

# Foundations of Artificial Intelligence

## 30. Propositional Logic: Reasoning and Resolution

Malte Helmert

University of Basel

April 17, 2019

Foundations of Artificial Intelligence  
April 17, 2019 — 30. Propositional Logic: Reasoning and Resolution

### 30.1 Reasoning

### 30.2 Resolution

### 30.3 Summary

## Propositional Logic: Overview

Chapter overview: propositional logic

- ▶ [29. Basics](#)
- ▶ [30. Reasoning and Resolution](#)
- ▶ [31. DPLL Algorithm](#)
- ▶ [32. Local Search and Outlook](#)

### 30.1 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:**  $\varphi = (P \vee Q) \wedge (R \vee \neg P) \wedge S$
- ▶  $S$  holds in every model of  $\varphi$ . What about  $P$ ,  $Q$  and  $R$ ?
- ↪ consider all models of  $\varphi$ :

$P$	$Q$	$R$	$S$
F	T	F	T
F	T	T	T
T	F	T	T
T	T	T	T

### Observation

- ▶ In all models of  $\varphi$ , the formula  $Q \vee R$  holds as well.
- ▶ We say: " $Q \vee R$  **logically follows** from  $\varphi$ ."

## Reasoning: Formally

### Definition (logical consequence)

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

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  $\Phi$  be a finite set of formulas and let  $\psi$  be a formula. Then

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

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

□

## Reasoning

### Consequence of Deduction Theorem

Reasoning can be reduced to testing validity.

### Algorithm

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

- ① test if  $(\bigwedge_{\varphi \in \Phi} \varphi) \rightarrow \psi$  is tautology
- ② if yes, then  $\Phi \models \psi$ , otherwise  $\Phi \not\models \psi$

In the following: Can we test for validity “efficiently”, i.e., without computing the whole truth table?

## 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  $\Delta$  of clauses**

### Example

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

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

Distinguish  $\square$  (empty clause) vs.  $\emptyset$  (empty set of clauses).

## 30.2 Resolution

### Resolution: Idea

#### Observation

- ▶ Testing for validity can be reduced to testing unsatisfiability.
- ▶ formula  $\varphi$  valid iff  $\neg\varphi$  unsatisfiable

#### Resolution: Idea

- ▶ method to test formula  $\varphi$  for unsatisfiability
- ▶ **idea:** derive new formulas from  $\varphi$  that logically follow from  $\varphi$
- ▶ if empty clause  $\square$  can be derived  $\leadsto \varphi$  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  $\ell$  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: 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  $\square$

### Proposition (refutation-completeness of resolution)

$\Delta$  is unsatisfiable iff  $\Delta \vdash \square$

**German:** Widerlegungsvollständigkeit

**consequences:**

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

## 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  $\Delta$  (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  $\square$

### Proposition (refutation-completeness of resolution)

$\Delta$  is unsatisfiable iff  $\Delta \vdash \square$

**German:** Widerlegungsvollständigkeit

**consequences:**

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

## 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  $\square$
- ▶ observation: empty clause can be derived, hence  $\Phi'$  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.

## 30.3 Summary

## Summary

- ▶ **Reasoning:** the formula  $\psi$  follows from the set of formulas  $\Phi$  if all models of  $\Phi$  are also models of  $\psi$ .
- ▶ 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