

Discrete Mathematics in Computer Science

E4. Inference

Malte Helmert, Gabriele Röger

University of Basel

Discrete Mathematics in Computer Science

— E4. Inference

E4.1 Inference Rules and Calculi

E4.2 Resolution Calculus

E4.1 Inference Rules and Calculi

Inference: Motivation

- ▶ **up to now:** proof of **logical consequence** with **semantic arguments**
- ▶ no general algorithm
- ▶ **solution:** produce formulas that are logical consequences of given formulas with **syntactic inference rules**
- ▶ **advantage:** **mechanical method** that can easily be implemented as an algorithm

Inference Rules

- ▶ **Inference rules** have the form

$$\frac{\varphi_1, \dots, \varphi_k}{\psi}$$

- ▶ Meaning: “Every model of $\varphi_1, \dots, \varphi_k$ is a model of ψ .”
- ▶ An **axiom** is an inference rule with $k = 0$.
- ▶ A set of inference rules is called a **calculus** or **proof system**.

German: Inferenzregel, Axiom, (der) Kalkül, Beweissystem

Some Inference Rules for Propositional Logic

$$\text{Modus ponens} \quad \frac{\varphi, (\varphi \rightarrow \psi)}{\psi}$$

$$\text{Modus tollens} \quad \frac{\neg\psi, (\varphi \rightarrow \psi)}{\neg\varphi}$$

$$\wedge\text{-elimination} \quad \frac{(\varphi \wedge \psi)}{\varphi} \quad \frac{(\varphi \wedge \psi)}{\psi}$$

$$\wedge\text{-introduction} \quad \frac{\varphi, \psi}{(\varphi \wedge \psi)}$$

$$\vee\text{-introduction} \quad \frac{\varphi}{(\varphi \vee \psi)}$$

$$\leftrightarrow\text{-elimination} \quad \frac{(\varphi \leftrightarrow \psi)}{(\varphi \rightarrow \psi)} \quad \frac{(\varphi \leftrightarrow \psi)}{(\psi \rightarrow \varphi)}$$

Derivation

Definition (Derivation)

A **derivation** or **proof** of a formula φ from a knowledge base KB is a sequence of formulas ψ_1, \dots, ψ_k with

- ▶ $\psi_k = \varphi$ and
- ▶ for all $i \in \{1, \dots, k\}$:
 - ▶ $\psi_i \in \text{KB}$, or
 - ▶ ψ_i is the result of the application of an inference rule to elements from $\{\psi_1, \dots, \psi_{i-1}\}$.

German: Ableitung, Beweis

Derivation: Example

Example

Given: KB = $\{P, (P \rightarrow Q), (P \rightarrow R), ((Q \wedge R) \rightarrow S)\}$

Task: Find derivation of $(S \wedge R)$ from KB.

- 1 P (KB)
- 2 $(P \rightarrow Q)$ (KB)
- 3 Q (1, 2, Modus ponens)
- 4 $(P \rightarrow R)$ (KB)
- 5 R (1, 4, Modus ponens)
- 6 $(Q \wedge R)$ (3, 5, \wedge -introduction)
- 7 $((Q \wedge R) \rightarrow S)$ (KB)
- 8 S (6, 7, Modus ponens)
- 9 $(S \wedge R)$ (8, 5, \wedge -introduction)

Correctness and Completeness

Definition (Correctness and Completeness of a Calculus)

We write $KB \vdash_C \varphi$ if there is a derivation of φ from KB in calculus C .

(If calculus C is clear from context, also only $KB \vdash \varphi$.)

A calculus C is **correct** if for all KB and φ
 $KB \vdash_C \varphi$ implies $KB \models \varphi$.

A calculus C is **complete** if for all KB and φ
 $KB \models \varphi$ implies $KB \vdash_C \varphi$.

Consider calculus C , consisting of the derivation rules seen earlier.

Question: Is C correct?

Question: Is C complete?

German: korrekt, vollständig

Refutation-completeness

- ▶ We obviously want **correct** calculi.
- ▶ Do we always need a **complete** calculus?
- ▶ **Contradiction theorem**:
 $KB \cup \{\varphi\}$ is unsatisfiable iff $KB \models \neg\varphi$
- ▶ This implies that $KB \models \varphi$ iff $KB \cup \{\neg\varphi\}$ is unsatisfiable.
- ▶ We can reduce the **general** implication problem to a **test of unsatisfiability**.
- ▶ In calculi, we use the special symbol \square for (provably) unsatisfiable formulas.

Definition (Refutation-Completeness)

A calculus C is **refutation-complete** if $KB \vdash_C \square$ for all unsatisfiable KB.

German: widerlegungsvollständig

E4.2 Resolution Calculus

Resolution: Idea

- ▶ **Resolution** is a refutation-complete calculus for knowledge bases in **conjunctive normal form**.
- ▶ Every knowledge base can be transformed into equivalent formulas in CNF.
 - ▶ Transformation can require exponential time.
 - ▶ Alternative: efficient transformation into **equisatisfiable** formulas (not part of this course)
- ▶ Show $KB \models \varphi$ by deriving $KB \cup \{\neg\varphi\} \vdash_R \square$ with **resolution calculus R** .
- ▶ Resolution can require exponential time.
- ▶ This is probably the case for **all** refutation-complete proof methods. \rightsquigarrow **complexity theory**

