# A list of proof systems you are learning

- Prove by the truth tables: prop1.ppt
- Prove by S/I rules (apply to WFFs directly, but need many rules): prop2.ppt
- Prove by resolution (need to convert WFFs to CNF, but need only one rule): prop3/4.ppt

# Propositional Resolution

Based on Michael Genesereth
Stanford University

Modified by Charles Ling, for CS2209
Use with permission

# Propositional Resolution

*Propositional resolution* is a rule of inference.

Using propositional resolution alone (without other rules of inference), it is possible to build a theorem prover that is sound and complete for all of Propositional Logic.

The search space using propositional resolution is much smaller than for Modus Ponens and the Standard Axiom Schemata.

# Clausal Form (Review)

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

Fortunately, it is possible to convert any set of propositional calculus sentences into an equivalent set of sentences in clausal form (same as CNF)

# Clausal Form

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

$$P$$
$$\sim P$$

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

$$P$$
$$\sim P$$
$$P \vee Q$$

A *clause* is a set of literals (same as above).

$$\{P\}$$
$$\{\sim P\}$$
$$\{P, Q\}$$

# Empty Sets

The empty clause {} is unsatisfiable.

Why? It is equivalent to an empty disjunction.

# Conversion to Clausal Form (or CNF)

Implications Out:

$$\varphi_1 \supset \varphi_2 \rightarrow \sim \varphi_1 \vee \varphi_2$$

$$\varphi_1 \equiv \varphi_2 \rightarrow (\sim \varphi_1 \vee \varphi_2) \bullet (\varphi_1 \vee \sim \varphi_2)$$

Negations In:

$$\sim\sim \varphi \rightarrow \varphi$$

$$\sim (\varphi_1 \bullet \varphi_2) \rightarrow \sim \varphi_1 \vee \sim \varphi_2$$

$$\sim (\varphi_1 \vee \varphi_2) \rightarrow \sim \varphi_1 \bullet \sim \varphi_2$$

# Conversion to Clausal Form

Distribution

$$\varphi_1 \vee (\varphi_2 \bullet \varphi_3) \quad \rightarrow \quad (\varphi_1 \vee \varphi_2) \bullet (\varphi_1 \vee \varphi_n)$$

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

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

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

$$\varphi_1 \bullet (\varphi_2 \bullet \varphi_3) \quad \rightarrow \quad \varphi_1 \bullet \varphi_2 \bullet \varphi_3$$

$$(\varphi_1 \bullet \varphi_2) \bullet \varphi_3 \quad \rightarrow \quad \varphi_1 \bullet \varphi_2 \bullet \varphi_3$$

Operators Out

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

$$\varphi_1 \bullet \ldots \bullet \varphi_n \quad \rightarrow \quad \varphi_1, \ldots, \varphi_n$$

# Example

| | |
|---|---|
| | $G \bullet (R \supset F)$ |
| I | $G \bullet (\sim R \vee F)$ |
| N | $G \bullet (\sim R \vee F)$ |
| D | $G \bullet (\sim R \vee F)$ |
| O | $\{G\}$ |
| | $\{\sim R, F\}$ |

# Example

| | |
|---|---|
| | $\sim(G \bullet (R \supset F))$ |
| I | $\sim(G \bullet (\sim R \vee F))$ |
| N | $\sim G \vee \sim(\sim R \vee F))$ |
| | $\sim G \vee (\sim\sim R \bullet \sim F)$ |
| | $\sim G \vee (R \bullet \sim F)$ |
| D | $(\sim G \vee R) \bullet (\sim G \vee \sim F)$ |
| O | $\{\sim G, R\}$ |
| | $\{\sim G, \sim F\}$ |

# Resolution Principle

General:

$$\frac{\{\varphi_1, \ldots, \chi, \ldots, \varphi_m\} \quad \{\psi_1, \ldots, \sim\chi, \ldots, \psi_n\}}{\{\varphi_1, \ldots, \varphi_m, \psi_1, \ldots, \psi_n\}}$$

