

Foundations of Artificial Intelligence

E3. Propositional Logic: Resolution

Malte Helmert

University of Basel

April 22, 2026

Foundations of Artificial Intelligence

April 22, 2026 — E3. Propositional Logic: Resolution

E3.1 Motivation

E3.2 Resolution

E3.3 Summary

Propositional Logic: Overview

Chapter overview: propositional logic

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

E3.1 Motivation

Reasoning and (Un-) Satisfiability

- ▶ last chapter: reduce reasoning to unsatisfiability
- ▶ also: interest in (un-) satisfiability in its own right

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

- ▶ E3 & E4: complete algorithms for determining satisfiability/unsatisfiability of a formula
- ▶ E5: incomplete algorithms that find satisfying assignments

E3.2 Resolution

Sets of Clauses

in 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 \perp (empty clause = empty set of literals)
vs. \emptyset (empty set of clauses).

- ▶ $C = \perp (= \emptyset)$ represents a **disjunction over zero literals**:

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

- ▶ $\Delta_1 = \{\perp\}$ represents a **conjunction over one clause**:

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

- ▶ $\Delta_2 = \emptyset$ 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 \perp can be derived $\rightsquigarrow \varphi$ unsatisfiable

German: Resolution

The Resolution Rule

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

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

German: Resolutionsregel, Resolvent, Elternklauseln, Resolutionsliterals, 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
- ▶ $\{\{A, B\}, \{\neg B, C\}\} \not\models \{A, B, C\}$

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

Theorem (refutation-completeness of resolution)

Δ is unsatisfiable iff $\Delta \vdash \perp$

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

- ▶ **resolution rule** (for clauses in set notation):
“from $C_1 \cup \{l\}$ and $C_2 \cup \{\bar{l}\}$, conclude $C_1 \cup C_2$.”
 - ▶ **resolution**: repeatedly apply the rule to see if a contradiction can be derived (empty clause \perp)
 - ▶ is a **refutation-complete** proof method applicable to formulas in conjunctive normal form.
- ↪ can be used to test if a set of clauses is unsatisfiable