Introduction to Logic

Refutation Proofs

Michael Genesereth
Computer Science Department
Stanford University
Popular Types of Proof Systems:
  Direct Proofs (Hilbert)
  Natural Deduction (Fitch)
  → Refutation proofs (Resolution / Robinson)

Others:
  Gentzen Systems
  Sequent Calculi
  and so forth
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*. 
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
2. premise
3. \(\neg\) conclusion
...
\(n\). contradiction

A refutation proof is a proof by contradiction.
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
Propositional resolution works only on expressions in *clausal form*.

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

\[(p \Rightarrow q) \rightarrow \{-p, q\}\]

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

\[ p, \neg p, \quad p \lor \neg q \]

A *clause* is a set of literals.

\[ \{p\}, \{\neg p\}, \{p, \neg q\} \]
What about the empty clause?

The empty clause \{\}\ is unsatisfiable.

Why? It is equivalent to an empty disjunction.
Implications Out:

\[ \varphi_1 \Rightarrow \varphi_2 \quad \Rightarrow \quad \neg \varphi_1 \lor \varphi_2 \]

\[ \varphi_1 \iff \varphi_2 \quad \Rightarrow \quad (\neg \varphi_1 \lor \varphi_2) \land (\varphi_1 \lor \neg \varphi_2) \]
Conversion to Clausal Form

Implications Out:

\[
\varphi_1 \implies \varphi_2 \implies \neg \varphi_1 \lor \varphi_2
\]

\[
\varphi_1 \iff \varphi_2 \implies (\neg \varphi_1 \lor \varphi_2) \land (\varphi_1 \lor \neg \varphi_2)
\]

Negations In:

\[
\neg \neg \varphi \implies \varphi
\]

\[
\neg (\varphi_1 \land \varphi_2) \implies \neg \varphi_1 \lor \neg \varphi_2
\]

\[
\neg (\varphi_1 \lor \varphi_2) \implies \neg \varphi_1 \land \neg \varphi_2
\]
Conversion to Clausal Form

**Distribution**

\[ \varphi_1 \lor (\varphi_2 \land \varphi_3) \rightarrow (\varphi_1 \lor \varphi_2) \land (\varphi_1 \lor \varphi_3) \]

\[ (\varphi_1 \land \varphi_2) \lor \varphi_3 \rightarrow (\varphi_1 \lor \varphi_3) \land (\varphi_2 \lor \varphi_3) \]

\[ \varphi \lor (\varphi_1 \lor \ldots \lor \varphi_n) \rightarrow (\varphi \lor \varphi_1 \lor \ldots \lor \varphi_n) \]

\[ (\varphi_1 \lor \ldots \lor \varphi_n) \lor \varphi \rightarrow (\varphi_1 \lor \ldots \lor \varphi_n \lor \varphi) \]

\[ \varphi \land (\varphi_1 \land \ldots \land \varphi_n) \rightarrow (\varphi \land \varphi_1 \land \ldots \land \varphi_n) \]

\[ (\varphi_1 \land \ldots \land \varphi_n) \land \varphi \rightarrow (\varphi_1 \land \ldots \land \varphi_n \land \varphi) \]

\[ a^*(b + c) \rightarrow (a^*b + a^*c) \]

\[ (a + b)^*c \rightarrow (a^*c + b^*c) \]

\[ a + (b + c) \rightarrow a + b + c \]

\[ a * (b * c) \rightarrow a * b * c \]
Conversion to Clausal Form

Operators Out

\[ \varphi_1 \land \ldots \land \varphi_n \rightarrow \varphi_1 \]

\[ \ldots \]

\[ \varphi_n \]

\[ \varphi_1 \lor \ldots \lor \varphi_n \rightarrow \{\varphi_1, \ldots, \varphi_n\} \]

Example

\[ (p \lor q) \land (r \lor r) \land \neg s \rightarrow p \lor q \rightarrow \{p, q\} \]
\[ \quad r \lor r \rightarrow \{r\} \]
\[ \quad \neg s \rightarrow \{-s\} \]
Example

\[
g \land (r \Rightarrow f)
\]