German: Resolution, erfüllbarkeitsäquivalent

Knowledge Base as Set of Clauses

Simplified notation of knowledge bases in CNF

- ▶ **Formula** in CNF as **set of clauses**
(due to commutativity, idempotence, associativity of \wedge)
- ▶ **Set of formulas** as **set of clauses**
- ▶ **Clause** as **set of literals**
(due to commutativity, idempotence, associativity of \vee)
- ▶ Knowledge base as **set of sets of literals**

Example

KB = $\{(P \vee P), ((\neg P \vee Q) \wedge (\neg P \vee R) \wedge (Q \vee \neg P) \wedge R),$
 $((\neg Q \vee \neg R \vee S) \wedge P)\}$

as set of clauses:

$\Delta = \{\{P\}, \{\neg P, Q\}, \{\neg P, R\}, \{R\}, \{\neg Q, \neg R, S\}\}$

Resolution Rule

The **resolution calculus** consists of a single rule, called **resolution rule**:

$$\frac{C_1 \cup \{X\}, C_2 \cup \{\neg X\}}{C_1 \cup C_2},$$

where C_1 and C_2 are (possibly empty) clauses and X is an atomic proposition.

If we derive the empty clause, we write \square instead of $\{\}$.

Terminology:

- ▶ X and $\neg X$ are the **resolution literals**,
- ▶ $C_1 \cup \{X\}$ and $C_2 \cup \{\neg X\}$ are the **parent clauses**, and
- ▶ $C_1 \cup C_2$ is the **resolvent**.

German: Resolutionskalkül, Resolutionsregel, Resolutionsliterals, Elternklauseln, Resolvent

Proof by Resolution

Definition (Proof by Resolution)

A **proof by resolution** of a clause D from a knowledge base Δ is a sequence of clauses C_1, \dots, C_n with

- ▶ $C_n = D$ and
- ▶ for all $i \in \{1, \dots, n\}$:
 - ▶ $C_i \in \Delta$, or
 - ▶ C_i is resolvent of two clauses from $\{C_1, \dots, C_{i-1}\}$.

If there is a proof of D by resolution from Δ , we say that D can be **derived with resolution from Δ** and write $\Delta \vdash_R D$.

Remark: Resolution is a **correct, refutation-complete**, but **incomplete** calculus.

German: Resolutionsbeweis, mit Resolution aus Δ abgeleitet

Proof by Resolution: Example

Proof by Resolution for Testing a Logical Consequence: Example

Given: KB = $\{P, (P \rightarrow (Q \wedge R))\}$.

Show with resolution that $\text{KB} \models (R \vee S)$.

Three steps:

- 1 Reduce logical consequence to unsatisfiability.
- 2 Transform knowledge base into clause form (CNF).
- 3 Derive empty clause \square with resolution.

Step 1: Reduce logical consequence to unsatisfiability.

$\text{KB} \models (R \vee S)$ iff $\text{KB} \cup \{\neg(R \vee S)\}$ is unsatisfiable.

Thus, consider

$\text{KB}' = \text{KB} \cup \{\neg(R \vee S)\} = \{P, (P \rightarrow (Q \wedge R)), \neg(R \vee S)\}$.

...

Proof by Resolution: Example (continued)

Proof by Resolution for Testing a Logical Consequence: Example

$KB' = \{P, (P \rightarrow (Q \wedge R)), \neg(R \vee S)\}$.

Step 2: Transform knowledge base into clause form (CNF).

- ▶ P
 \rightsquigarrow Clauses: $\{P\}$
- ▶ $P \rightarrow (Q \wedge R) \equiv (\neg P \vee (Q \wedge R)) \equiv ((\neg P \vee Q) \wedge (\neg P \vee R))$
 \rightsquigarrow Clauses: $\{\neg P, Q\}, \{\neg P, R\}$
- ▶ $\neg(R \vee S) \equiv (\neg R \wedge \neg S)$
 \rightsquigarrow Clauses: $\{\neg R\}, \{\neg S\}$

$\Delta = \{\{P\}, \{\neg P, Q\}, \{\neg P, R\}, \{\neg R\}, \{\neg S\}\}$

...

Proof by Resolution: Example (continued)

Proof by Resolution for Testing a Logical Consequence: Example

$\Delta = \{\{P\}, \{\neg P, Q\}, \{\neg P, R\}, \{\neg R\}, \{\neg S\}\}$

Step 3: Derive empty clause \square with resolution.

- ▶ $C_1 = \{P\}$ (from Δ)
- ▶ $C_2 = \{\neg P, Q\}$ (from Δ)
- ▶ $C_3 = \{\neg P, R\}$ (from Δ)
- ▶ $C_4 = \{\neg R\}$ (from Δ)
- ▶ $C_5 = \{Q\}$ (from C_1 and C_2)
- ▶ $C_6 = \{\neg P\}$ (from C_3 and C_4)
- ▶ $C_7 = \square$ (from C_1 and C_6)

Note: There are shorter proofs. (For example?)

Another Example

Another Example for Resolution

Show with resolution, that $KB \models \text{DrinkBeer}$, where

$$KB = \{(\neg \text{DrinkBeer} \rightarrow \text{EatFish}), \\ ((\text{EatFish} \wedge \text{DrinkBeer}) \rightarrow \neg \text{EatIceCream}), \\ ((\text{EatIceCream} \vee \neg \text{DrinkBeer}) \rightarrow \neg \text{EatFish})\}.$$

Proving that Something Does Not Follow

- ▶ We can now use resolution proofs to mechanically show $KB \models \varphi$ whenever a given knowledge base logically implies φ .
- ▶ Question: How can we use the same mechanism to show that something does **not** follow ($KB \not\models \varphi$)?