Example:

$$\frac{\{P, Q\} \quad \{\sim P, R\}}{\{Q, R\}}$$

# Issues

Collapse

$$\frac{\begin{array}{l}\{\sim P, Q\} \\ \{P, Q\}\end{array}}{\{Q\}}$$

Singletons

$$\frac{\begin{array}{l}\{\sim P, Q\} \\ \{P\}\end{array}}{\{Q\}}$$

$$\frac{\begin{array}{l}\{P\} \\ \{\sim P\}\end{array}}{\{\}}$$

# Issues

Multiple Conclusions

$$\frac{\begin{array}{c}\{P, Q\} \\ \{\sim P, \sim Q\}\end{array}}{\begin{array}{c}\{P, \sim P\} = T \\ \{Q, \sim Q\} = T\end{array}}$$

Single Application Only

$$\frac{\begin{array}{c}\{P, Q\} \\ \{\sim P, \sim Q\}\end{array}}{\{\} \text{ Wrong!!}}$$

# Special Cases

**Modus Ponens**

$$\begin{array}{l} P \supset Q \\ P \\ \hline Q \end{array}$$

$$\begin{array}{l} \{\sim P, Q\} \\ \{P\} \\ \hline \{Q\} \end{array}$$

**Modus Tolens**

$$\begin{array}{l} P \supset Q \\ \sim Q \\ \hline \sim P \end{array}$$

$$\begin{array}{l} \{\sim P, Q\} \\ \{\sim Q\} \\ \hline \{\sim P\} \end{array}$$

**Chaining**

$$\begin{array}{l} P \supset Q \\ Q \supset R \\ \hline P \supset R \end{array}$$

$$\begin{array}{l} \{\sim P, Q\} \\ \{\sim Q, R\} \\ \hline \{\sim P, R\} \end{array}$$

# Incompleteness?

Propositional Resolution is not *generatively* complete.

We cannot generate $P \supset (Q \supset P)$ using propositional resolution. There are no premises. Consequently, there are no conclusions.

# Answer

This apparent problem disappears if we take the clausal form of the premises (if any) together with the negated goal (also in clausal form), and try to derive the empty clause.

General Method: To determine whether a set $\Delta$ of sentences logically entails a sentence $\phi$, rewrite $\Delta \vee \{\sim\phi\}$ in clausal form and try to derive the empty clause using the resolution rule of inference.

# Example

(Prove a wff from empty premise)

| | |
|---|---|
| | $\sim (P \supset (Q \supset P))$ |
| I | $\sim (\sim P \vee \sim Q \vee P)$ |
| N | $\sim\sim P \bullet \sim\sim Q \bullet \sim P$ |
| | $P \bullet Q \bullet \sim P$ |
| D | $P \bullet Q \bullet \sim P$ |
| O | $\{P\}$ |
| | $\{Q\}$ |
| | $\{\sim P\}$ |

