} \varphi) \rightarrow \psi$

iff $(\bigwedge_{\varphi \in \Phi} \varphi) \rightarrow \psi$ is tautology $\square$

Reasoning by Unsatisfiability Testing

Consequence of Deduction Theorem

Reasoning can be reduced to testing unsatisfiability.

Question: Does $\Phi \models \psi$ hold?

Idea:

- Let $\chi = (\bigwedge_{\varphi \in \Phi} \varphi) \rightarrow \psi$.
- We know that $\Phi \models \psi$ iff χ is a tautology.
- A formula is a tautology iff its negation is unsatisfiable.
- Hence, $\Phi \models \psi$ iff $\neg\chi$ is unsatisfiable.
- Use equivalences:
  $$\begin{aligned}\neg\chi = \neg((\textstyle\bigwedge_{\varphi \in \Phi} \varphi) \rightarrow \psi) &\equiv \neg(\neg(\textstyle\bigwedge_{\varphi \in \Phi} \varphi) \vee \psi) \\ &\equiv (\neg\neg(\textstyle\bigwedge_{\varphi \in \Phi} \varphi) \wedge \neg\psi) \equiv \textstyle\bigwedge_{\varphi \in \Phi} \varphi \wedge \neg\psi\end{aligned}$$
- We have that $\Phi \models \psi$ iff $\bigwedge_{\varphi \in \Phi} \varphi \wedge \neg\psi$ is unsatisfiable.

Algorithm for Reasoning

Question: Does $\Phi \models \psi$ hold?

Algorithm (given an algorithm for testing unsatisfiability):

1. Let $\eta = \bigwedge_{\varphi \in \Phi} \varphi \land \neg\psi$.
2. Test if η is unsatisfiable.
3. If yes, return "$\Phi \models \psi$".
4. Otherwise, return "$\Phi \not\models \psi$".

In the following: Can we test unsatisfiability in a more efficient way than by computing the whole truth table?

E3.2 Resolution

Sets of Clauses

for the rest of this chapter:

- prerequisite: formulas in conjunctive normal form
- clause represented as a set C of literals
- formula represented as a set Δ of clauses

Example

Let $\varphi = (P \vee Q) \wedge \neg P$.

- φ in conjunctive normal form
- φ consists of clauses $(P \vee Q)$ and $\neg P$
- representation of φ as set of sets of literals: $\{\{P, Q\}, \{\neg P\}\}$

Sets of Clauses (Corner Cases)

Distinguish $\bot$ (empty clause = empty set of literals) vs. $\emptyset$ (empty set of clauses).

- $C = \bot$ $(= \emptyset)$ represents a disjunction over zero literals:

$$\bigvee_{L \in \emptyset} L = \bot$$

- $\Delta_1 = \{\bot\}$ represents a conjunction over one clause:

$$\bigwedge_{\varphi \in \{\bot\}} \varphi = \bot$$

- $\Delta_2 = \emptyset$ represents a conjunction over zero clauses:

$$\bigwedge_{\varphi \in \emptyset} \varphi = \top$$

Resolution: Idea

Resolution

- method to test CNF formula φ for unsatisfiability
- idea: derive new clauses from φ that logically follow from φ
- if empty clause $\bot$ can be derived $\rightsquigarrow$ φ unsatisfiable

German: Resolution

The Resolution Rule

$$\frac{C_1 \cup \{\ell\}, C_2 \cup \{\bar{\ell}\}}{C_1 \cup C_2}$$

- "From $C_1 \cup \{\ell\}$ and $C_2 \cup \{\bar{\ell}\}$, we can conclude $C_1 \cup C_2$."
- $C_1 \cup C_2$ is resolvent of parent clauses $C_1 \cup \{\ell\}$ and $C_2 \cup \{\bar{\ell}\}$.
- The literals ℓ and $\bar{\ell}$ are called resolution literals, the corresponding proposition is called resolution variable.
- resolvent follows logically from parent clauses (Why?)

German: Resolutionsregel, Resolvent, Elternklauseln, Resolutionsliterale, Resolutionsvariable

Example

- resolvent of $\{A, B, \neg C\}$ and $\{A, D, C\}$?
- resolvents of $\{\neg A, B, \neg C\}$ and $\{A, D, C\}$?

Resolution: Derivations

Definition (derivation)

Notation: $R(\Delta) = \Delta \cup \{C \mid C \text{ is resolvent of two clauses in } \Delta\}$

A clause D can be derived from Δ (in symbols $\Delta \vdash D$) if there is a sequence of clauses $C_1, \dots, C_n = D$ such that for all $i \in \{1, \dots, n\}$ we have $C_i \in R(\Delta \cup \{C_1, \dots, C_{i-1}\})$.

German: Ableitung, abgeleitet

Lemma (soundness of resolution)

If $\Delta \vdash D$, then $\Delta \models D$.

Does the converse direction hold as well (completeness)?

German: Korrektheit, Vollständigkeit

Resolution: Completeness?

The converse of the lemma does not hold in general.

example:

- $\{\{A, B\}, \{\neg B, C\}\} \models \{A, B, C\}$, but
- $\{\{A, B\}, \{\neg B, C\}\} \nvdash \{A, B, C\}$

but: converse holds for special case of empty clause $\bot$ (no proof)

Theorem (refutation-completeness of resolution)

Δ *is unsatisfiable iff* $\Delta \vdash \bot$

German: Widerlegungsvollständigkeit

consequences:

- Resolution is a complete proof method for testing unsatisfiability of CNF formulas.
- Resolution can be used for general reasoning by reducing to a test for unsatisfiability of CNF formulas.

Example

Let $\Phi = \{P \vee Q, \neg P\}$. Does $\Phi \models Q$ hold?

Solution

- test if $((P \vee Q) \wedge \neg P) \rightarrow Q$ is tautology
- equivalently: test if $((P \vee Q) \wedge \neg P) \wedge \neg Q$ is unsatisfiable
- resulting set of clauses: $\Phi' = \{\{P, Q\}, \{\neg P\}, \{\neg Q\}\}$
- resolving $\{P, Q\}$ with $\{\neg P\}$ yields $\{Q\}$
- resolving $\{Q\}$ with $\{\neg Q\}$ yields $\perp$
- observation: empty clause can be derived, hence Φ' unsatisfiable
- consequently $\Phi \models Q$

Resolution: Discussion

- Resolution is a complete proof method to test formulas for unsatisfiability.
- In the worst case, resolution proofs can take exponential time.
- In practice, a strategy which determines the next resolution step is needed.
- In the following chapter, we discuss the DPLL algorithm, which is a combination of backtracking and resolution.

E3.3 Summary

Summary

- Reasoning: the formula ψ follows from the set of formulas Φ if all models of Φ are also models of ψ.
- Reasoning can be reduced to testing validity (with the deduction theorem).
- Testing validity can be reduced to testing unsatisfiability.
- Resolution is a refutation-complete proof method applicable to formulas in conjunctive normal form.

$\rightsquigarrow$ can be used to test if a set of clauses is unsatisfiable