

Proof Methods in FOL

Major Families:

- GMP
- Reduction
- Resolution
- Forward chaining
- Backward chaining

Some Other inference tools:

Entailment/ Unification/

Existential instantiation (EI)

- For any sentence α , variable v and constant symbol k that does **not** appear elsewhere in the knowledge base:

$$\frac{\exists v \alpha}{\text{Subst}(\{v/k\}, \alpha)}$$

- E.g., $\exists x \text{Crow}(x) \wedge \text{OnHead}(x, \text{John})$ yields:

$$\text{Crow}(C_j) \wedge \text{OnHead}(C_j, \text{John})$$

provided C_j is a new constant symbol, called a **Skolem constant**

Universal instantiation (UI)

- Every instantiation of a universally quantified sentence is entailed by it:

$$\frac{\forall v \alpha}{\text{Subst}(\{v/g\}, \alpha)}$$

for any variable v and ground term g

- E.g., $\forall x \text{King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$ yields:
 $\text{King}(\text{John}) \wedge \text{Greedy}(\text{John}) \Rightarrow \text{Evil}(\text{John})$
 $\text{King}(\text{Richard}) \wedge \text{Greedy}(\text{Richard}) \Rightarrow \text{Evil}(\text{Richard})$
 $\text{King}(\text{Father}(\text{John})) \wedge \text{Greedy}(\text{Father}(\text{John})) \Rightarrow \text{Evil}(\text{Father}(\text{John}))$

Resolution in PC

- **Propositional version.**

$$\{a \vee b, \neg b \vee g\} \mid - a \vee g \quad \text{OR} \quad \{\neg a \Rightarrow b, b \Rightarrow g\} \mid - \neg a \Rightarrow g$$

- Reasoning by cases **OR** transitivity of implication

- **First-order form**

- For two literals p_k and q_l in two clauses

$$p_1 \vee p_2 \vee \dots \vee p_n$$

$$q_1 \vee q_2 \vee \dots \vee q_n$$

such that $\theta = \text{UNIFY}(p_k, \neg q_l)$, derives

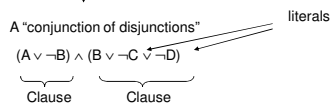
$$\text{SUBST}(\theta, p_1 \vee p_2 \vee \dots \vee p_{k-1} \vee p_{k+1} \vee \dots \vee p_n \vee q_1 \vee q_2 \vee \dots \vee q_{l-1} \vee q_{l+1} \vee \dots \vee q_n)$$

- For resolution to apply, all sentences must be in **conjunctive normal form**,

Conjunctive Normal Form (CNF) in PC

We like to prove: $KB \models \alpha$
 equivalent to: $KB \wedge \neg\alpha$ unsatisfiable

We first rewrite $KB \wedge \neg\alpha$ into conjunctive normal form (CNF).



In theory

- Any KB can be converted into CNF.
- In fact, any KB can be converted into CNF-3, i.e. using clauses with at most 3 literals.

Example: Conversion to CNF in PC

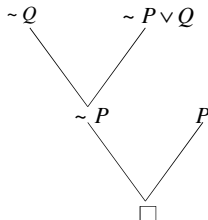
$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

1. Eliminate \Leftrightarrow , replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
 $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
2. Eliminate \Rightarrow , replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
 $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$
3. Move \neg inwards using de Morgan's rules and double-negation: $\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$
 $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$
4. Apply distributive law (\wedge over \vee) and flatten:
 $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Resolution in PC

1. P
2. $P \rightarrow Q$ converted to $\sim P \vee Q$
3. $\sim Q$

Draw the resolution tree (actually an inverted tree). Every node is a clausal form and branches are intermediate inference steps.



Resolution Algorithm in PC

- The resolution algorithm tries to prove: $KB \models \alpha$ equivalent to $KB \wedge \neg\alpha$ unsatisfiable
- Generate all new sentences from KB and the query.
- One of two things can happen:
 1. We find $P \wedge \neg P$ which is unsatisfiable. i.e. we can entail the query.
 2. We find no contradiction: there is a model that satisfies the sentence $KB \wedge \neg\alpha$ (non-trivial) and hence we cannot entail the query.