# Example

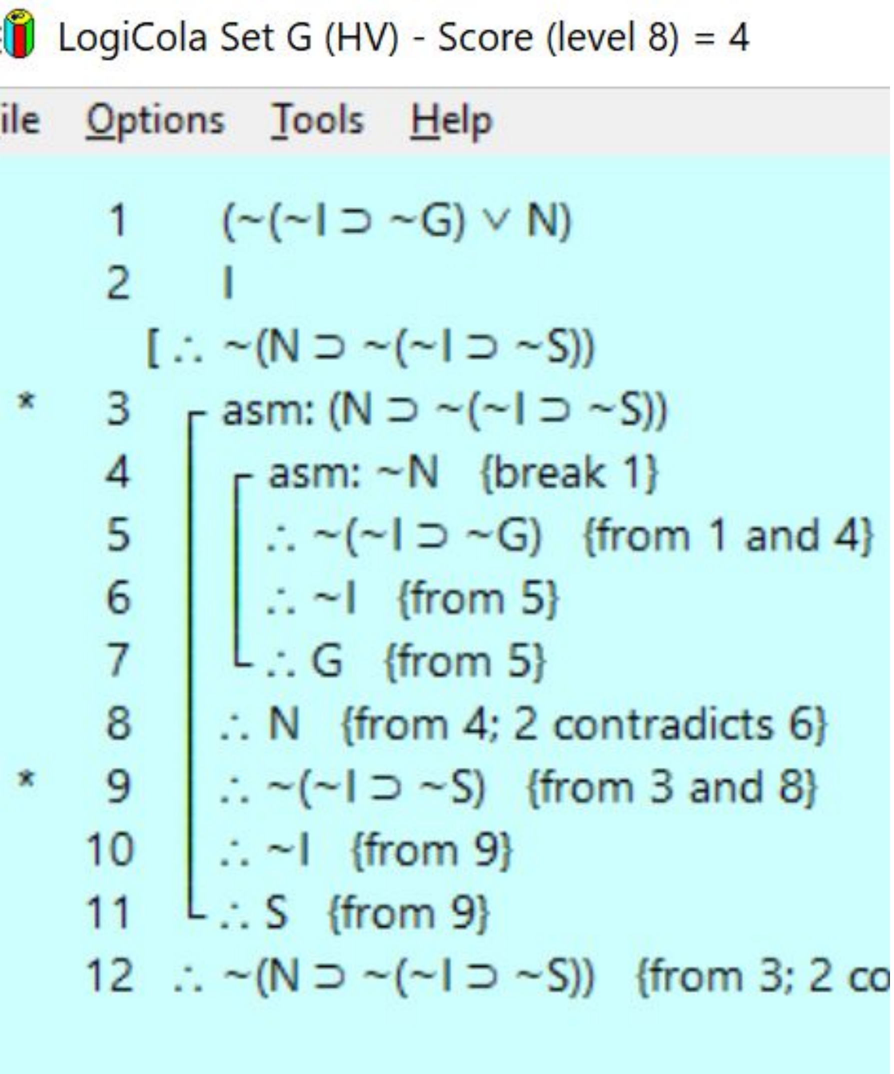
If Mary loves Pat, then Mary loves Quincy. If it is Monday, Mary loves Pat or Quincy. Prove that, if it is Monday, then Mary loves Quincy.

**$P \rightarrow Q$**
**$M \rightarrow (P \vee Q)$**
**Prove: $M \rightarrow Q$**

1. **Convert each premise to clause(s)**
2. **Convert the negated goal to clause(s)**
3. **Apply resolution, to derive empty clause**

| | | |
|---|---|---|
| 1. | $\{\sim P, Q\}$ | Premise |
| 2. | $\{\sim M, P, Q\}$ | Premise |
| 3. | $\{M\}$ | Negated Goal |
| 4. | $\{\sim Q\}$ | Negated Goal |
| 5. | $\{P, Q\}$ | 3,2 |
| 6. | $\{Q\}$ | 5,1 |
| 7. | $\{\}$ | 6,4 |

# Lets try to do resolution proof




**1 = ~(I v ~G v N)**

**= ~I * G * ~N**

**= {~I}, {G}, {~N}**

**2 = I = {I}**

1 $(A \supset (B \cdot C))$

2 $(B \supset (A \cdot C))$

$[\therefore ((A \vee B) \supset C)$

3 asm: $\sim((A \vee B) \supset C)$

4 $\therefore (A \vee B)$ {from 3}

5 $\therefore \sim C$ {from 3}

6 asm: $A$ {break up 1}

7 $\therefore (B \cdot C)$ {from 1 and 6}

8 $\therefore B$ {from 7}

9 $\therefore C$ {from 7}

10 $\therefore \sim A$ {from 6; 5 contradicts 9} $\Leftarrow$

11 $\therefore B$ {from 4 and 10}

12 $\therefore (A \cdot C)$ {from 2 and 11}

13 $\therefore A$ {from 12} $\Leftarrow$

14 $\therefore ((A \vee B) \supset C)$ {from 3; 10 contradicts 13}

# Example

Heads you win. Tails I lose. Show that you always win.

