Introduction to Logic
Refutation Proofs

Michael Genesereth
Computer Science Department
Stanford University

Proof Systems

Popular Types of Proof Systems:

- Direct Proofs (Hilbert)
- Natural Deduction (Fitch)
- → Refutation proofs (Resolution / Robinson)

Others:

- Gentzen Systems
- Sequent Calculi
- and so forth

Direct Proof

A *direct proof* is a sequence of sentences terminating in a conclusion in which each item is either a premise, an instance of a valid schema, or the result of applying a rule of inference to earlier items in sequence.

1.	*premise*	Premise
2.	*premise*	Premise
	...	...
n.	*conclusion*	Some rule of inference

The conclusion is proved *directly*.

Refutation Proof

A *refutation proof* is a sequence of sentences in which each sentence is a premise, the *negation* of a desired conclusion, or the result of applying a rule of inference to earlier items in sequence that terminates in some form of contradiction.

1.	*premise*	Premise
2.	*premise*	Premise
3.	$\neg$*conclusion*	Negated conclusion
	...	...
n.	*contradiction*	Some rule of inference

A refutation proof is a proof *by contradiction*.

Propositional Resolution

Propositional Resolution is a refutation proof system.

Just one rule of inference - the *Resolution Principle*.

Propositional Resolution is sound and complete.

The search space in propositional resolution is smaller than that of direct proof systems or natural deduction systems.

Hitch: To order to use resolution, we need to transform sentences into a representation called clausal form.

Programme

Clausal Form

Resolution Rule of Inference
Resolution Reasoning

Soundness and Completeness
Practical Matters

Box Logic

Clausal Form

Clausal Form

Propositional resolution works only on expressions in *clausal form*.

Before we can apply resolution, we must first transform our sentences into clausal form.

$$(p \Rightarrow q) \longrightarrow \{\neg p, q\}$$

Fortunately, there is a simple algorithm for converting any set of Propositional Logic sentences into a logically equivalent set of sentences in clausal form.

Clausal Form

A *literal* is either an atomic sentence or a negation of an atomic sentence.

$$p, \neg p$$

A *clausal sentence* is either a literal or a disjunction of literals.

$$p, \quad \neg p, \quad p \vee \neg q$$

A *clause* is a set of literals.

$$\{p\}, \{\neg p\}, \{p, \neg q\}$$

Clauses are Disjunctions

$$p \vee \neg q \longrightarrow \{p, \neg q\}$$

What about the empty clause?

The empty clause {} is unsatisfiable.

Why? It is equivalent to an empty disjunction.

Conversion to Clausal Form

Implications Out:

$$\varphi_1 \Rightarrow \varphi_2 \quad \rightarrow \quad \neg\varphi_1 \vee \varphi_2$$

$$\varphi_1 \Leftrightarrow \varphi_2 \quad \rightarrow \quad (\neg\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \neg\varphi_2)$$

Conversion to Clausal Form

Implications Out:

$$\varphi_1 \Rightarrow \varphi_2 \quad \rightarrow \quad \neg\varphi_1 \vee \varphi_2$$

$$\varphi_1 \Leftrightarrow \varphi_2 \quad \rightarrow \quad (\neg\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \neg\varphi_2)$$

Negations In:

$$\neg\neg\varphi \quad \rightarrow \quad \varphi$$

$$\neg(\varphi_1 \wedge \varphi_2) \quad \rightarrow \quad \neg\varphi_1 \vee \neg\varphi_2$$

$$\neg(\varphi_1 \vee \varphi_2) \quad \rightarrow \quad \neg\varphi_1 \wedge \neg\varphi_2$$