Resolution as Search

Given a database in clausal normal form KB

Find a sequence of resolution steps from KB to the empty clauses

States: current cnf KB + new clauses

Operators: resolution

Initial state: KB + negated goal

Goal State: a database containing the empty clause

Resolution Algorithm in FOPC

- 1) Convert sentences in the KB to CNF (clausal form)
- 2) Take the negation of the proposed query, convert it to CNF, and add it to the KB.
- 3) Repeatedly apply the resolution rule to derive new clauses.
- 4) If the empty clause (False) is eventually derived, stop and conclude that the proposed theorem is true.

Procedure:

- ✓ Eliminate implications and biconditionals
- ✓ Move \neg inward
- ✓ Standardize variables
- ✓ Move quantifiers left
- ✓ Skolemize: replace each existentially quantified variable with a Skolem constant or Skolem function
- ✓ Distribute \wedge over \vee to convert to conjunctions of clauses
- ✓ Convert clauses to implications if desired for readability
($\neg a \vee \neg b \vee c \vee d$) To $a \vee b \Rightarrow c \vee d$

Conversion to CNF

- Everyone who loves all animals is loved by someone:

$$\forall x ([\forall y \text{Animal}(y) \Rightarrow \text{Loves}(x,y)] \Rightarrow [\exists y \text{Loves}(y,x)])$$

1. Eliminate biconditionals and implications

$$\forall x ([\neg \forall y (\neg \text{Animal}(y) \vee \text{Loves}(x,y))] \vee [\exists y \text{Loves}(y,x)])$$

2. Move \neg inwards: " $\neg \forall x p \equiv \exists x \neg p$, $\neg \exists x p \equiv \forall x \neg p$ "

$$\forall x ([\exists y (\neg (\neg \text{Animal}(y) \vee \text{Loves}(x,y)))] \vee [\exists y \text{Loves}(y,x)])$$

$$\forall x ([\exists y (\neg \neg \text{Animal}(y) \wedge \neg \text{Loves}(x,y))] \vee [\exists y \text{Loves}(y,x)])$$

$$\forall x ([\exists y (\text{Animal}(y) \wedge \neg \text{Loves}(x,y))] \vee [\exists y \text{Loves}(y,x)])$$

Conversion to CNF contd.

3. Standardize variables: each quantifier should use a different one

$$\forall x ([\exists y \text{Animal}(y) \wedge \neg \text{Loves}(x,y)] \vee [\exists z \text{Loves}(z,x)])$$

4. Skolemize: a more general form of existential instantiation. Each existential variable is replaced by a Skolem function of the enclosing universally quantified variables:

$$\forall x ([\text{Animal}(F(x)) \wedge \neg \text{Loves}(x,F(x))] \vee \text{Loves}(G(x),x))$$

5. Drop universal quantifiers:

$$[\text{Animal}(F(x)) \wedge \neg \text{Loves}(x,F(x))] \vee \text{Loves}(G(x),x)$$

6. Distribute \vee over \wedge :

$$[\text{Animal}(F(x)) \vee \text{Loves}(G(x),x)] \wedge [\neg \text{Loves}(x,F(x)) \vee \text{Loves}(G(x),x)]$$

Recall: Resolution in PC

- Resolution inference rule (for CNF):

$$\frac{\ell_1 \vee \dots \vee \ell_k, \quad m_1 \vee \dots \vee m_n}{\ell_1 \vee \dots \vee \ell_{s-1} \vee \ell_{s+1} \vee \dots \vee \ell_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n}$$

where ℓ_s and m_j are complementary literals.

E.g., $\frac{P_{1,3} \vee P_{2,2}, \quad \neg P_{2,2}}{P_{1,3}}$

Resolution is sound and complete for propositional logic

Resolution in FOL

- Full first-order version:

$$\frac{\ell_1 \vee \dots \vee \ell_k, \quad m_1 \vee \dots \vee m_n}{(\ell_1 \vee \dots \vee \ell_{k-1} \vee \ell_{k+1} \vee \dots \vee \ell_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n)\theta}$$

where $\text{Unify}(\ell_i, \neg m_j) = \theta$.