**$H \rightarrow Y$**
**$T \rightarrow \sim M$**
**Prove Y**
**How??**
**Need background knowledge**
**$\sim H \rightarrow T$**
**$H \rightarrow \sim T$**
**$\sim M \rightarrow Y$**
**$M \rightarrow \sim Y$**

**(Why use logic for real-world common sense reasoning is hard!**

| | | |
|---|---|---|
| 1. | $\{\sim H, Y\}$ | Premise |
| 2. | $\{\sim T, \sim M\}$ | Premise |
| 3. | $\{H, T\}$ | Premise |
| 4. | $\{\sim H, \sim T\}$ | Premise |
| 5. | $\{M, Y\}$ | Premise |
| 6. | $\{\sim M, \sim Y\}$ | Premise |
| 7. | $\{\sim Y\}$ | Negated Goal |
| 8. | $\{T, Y\}$ | 3,1 |
| 9. | $\{\sim M, Y\}$ | 8, 2 |
| 10. | $\{Y\}$ | 9, 5 |
| 11. | $\{\}$ | 10, 7 |

# Soundness and Completeness

A sentence is *provable* from a set of sentences by propositional resolution if and only if there is a derivation of the empty clause from the clausal form of $\Delta \vee \{\sim\phi\}$.

Theorem: Propositional Resolution is sound and complete, i.e. $\Delta$ |= $\phi$ if and only if $\Delta$ |- $\phi$.

# Two Finger Method

```
function tfm (Δ)                    ; Δ is a linked list of clauses
  {var fast ← Δ;
   var slow ← Δ;
   do {if slow = [] then return failure;
       Δ ← concat(Δ, resolvents(first(fast),first(slow)));
       if {} ∈ Δ then return Δ;
       if fast=slow then {fast ← Δ; slow←next(slow)}
                     else   fast←next(fast)}}
```

function $\quad \text{resolvents}(\Phi_1, \Phi_2)$

$$\{\Phi_1 - \{\varphi\} \cup \Phi_2 - \{\sim\varphi\} \mid \varphi \in \Phi_1 \text{and} \sim \varphi \in \Phi_2\} \cup$$

$$\{\Phi_1 - \{\sim\varphi\} \cup \Phi_2 - \{\varphi\} \mid \sim \varphi \in \Phi_1 \text{ and } \quad \varphi \in \Phi_2\}$$

- Slow (slowly move down to bottom, which is expanding)
- Fast, from the very top to slow (reset to top every time when it meets slow)

# Two Finger Method

| | | |
|---|---|---|
| 1. | $\{P, Q\}$ | Premise |
| 2. | $\{\sim P, R\}$ | Premise |
| 3. | $\{\sim Q, R\}$ | Premise |
| 4. | $\{\sim R\}$ | Premise |
| 5. | $\{Q, R\}$ | 1,2 |
| 6. | $\{P, R\}$ | 1,3 |
| 7. | $\{\sim P\}$ | 2,4 |
| 8. | $\{\sim Q\}$ | 3, 4 |
| 9. | $\{R\}$ | 3, 5 |
| 10. | $\{Q\}$ | 4, 5 |
| 11. | $\{R\}$ | 2,6 |
| 12. | $\{P\}$ | 4,6 |
| 13. | $\{Q\}$ | 1,7 |
| 14. | $\{R\}$ | 6,7 |
| 15. | $\{P\}$ | 1,8 |
| 16. | $\{R\}$ | 5,8 |
| 17. | $\{\}$ | 4,9 |

# TFM With Identical Clause Elimination

| | | |
|---|---|---|
| 1. | $\{P, Q\}$ | Premise |
| 2. | $\{\sim P, R\}$ | Premise |
| 3. | $\{\sim Q, R\}$ | Premise |
| 4. | $\{\sim R\}$ | Premise |
| 5. | $\{Q, R\}$ | 1,2 |
| 6. | $\{P, R\}$ | 1,3 |
| 7. | $\{\sim P\}$ | 2,4 |
| 8. | $\{\sim Q\}$ | 3, 4 |
| 9. | $\{R\}$ | 3, 5 |
| 10. | $\{Q\}$ | 4, 5 |
| 11. | $\{P\}$ | 4, 6 |
| 12. | $\{\}$ | 4, 9 |