Conversion to Clausal Form

Distribution

$$\varphi_1 \vee (\varphi_2 \wedge \varphi_3) \quad \rightarrow \quad (\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \varphi_3)$$

$$(\varphi_1 \wedge \varphi_2) \vee \varphi_3 \quad \rightarrow \quad (\varphi_1 \vee \varphi_3) \wedge (\varphi_2 \vee \varphi_3)$$

$$\varphi \vee (\varphi_1 \vee \ldots \vee \varphi_n) \quad \rightarrow \quad (\varphi \vee \varphi_1 \vee \ldots \vee \varphi_n)$$

$$(\varphi_1 \vee \ldots \vee \varphi_n) \vee \varphi \quad \rightarrow \quad (\varphi_1 \vee \ldots \vee \varphi_n \vee \varphi)$$

$$\varphi \wedge (\varphi_1 \wedge \ldots \wedge \varphi_n) \quad \rightarrow \quad (\varphi \wedge \varphi_1 \wedge \ldots \wedge \varphi_n)$$

$$(\varphi_1 \wedge \ldots \wedge \varphi_n) \wedge \varphi \quad \rightarrow \quad (\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \varphi)$$

$$a*(b+c) \longrightarrow (a*b + a*c)$$

$$(a+b)*c \longrightarrow (a*c + b*c)$$

$$a + (b + c) \longrightarrow a + b + c$$

$$a * (b * c) \longrightarrow a * b * c$$

Conversion to Clausal Form

Operators Out

$$\varphi_1 \wedge \ldots \wedge \varphi_n \quad \rightarrow \quad \begin{array}{l} \varphi_1 \\ \ldots \\ \varphi_n \end{array}$$

$$\varphi_1 \vee \ldots \vee \varphi_n \quad \rightarrow \quad \{\varphi_1, \ldots, \varphi_n\}$$

Example

$$(p \vee q) \wedge (r \vee r) \wedge \neg s \longrightarrow \begin{array}{c} p \vee q \\ r \vee r \\ \neg s \end{array} \longrightarrow \begin{array}{l} \{p, q\} \\ \{r\} \\ \{\neg s\} \end{array}$$

Example

$$g \wedge (r \Rightarrow f)$$

I $\quad g \wedge (\neg r \vee f)$

N $\quad g \wedge (\neg r \vee f)$

D $\quad g \wedge (\neg r \vee f)$

O $\quad \{g\}$

$\{\neg r, f\}$

Example

$$\neg(g \wedge (r \Rightarrow f))$$

$$\text{I} \quad \neg(g \wedge (\neg r \vee f))$$

$$\text{N} \quad \neg g \vee \neg(\neg r \vee f)$$

$$\text{N} \quad \neg g \vee (\neg\neg r \wedge \neg f)$$

$$\text{N} \quad \neg g \vee (r \wedge \neg f)$$

$$\text{D} \quad (\neg g \vee r) \wedge (\neg g \vee \neg f)$$

$$\text{O} \quad \{\neg g, r\}$$

$$\{\neg g, \neg f\}$$

Equivalence

Good News: The result of converting a set of sentences is logically equivalent to that set of sentences.

Upshot: Whatever we prove from clausal form is logically entailed by the original sentences.

Equivalence

Good News: The result of converting a set of sentences is logically equivalent to that set of sentences.

Upshot: Whatever we prove from clausal form is logically entailed by the original sentences.

Resolution Principle

Intuition

Premises: $\{p, q\}$ and $\{\neg q, r\}$

If q is false, then the first clause tells us p must be true.
If q is true, then the second clause tells us r must be true.

Conclusion: $\{p, r\}$

Resolution Principle

$$\{\phi_1, \ldots, \cancel{\chi}, \ldots, \phi_m\}$$
$$\{\psi_1, \ldots, \neg\cancel{\chi}, \ldots, \psi_n\}$$
$$\overline{\{\phi_1, \ldots, \phi_m, \psi_1, \ldots, \psi_n\}}$$

Example

$$\frac{\begin{array}{l}\{p, q\} \\ \{\neg q, r\}\end{array}}{\{p, r\}}$$

Example

$$\frac{\begin{array}{l}\{p, q, r\} \\ \{\neg p\}\end{array}}{\{q, r\}}$$

Example

$$\{p\}$$
$$\{\neg p\}$$

$$\{\}$$

Example

$$\frac{\begin{array}{l}\{p, q\} \\ \{\neg p, \neg q\}\end{array}}{\begin{array}{l}\{q, \neg q\} \\ \{p, \neg p\}\end{array}}$$

Example

$$\{p, q\}$$
$$\{\neg p, \neg q\}$$

$$\{\}$$

Wrong!!!