I  \( g \land (\neg r \lor f) \)

N  \( g \land (\neg r \lor f) \)

D  \( g \land (\neg r \lor f) \)

O  \{g\}

\{\neg r, f\}
\(\neg (g \land (r \Rightarrow f))\)

I \(\neg (g \land (\neg r \lor f))\)

N \(\neg g \lor \neg (\neg r \lor f)\)

N \(\neg g \lor (\neg \neg r \land \neg f)\)

N \(\neg g \lor (r \land \neg f)\)

D \((\neg g \lor r) \land (\neg g \lor \neg f)\)

O \(\{\neg g, r\}\)

\(\{\neg g, \neg f\}\)
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.
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
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

\[
\{ \phi_1, \ldots, \chi, \ldots, \phi_m \} \\
\{ \psi_1, \ldots, \neg \chi, \ldots, \psi_n \} \\
\hline \\
\{ \phi_1, \ldots, \phi_m, \psi_1, \ldots, \psi_n \}
\]
Example

\[
\begin{align*}
\{p, q\} \\
\{\neg q, r\} \\
\hline
\{p, r\}
\end{align*}
\]
Example

\{p, q, r\}

\{\neg p\}

\hline

\{q, r\}
Example

\[ \{p\} \]
\[ \{\neg p\} \]
\[ \rightarrow \]
\[ \{\} \]
Example

\[
\begin{align*}
\{p, q\} \\
\{\neg p, \neg q\} \\
\hline \\
\{q, \neg q\} \\
\{p, \neg p\}
\end{align*}
\]
Example

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

\hline
\{}
\quad \text{Wrong}!!!
Implication Elimination

\[ p \Rightarrow q \]
\[ p \]
\[ q \]

\[ \{ \neg p, q \} \]
\[ \{ p \} \]
\[ \{ q \} \]
Negation Introduction

\[
\begin{align*}
\neg p, q & \quad \{\neg p, q\} \\
\neg p, \neg q & \quad \{\neg p, \neg q\} \\
\neg p & \quad \{\neg p\}
\end{align*}
\]
1. \( p \Rightarrow q \) \hspace{1cm} \text{Premise}
2. \( q \Rightarrow r \) \hspace{1cm} \text{Premise}
3. \( p \) \hspace{1cm} \text{Assumption}
4. \( q \) \hspace{1cm} \text{Implication Elimination: 1, 3}
5. \( r \) \hspace{1cm} \text{Implication Elimination: 2, 4}
6. \( p \Rightarrow r \) \hspace{1cm} \text{Implication Introduction: 3, 5}
Transitivity

\[ p \Rightarrow q \]
\[ q \Rightarrow r \]
\[ \begin{array}{c}
\hline
p \Rightarrow r \\
\end{array} \]
\[ \{ \neg p, q \} \]
\[ \{ \neg q, r \} \]
\[ \{ \neg p, r \} \]
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. \( m \Rightarrow p \lor q \)  Premise
3. \( m \)  Assumption

4. \( p \lor q \)  Implication Elimination: 2, 3

5. \( q \)  Assumption

6. \( q \Rightarrow q \)  Implication Introduction: 5, 5
7. \( q \)  Or Elimination: 4, 1, 6

8. \( m \Rightarrow q \)  Implication Introduction: 3, 7
Example

\[ p \Rightarrow q \]
\[ m \Rightarrow p \lor q \]

\[ m \Rightarrow q \]

\[ \{ \neg p, q \} \]
\[ \{ \neg m, p, q \} \]

\[ \{ \neg m, q \} \]
Resolution Reasoning
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\} \quad p \Rightarrow r
2. \{\neg q, r\} \quad q \Rightarrow r
3. \{p, q\} \quad p \lor q
4. \{q, r\} \quad 1, 3
5. \{r\} \quad 2, 4
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!
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
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 $\Delta$ of sentences logically entails a conclusion $\phi$, write $\Delta \cup \{\neg \phi\}$ in clausal form and derive the empty clause.
Given \( p, (p \Rightarrow q), \) and \( (p \Rightarrow q) \Rightarrow (q \Rightarrow r) \), prove \( r \).