# TFM With ICE, Complement Detection

| | | |
|---|---|---|
| 1. | $\{P, Q\}$ | Premise |
| 2. | $\{\sim P, R\}$ | Premise |
| 3. | $\{\sim Q, R\}$ | Premise |
| 4. | $\{\sim R\}$ | Premise |
| 5. | $\{Q, R\}$ | 1,2 |
| 6. | $\{P, R\}$ | 1,3 |
| 7. | $\{\sim P\}$ | 2,4 |
| 8. | $\{\sim Q\}$ | 3, 4 |
| 9. | $\{R\}$ | 3, 5 |
| 10. | $\{\}$ | 4, 9 |

# Termination

Theorem: There is a resolution derivation of a conclusion from a set of premises if and only if there is a derivation using the two finger method.

Theorem: Propositional resolution using the two-finger method always terminates.

Proof: There are only finitely many clauses that can be constructed from a finite set of logical constants.

# Decidability of Propositional Entailment

Propositional resolution is a decision procedure for Propositional Logic.

Logical entailment for Propositional Logic is decidable.

Sadly, the problem in general is NP-complete.

# Horn Clauses and Horn Chains

A *Horn clause* is a clause containing at most one positive literal.

Example: $\{R, \sim P, \sim Q\}$
Example: $\{\sim P, \sim Q, \sim R\}$
Example: $\langle P \rangle$

Non-Example: $\{Q, R, \sim P\}$

NB: Every Horn clause can be written as a "rule".

$$\{\sim P, \sim Q, R\} \quad \rightarrow \quad P \cdot Q \Rightarrow R$$

# Complexity

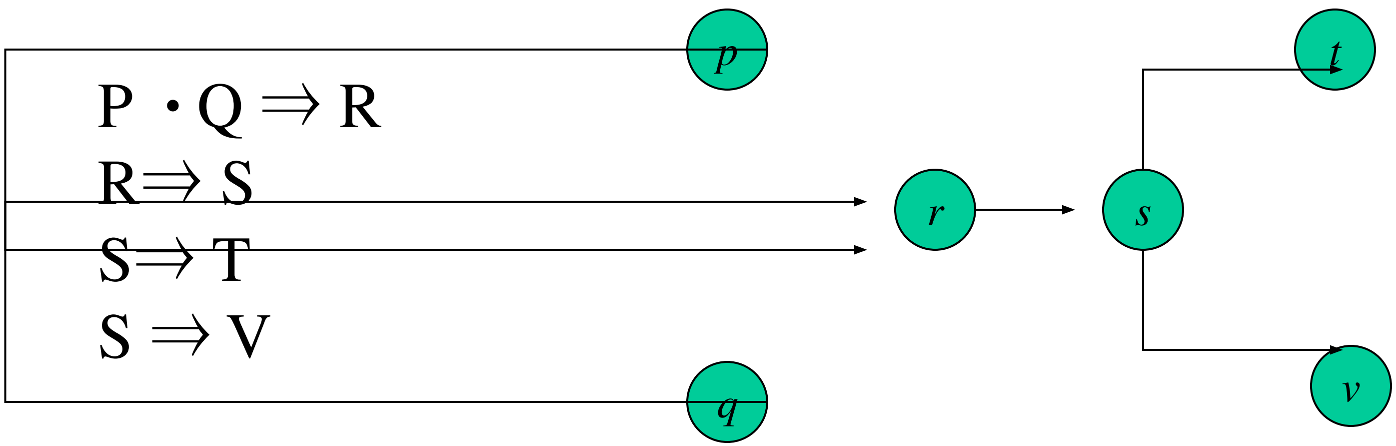
Good news: When a set of propositional sentences is Horn, satisfiability and, consequently, logical entailment can be decided in time linear in the size of the sentence set.

$P \cdot Q \Rightarrow R$

$R \Rightarrow S$

$S \Rightarrow T$

$S \Rightarrow V$




Does {P, Q} |= V?