Implication Elimination

$$\frac{p \Rightarrow q \qquad p}{q}$$

$$\frac{\{\neg p, q\} \qquad \{p\}}{\{q\}}$$

Negation Introduction

$$\frac{p \Rightarrow q \qquad p \Rightarrow \neg q}{\neg p}$$

$$\frac{\{\neg p, q\} \qquad \{\neg p, \neg q\}}{\{\neg p\}}$$

Transitivity

1.	$p \Rightarrow q$	Premise
2.	$q \Rightarrow r$	Premise
3.	$\mid p$	Assumption
4.	$\mid q$	Implication Elimination: 1, 3
5.	$\mid r$	Implication Elimination: 2, 4
6.	$p \Rightarrow r$	Implication Introduction: 3, 5

Transitivity

$$\frac{\begin{array}{l} p \Rightarrow q \\ q \Rightarrow r \end{array}}{p \Rightarrow r} \qquad \frac{\begin{array}{l} \{\neg p, q\} \\ \{\neg q, r\} \end{array}}{\{\neg p, r\}}$$

Mary, Pat, and Quincy

If Mary loves Pat, then Mary loves Quincy. If it is Monday, then Mary loves Pat or Quincy. If it is Monday, does Mary love Quincy?

1.	$p \Rightarrow q$	Premise
2.	$m \Rightarrow p \vee q$	Premise
3.	$\quad m$	Assumption
4.	$\quad p \vee q$	Implication Elimination: 2, 3
5.	$\quad\quad q$	Assumption
6.	$\quad q \Rightarrow q$	Implication Introduction: 5, 5
7.	$\quad q$	Or Elimination: 4, 1, 6
8.	$m \Rightarrow q$	Implication Introduction: 3, 7

Example

$$\frac{\begin{array}{l} p \Rightarrow q \\ m \Rightarrow p \vee q \end{array}}{m \Rightarrow q}$$

$$\frac{\begin{array}{l} \{\neg p, q\} \\ \{\neg m, p, q\} \end{array}}{\{\neg m, q\}}$$

Resolution Reasoning

Resolution Derivation

A *resolution derivation* of a conclusion from a set of premises is a finite sequence of clauses terminating in the conclusion in which each clause is either a premise or the result of applying the resolution principle to earlier elements of the sequence.

Or Elimination

1.	$\{\neg p, r\}$	$p \Rightarrow r$
2.	$\{\neg q, r\}$	$q \Rightarrow r$
3.	$\{p, q\}$	$p \vee q$
4.	$\{q, r\}$	$1, 3$
5.	$\{r\}$	$2, 4$

Resolution Not Generatively Complete

Seemingly Bad News: Using the Resolution Principle alone, it is not possible to generate every clause that is logically entailed by a set of premises.

Premises: $\{p\}$ and $\{q\}$
Conclusion: $\{p, q\}$

Premises: none
Conclusion: $\{p, \neg p\}$

But resolution cannot generate these results!

Resolution Determines Unsatisfiability

Good News: If a set of clauses is unsatisfiable, it is possible to derive the empty clause using the Resolution Principle.

1.	$\{p, q\}$	Premise
2.	$\{p, \neg q\}$	Premise
3.	$\{\neg p, q\}$	Premise
4.	$\{\neg p, \neg q\}$	Premise
5.	$\{p\}$	$1, 2$
6	$\{\neg p\}$	$3, 4$
7	$\{\}$	$5, 6$

Resolution Method

Unsatisfiability Determination: If a set of clauses is unsatisfiable, it is possible to derive the empty clause using the Resolution Principle.

Unsatisfiability Theorem: $\Delta \models \varphi$ if and only if $\Delta \cup \{\neg \varphi\}$ is unsatisfiable.

Resolution Method: To prove that a set Δ of sentences logically entails a conclusion ϕ, write $\Delta \cup \{\neg\phi\}$ in clausal form and derive the empty clause.