\[
(p \Rightarrow q) \Rightarrow (q \Rightarrow r)
\]

I  \( \neg(\neg p \lor q) \lor (\neg q \lor r) \)

N  \( (\neg p \land \neg q) \lor (\neg q \lor r) \)

N  \( (p \land \neg q) \lor (\neg q \lor r) \)

D  \( (p \lor (\neg q \lor r)) \land (\neg q \lor (\neg q \lor r)) \)

D  \( (p \lor \neg q \lor r) \land (\neg q \lor \neg q \lor r) \)

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

\{\neg q, r\}
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. \(\emptyset \vdash (p \Rightarrow (q \Rightarrow p))\).

\[
\neg(p \Rightarrow (q \Rightarrow p))
\]

I \(\neg(\neg p \lor (\neg q \lor p))\)

N \(\neg\neg p \land \neg(\neg q \lor p)\)

N \(p \land \neg(\neg q \lor p)\)

N \(p \land (\neg\neg q \land \neg p)\)

N \(p \land (q \land \neg p)\)

D \(p \land q \land \neg p\)

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

If there exists a proof of a sentence $\phi$ from a set $\Delta$ of premises using the rules of inference in $R$, we say that $\phi$ is \textit{provable} from $\Delta$ using $R$ (written $\Delta \vdash_R \varphi$).
A proof system is *sound* if and only if every provable conclusion is logically entailed.

If $\Delta \vdash \varphi$, then $\Delta \models \varphi$.

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

If $\Delta \models \varphi$, then $\Delta \vdash \varphi$. 
Theorem: Resolution is sound and complete for Propositional Logic.

\[ \Delta \models \varphi \text{ if and only if } \Delta \vdash_{\text{Resolution}} \varphi. \]

Upshot: The truth table method and the resolution method succeed in exactly the same cases!
Practical Matters
1. \{p, q\}  Premise
2. \{p, \neg q\}  Premise
3. \{\neg p, q\}  Premise
4. \{\neg p, \neg q\}  Premise
1. \{p, q\}   Premise
2. \{p, \neg q\}   Premise
3. \{\neg p, q\}   Premise
4. \{\neg p, \neg q\}   Premise
1. \{p, q\}  
2. \{p, \neg q\}  
3. \{\neg p, q\}  
4. \{\neg p, \neg q\}  
5. \{p\}  