The two clauses are assumed to be standardized apart so that they share no variables.

- For example,

$$\frac{\neg \text{Rich}(x) \vee \text{Unhappy}(x)}{\text{Rich}(Ken)}{\text{Unhappy}(Ken)}$$

with $\theta = \{x/Ken\}$

Empty Clause means False

- Resolution theorem proving ends
 - When the resolved clause has no literals (empty)
- This can only be because:
 - Two **unit clauses** were resolved
 - One was the negation of the other (after substitution)
 - Example: $q(X)$ and $\neg q(X)$ or: $p(X)$ and $\neg p(\text{bob})$
- Hence if we see the empty clause
 - This was because there was an inconsistency
 - Hence the proof by refutation

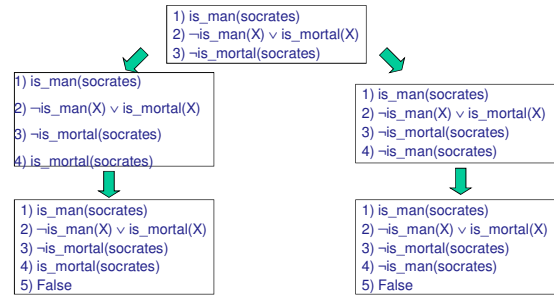
Resolution as Search

- Initial State:** Knowledge base (KB) of axioms and negated theorem in CNF
- Operators:** Resolution rule picks 2 clauses and adds new clause
- Goal Test:** Does KB contain the empty clause?
- Search space of KB states

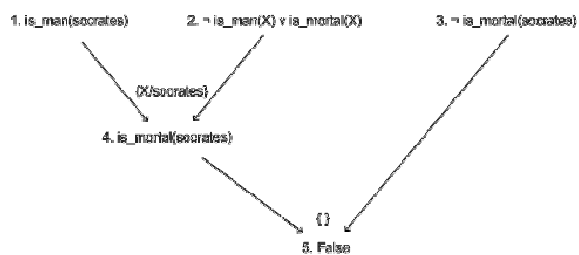
Socrates' Example

- KB: *Socrates is a man and all men are mortal*
Therefore Socrates is mortal
- Initial state
 - 1) `is_man(socrates)`
 - 2) $\neg is_man(X) \vee is_mortal(X)$
 - 3) $\neg is_mortal(socrates)$ (negation of theorem)
- Resolving (1) & (2) gives new state
 - 4) `is_mortal(socrates)`
- Resolving (3) & (4) gives new state empty

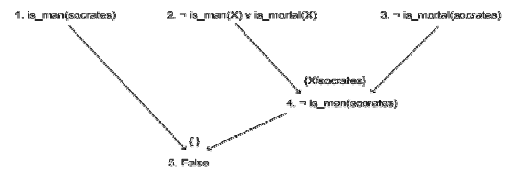
Aristotle's Example: Search Space



Resolution Proof Tree (Proof 1)



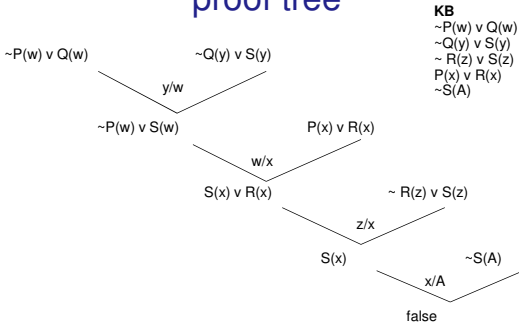
Resolution Proof Tree (Proof 2)



Read as:

You said that all men were mortal. That means that for all things X, either X is not a man, or X is mortal. If we assume that Socrates is not mortal, then, given your previous statement, this means Socrates is not a man. But you said that Socrates *is* a man, which means that our assumption was false, so Socrates must be mortal.