Example

Given p, $(p \Rightarrow q)$, and $(p \Rightarrow q) \Rightarrow (q \Rightarrow r)$, prove r.

$$(p \Rightarrow q) \Rightarrow (q \Rightarrow r)$$

I $\neg(\neg p \vee q) \vee (\neg q \vee r)$

N $(\neg\neg p \wedge \neg q) \vee (\neg q \vee r)$

N $(p \wedge \neg q) \vee (\neg q \vee r)$

D $(p \vee (\neg q \vee r)) \wedge (\neg q \vee (\neg q \vee r))$

D $(p \vee \neg q \vee r) \wedge (\neg q \vee \neg q \vee r)$

O $\{p, \neg q, r\}$

$\{\neg q, r\}$

Proof

1.	$\{p\}$	p
2.	$\{\neg p, q\}$	$p \Rightarrow q$
3.	$\{p, \neg q, r\}$	$(p \Rightarrow q) \Rightarrow (q \Rightarrow r)$
4.	$\{\neg q, r\}$	$(p \Rightarrow q) \Rightarrow (q \Rightarrow r)$
5.	$\{\neg r\}$	Negated Goal
6.	$\{q\}$	1, 2
7.	$\{r\}$	6, 4
8.	$\{\}$	7, 5

Example

Show $(p \Rightarrow (q \Rightarrow p))$ is valid, i.e. $\{\} \models (p \Rightarrow (q \Rightarrow p))$.

$$\neg(p \Rightarrow (q \Rightarrow p))$$

I $\neg(\neg p \vee (\neg q \vee p))$

N $\neg\neg p \wedge \neg(\neg q \vee p)$

N $p \wedge \neg(\neg q \vee p)$

N $p \wedge (\neg\neg q \wedge \neg p)$

N $p \wedge (q \wedge \neg p)$

D $p \wedge q \wedge \neg p$

O $\{p\}$

$\{q\}$

$\{\neg p\}$

Proof

1.	$\{p\}$	$(p \Rightarrow (q \Rightarrow p))$
2.	$\{q\}$	$(p \Rightarrow (q \Rightarrow p))$
3.	$\{\neg p\}$	$(p \Rightarrow (q \Rightarrow p))$
4.	$\{\}$	$1, 3$

Soundness and Completeness

Logical Entailment and Provability

A set of premises Δ *logically entails* a conclusion φ (written $\Delta \models \varphi$) if and only if every interpretation that satisfies Δ also satisfies φ.

If there exists a proof of a sentence ϕ from a set Δ of premises using the rules of inference in R, we say that ϕ is *provable* from Δ using R (written $\Delta \vdash_R \varphi$).

Soundness and Completeness

A proof system is *sound* if and only if every provable conclusion is logically entailed.

$$\text{If } \Delta \vdash \varphi, \text{ then } \Delta \vDash \varphi.$$

A proof system is *complete* if and only if every logical conclusion is provable.

$$\text{If } \Delta \vDash \varphi, \text{ then } \Delta \vdash \varphi.$$

Resolution

Theorem: Resolution is sound and complete for Propositional Logic.

$$\Delta \vDash \varphi \text{ if and only if } \Delta \vdash_{\text{Resolution}} \varphi.$$

Upshot: The truth table method and the resolution method succeed in exactly the same cases!

Practical Matters

Two Finger Method

1. $\{p, q\}$ Premise
2. $\{p, \neg q\}$ Premise
3. $\{\neg p, q\}$ Premise
4. $\{\neg p, \neg q\}$ Premise

Two Finger Method

1.	$\{p, q\}$	Premise
2.	$\{p, \neg q\}$	Premise
3.	$\{\neg p, q\}$	Premise
4.	$\{\neg p, \neg q\}$	Premise

Two Finger Method

1.	$\{p, q\}$	Premise
2.	$\{p, \neg q\}$	Premise
3.	$\{\neg p, q\}$	Premise
4.	$\{\neg p, \neg q\}$	Premise
5.	$\{p\}$	$1, 2$