Premise  
Premise  
Premise  
Premise  

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

Two Finger Method
Proof as Produced by Two-Finger Method

1. \{p, q\} \quad p \vee q
2. \{p, \neg q\} \quad p \vee \neg q
3. \{\neg p, q\} \quad \neg p \vee q
4. \{\neg p, \neg q\} \quad \neg p \vee \neg q
5. \{p\} \quad 1, 2
6. \{q\} \quad 1, 3
7. \{\neg q, q\} \quad 2, 3
8. \{p, \neg p\} \quad 2, 3
9. \{q, \neg q\} \quad 1, 4
9.5 \{p, \neg p\} \quad 1, 4
10. \{\neg q\} \quad 2, 4
11. \{\neg p\} \quad 3, 4
12. \{q\} \quad 3, 5
13. \{\neg q\} \quad 4, 5
14. \{p\} \quad 2, 6
15. \{\neg p\} \quad 4, 6
16. \{p, q\} \quad 1, 7
17. \{\neg q, p\} \quad 2, 7
18. \{\neg p, q\} \quad 3, 7
19. \{\neg q, \neg p\} \quad 4, 7
20. \{q\} \quad 6, 7
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>21.</td>
<td>{-q,q}</td>
<td>7,7</td>
</tr>
<tr>
<td>22.</td>
<td>{-q,q}</td>
<td>7,7</td>
</tr>
<tr>
<td>23.</td>
<td>{q,p}</td>
<td>1,8</td>
</tr>
<tr>
<td>24.</td>
<td>{-q,p}</td>
<td>2,8</td>
</tr>
<tr>
<td>25.</td>
<td>{-p,q}</td>
<td>3,8</td>
</tr>
<tr>
<td>26.</td>
<td>{-p,\neg q}</td>
<td>4,8</td>
</tr>
<tr>
<td>27.</td>
<td>{p}</td>
<td>5,8</td>
</tr>
<tr>
<td>28.</td>
<td>{-p,p}</td>
<td>8,8</td>
</tr>
<tr>
<td>29.</td>
<td>{-p,p}</td>
<td>8,8</td>
</tr>
<tr>
<td>30.</td>
<td>{p,q}</td>
<td>1,9</td>
</tr>
<tr>
<td>31.</td>
<td>{-q,p}</td>
<td>2,9</td>
</tr>
<tr>
<td>32.</td>
<td>{-p,q}</td>
<td>3,9</td>
</tr>
<tr>
<td>33.</td>
<td>{-q,\neg p}</td>
<td>4,9</td>
</tr>
<tr>
<td>34.</td>
<td>{q}</td>
<td>6,9</td>
</tr>
<tr>
<td>35.</td>
<td>{-q,q}</td>
<td>7,9</td>
</tr>
<tr>
<td>36.</td>
<td>{q,\neg q}</td>
<td>9,9</td>
</tr>
<tr>
<td>37.</td>
<td>{q,\neg q}</td>
<td>9,9</td>
</tr>
<tr>
<td>38.</td>
<td>{p}</td>
<td>1,10</td>
</tr>
<tr>
<td>39.</td>
<td>{-p}</td>
<td>3,10</td>
</tr>
<tr>
<td>40.</td>
<td>{}</td>
<td>6,10</td>
</tr>
</tbody>
</table>
Proof with Identical Clause Elimination

1. \{p, q\} \quad p \lor q
2. \{p, \neg q\} \quad p \lor \neg q
3. \{\neg p, q\} \quad \neg p \lor q
4. \{\neg p, \neg q\} \quad \neg p \lor \neg q
5. \{p\} \quad 1, 2
6. \{q\} \quad 1, 3
7. \{\neg q, q\} \quad 2, 3
8. \{p, \neg p\} \quad 2, 3
9. \{q, \neg q\} \quad 1, 4
10. \{\neg q\} \quad 2, 4
11. \{\neg p\} \quad 3, 4
12. \{\} \quad 6, 10
Identical Clause Elimination

Metatheorem: There is a resolution refutation of $\Delta$ if and only if there is a resolution refutation from $\Delta$ 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!!!
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 \(\Delta\) if and only if there is a resolution refutation from \(\Delta\) with tautology elimination.
1. \{p,q\} \quad p \lor q \\
2. \{p,\lnot q\} \quad p \lor \lnot q \\
3. \{\lnot p,q\} \quad \lnot p \lor q \\
4. \{\lnot p,\lnot q\} \quad \lnot p \lor \lnot q \\
5. \{p\} \quad 1,2 \\
6. \{q\} \quad 1,3 \\
7. \{\lnot q\} \quad 2,4 \\
8. \{\lnot p\} \quad 3,4 \\
9. \{} \quad 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
A clause $\Phi$ subsumes $\Psi$ if and only if $\Phi$ is a subset of $\Psi$.

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

Metatheorem: There is a resolution refutation of $\Delta$ if and only if there is a resolution refutation from $\Delta$ with Propositional Subsumption.
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
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 \( \Delta \) if and only if there is a resolution refutation from \( \Delta \) 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
The removal of a superfluous clause may create new pure literals and new superfluous clauses.

1. \( \{p,q\} \quad p \lor q \)
2. \( \{\neg p,r\} \quad p \Rightarrow r \)
3. \( \{\neg q,r\} \quad q \Rightarrow r \)
4. \( \{\neg q,s,t\} \quad q \Rightarrow s \lor t \)
5. \( \{\neg r\} \quad \neg r \)
6. \( \{\neg t\} \quad \neg t \)
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
Robinson
Robinson
Resolution Tools
http://logica.stanford.edu