Building Refutation resolution proof tree



Conversion to CNF

- Everyone who loves all animals is loved by someone:

$$\forall x ([\forall y \text{Animal}(y) \Rightarrow \text{Loves}(x,y)] \Rightarrow [\exists y \text{Loves}(y,x)])$$

1. Eliminate biconditionals and implications

$$\forall x ([\neg \forall y (\neg \text{Animal}(y) \vee \text{Loves}(x,y))] \vee [\exists y \text{Loves}(y,x)])$$

2. Move \neg inwards: " $\neg \forall x p \equiv \exists x \neg p$, $\neg \exists x p \equiv \forall x \neg p$ "

$$\forall x ([\exists y (\neg(\neg \text{Animal}(y) \vee \text{Loves}(x,y)))] \vee [\exists y \text{Loves}(y,x)])$$

$$\forall x ([\exists y (\neg \neg \text{Animal}(y) \wedge \neg \text{Loves}(x,y))] \vee [\exists y \text{Loves}(y,x)])$$

$$\forall x ([\exists y (\text{Animal}(y) \wedge \neg \text{Loves}(x,y))] \vee [\exists y \text{Loves}(y,x)])$$

Conversion to CNF contd.

- Standardize variables: each quantifier should use a different one

$$\forall x ([\exists y \text{Animal}(y) \wedge \neg \text{Loves}(x,y)] \vee [\exists z \text{Loves}(z,x)])$$

- Skolemize: a more general form of existential instantiation. Each existential variable is replaced by a Skolem function of the enclosing universally quantified variables:

$$\forall x ([\text{Animal}(F(x)) \wedge \neg \text{Loves}(x,F(x))] \vee \text{Loves}(G(x),x))$$

- Drop universal quantifiers:

$$[\text{Animal}(F(x)) \wedge \neg \text{Loves}(x,F(x))] \vee \text{Loves}(G(x),x)$$

- Distribute \vee over \wedge :

$$[\text{Animal}(F(x)) \vee \text{Loves}(G(x),x)] \wedge [\neg \text{Loves}(x,F(x)) \vee \text{Loves}(G(x),x)]$$

Example: KB

Jack owns a dog.

Every dog owner is an animal lover.

No animal lover kills an animal.

Either Jack or Curiosity killed the cat, who is named Tuna.

Did Curiosity kill the cat?

Example: KB

Jack owns a dog.
 Every dog owner is an animal lover.
 No animal lover kills an animal.
 Either Jack or Curiosity killed the cat, who is named Tuna.
 Did Curiosity kill the cat?

- A. $\exists x \text{ Dog}(x) \wedge \text{Owns}(\text{Jack}, x)$
- B. $\forall x (\exists y \text{ Dog}(y) \wedge \text{Owns}(x, y)) \Rightarrow \text{AnimalLover}(x)$
- C. $\forall x \text{ AnimalLover}(x) \Rightarrow \forall y \text{ Animal}(y) \Rightarrow \neg \text{Kills}(x, y)$
- D. $\text{Kills}(\text{Jack}, \text{Tuna}) \vee \text{Kills}(\text{Curiosity}, \text{Tuna})$
- E. $\text{Cat}(\text{Tuna})$
- F. $\forall x \text{ Cat}(x) \Rightarrow \text{Animal}(x)$

Example: (CNF)

- A1. $\text{Dog}(D)$
- A2. $\text{Owns}(\text{Jack}, D)$
- B. $\text{Dog}(y) \wedge \text{Owns}(x, y) \Rightarrow \text{AnimalLover}(x)$
- C. $\text{AnimalLover}(x) \wedge \text{Animal}(y) \wedge \text{Kills}(x, y) \Rightarrow \text{False}$
- D. $\text{Kills}(\text{Jack}, \text{Tuna}) \vee \text{Kills}(\text{Curiosity}, \text{Tuna})$
- E. $\text{Cat}(\text{Tuna})$
- F. $\text{Cat}(x) \Rightarrow \text{Animal}(x)$

Example: Proof Tree