Two Finger Method

1.	$\{p, q\}$	Premise
2.	$\{p, \neg q\}$	Premise
3.	$\{\neg p, q\}$	Premise
4.	$\{\neg p, \neg q\}$	Premise
5.	$\{p\}$	1, 2

Two Finger Method

⋯⋯→	1.	$\{p, q\}$	Premise
	2.	$\{p, \neg q\}$	Premise
⟶	3.	$\{\neg p, q\}$	Premise
	4.	$\{\neg p, \neg q\}$	Premise
	5.	$\{p\}$	1, 2
	6.	$\{q\}$	1, 3

Two Finger Method

	1.	$\{p, q\}$	Premise
⇢	2.	$\{p, \neg q\}$	Premise
→	3.	$\{\neg p, q\}$	Premise
	4.	$\{\neg p, \neg q\}$	Premise
	5.	$\{p\}$	1, 2
	6.	$\{q\}$	1, 3
	7.	$\{\neg q, q\}$	2, 3
		...	

Two Finger Method

1.	$\{p, q\}$	Premise
2.	$\{p, \neg q\}$	Premise
3.	$\{\neg p, q\}$	Premise
4.	$\{\neg p, \neg q\}$	Premise
5.	$\{p\}$	1, 2
6.	$\{q\}$	1, 3
7.	$\{\neg q, q\}$	2, 3

...

Proof as Produced by Two-Finger Method

1.	$\{p, q\}$	$p \vee q$
2.	$\{p, \neg q\}$	$p \vee \neg q$
3.	$\{\neg p, q\}$	$\neg p \vee q$
4.	$\{\neg p, \neg q\}$	$\neg p \vee \neg q$
5.	$\{p\}$	1,2
6.	$\{q\}$	1,3
7.	$\{\neg q, q\}$	2,3
8.	$\{p, \neg p\}$	2,3
9.	$\{q, \neg q\}$	1,4
9.5	$\{p, \neg p\}$	1,4
10.	$\{\neg q\}$	2,4
11.	$\{\neg p\}$	3,4
12.	$\{q\}$	3,5
13.	$\{\neg q\}$	4,5
14.	$\{p\}$	2,6
15.	$\{\neg p\}$	4,6
16.	$\{p, q\}$	1,7
17.	$\{\neg q, p\}$	2,7
18.	$\{\neg p, q\}$	3,7
19.	$\{\neg q, \neg p\}$	4,7
20.	$\{q\}$	6,7

Proof as Produced by Two-Finger Method

21.	$\{\neg q, q\}$	7,7
22.	$\{\neg q, q\}$	7,7
23.	$\{q, p\}$	1,8
24.	$\{\neg q, p\}$	2,8
25.	$\{\neg p, q\}$	3,8
26.	$\{\neg p, \neg q\}$	4,8
27.	$\{p\}$	5,8
28.	$\{\neg p, p\}$	8,8
29.	$\{\neg p, p\}$	8,8
30.	$\{p, q\}$	1,9
31.	$\{\neg q, p\}$	2,9
32.	$\{\neg p, q\}$	3,9
33.	$\{\neg q, \neg p\}$	4,9
34.	$\{q\}$	6,9
35.	$\{\neg q, q\}$	7,9
36.	$\{q, \neg q\}$	9,9
37.	$\{q, \neg q\}$	9,9
38.	$\{p\}$	1,10
39.	$\{\neg p\}$	3,10
40.	$\{\}$	6,10

Proof with Identical Clause Elimination

1.	$\{p, q\}$	$p \vee q$
2.	$\{p, \neg q\}$	$p \vee \neg q$
3.	$\{\neg p, q\}$	$\neg p \vee q$
4.	$\{\neg p, \neg q\}$	$\neg p \vee \neg q$
5.	$\{p\}$	1,2
6.	$\{q\}$	1,3
7.	$\{\neg q, q\}$	2,3
8.	$\{p, \neg p\}$	2,3
9.	$\{q, \neg q\}$	1,4
10.	$\{\neg q\}$	2,4
11.	$\{\neg p\}$	3,4
12.	$\{\}$	6,10

Identical Clause Elimination

Metatheorem: There is a resolution refutation of Δ if and only if there is a resolution refutation from Δ in which no clause occurs twice. (Obviously.)

Upshot: If you generate a clause that is already in the proof, do not include it again.

Metatheorem: There are only finitely many clauses that can be formed from a finite set of proposition constants.

Upshot: You will eventually run out of things to do. So possible to terminate search in finite time!!!

Tautology Elimination

A *tautology* is a clause with a complementary pair of literals.

$$\{q, \neg q\}$$

$$\{p, q, r, \neg q\}$$

Metatheorem: There is a resolution refutation of Δ if and only if there is a resolution refutation from Δ with tautology elimination.

Proof with ICE and TE

1.	$\{p,q\}$	$p \vee q$
2.	$\{p,\neg q\}$	$p \vee \neg q$
3.	$\{\neg p,q\}$	$\neg p \vee q$
4.	$\{\neg p,\neg q\}$	$\neg p \vee \neg q$
5.	$\{p\}$	1,2
6.	$\{q\}$	1,3
7.	$\{\neg q\}$	2,4
8.	$\{\neg p\}$	3,4
9.	$\{\}$	6,7

Motivation for Subsumption

1.	$\{p, q\}$	Premise
2.	$\{p, q, r\}$	Premise
3.	$\{q, r\}$	Premise
4.	$\{\neg p\}$	Premise
5.	$\{\neg q\}$	Premise
6.	$\{\neg r\}$	Premise

Propositional Subsumption

A clause Φ *subsumes* Ψ if and only if Φ is a subset of Ψ.

Example: $\{p, q\}$ subsumes $\{p, q, r\}$

Metatheorem: There is a resolution refutation of Δ if and only if there is a resolution refutation from Δ with Propositional Subsumption.

Note

The resolution of two clauses sometimes produces a clause that subsumes one of its parents.

1.	$\{p\}$	Premise
2.	$\{\neg r, q\}$	Premise
3.	$\{r\}$	Premise
4.	$\{\neg p, \neg q, \neg r\}$	Premise
5.	$\{\neg q, \neg r\}$	1,4
6.	$\{\neg r\}$	2,5
7.	$\{\}$	3,6

Example of Pure Literal Elimination

1.	$\{p, q\}$	Premise
2.	$\{\neg p, r\}$	Premise
3.	$\{\neg q, r\}$	Premise
4.	$\{\neg q, s\}$	Premise
5.	$\{\neg r\}$	Goal

Pure Literal Elimination

A *literal* in a database is *pure* if and only if there is no complementary occurrence of the literal in the database.

A *clause* is *superfluous* if and only if it contains a pure literal.

Metatheorem: There is a resolution refutation of Δ if and only if there is a resolution refutation from Δ in which all superfluous clauses are removed.

Example

1.	$\{p, q\}$	Premise
2.	$\{\neg p, r\}$	Premise
3.	$\{\neg q, r\}$	Premise
4.	$\{\neg q, s\}$	Premise
5.	$\{\neg r\}$	Goal

Note

The removal of a superfluous clause may create new pure literals and new superfluous clauses.

1.	$\{p, q\}$	$p \vee q$
2.	$\{\neg p, r\}$	$p \Rightarrow r$
3.	$\{\neg q, r\}$	$q \Rightarrow r$
4.	$\{\neg q, s, t\}$	$q \Rightarrow s \vee t$
5.	$\{\neg r\}$	$\neg r$
6.	$\{\neg t\}$	$\neg t$

Strategies

Elimination Strategies (Constraints on clauses):

- Identical Clause Elimination
- Pure Literal Elimination
- Tautology Elimination
- Subsumption Elimination

Restriction Strategies (Constraints on inferences):

- Unit Restriction
- Input Restriction
- Linear Restriction
- Set of Support Restriction

Word of the Day

Robinson

Word of the Day

Robinson

Resolution Tools

Course Website

http://logica.stanford.edu