Propositional Reasoning

CS 440 / ECE 448 Introduction to Artificial Intelligence

Instructor: Eyal Amir
Grad TAs: Wen Pu, Yonatan Bisk
Undergrad TAs: Sam Johnson, Nikhil Johri

Spring 2010
Outline: Propositional Logic

- Basics of Propositional Logic Review
  - Semantics and Deduction
- Generate and Test Review
  - Checking Satisfiability (SAT) using DPLL
- Syntactic Manipulation:
  - Entailment and Refutation using Resolution
Propositional Logic

- Semantics
  - Truth assignments that satisfy KB/formula

- Semantics VS. Deduction

\[ \alpha \models \beta \]
\[ \alpha \vdash \beta \]

- A deductive system or an inference algorithm is
  - Sound
    - If it derives only entailed sentences
  - Complete
    - If it can derive any sentence that is entailed
  - Decidable
    - If it terminates in finite steps

\models \text{ and } \vdash \text{ are equivalent if deduction is sound and complete}
More Notations

- **Interpretations** - models if “True”
- **Axioms** - formulae that are “assumed”
- **Signature** - the symbols used by a KB
- **Theory**
  - KB (a set of axioms)
  - The complete set of sentences entailed by the axioms
- **Sentence** = **Formula** (in prop. logic)
- **M[p]** - value of symbol p in model M
- **Clauses**: \{x_1, x_2, x_3, \ldots\} or \(x_1 \lor x_2 \lor x_3 \lor \ldots\)
Outline: Propositional Logic

- Basics of Propositional Logic Review
  - Semantics and Deduction

- Generate and Test Review
  - Checking Satisfiability (SAT) using DPLL

- Syntactic Manipulation:
  - Entailment and Refutation using Resolution
SAT Application: Hardware Verification

\[ f_5(x_1, x_2, x_3) = f_3 \land f_4 = \neg f_1 \land (f_2 \lor x_3) = \neg(x_1 \land x_2) \land (\neg x_2 \lor x_3) \]

\[ M[x_1] = False, \quad M[x_2] = False, \quad M[x_3] = False \]
Model Checking

KB in CNF, and we denote $a = True$

Removing clauses with $a$ positive from KB gives an equivalent theory.

Removing negative instances of $a$ from KB gives an equivalent theory.

$$KB \quad a \lor b \lor c$$
$$\neg a \lor b \lor d$$

Observe
Model Checking

KB in CNF, and we denote $a = True$

Removing clauses with $a$ positive from KB gives an equivalent theory.

Removing negative instances of $a$ from KB gives an equivalent theory.

$$KB \quad a \lor b \lor c$$
$$\neg a \lor b \lor d$$

Observe $a$
DPLL: Propagating a Truth-Value

- Model Checking
- KB in CNF, and we denote \( a = True \)
- Removing clauses with \( a \) positive from KB gives an equivalent theory.
- Removing negative instances of \( a \) from KB gives an equivalent theory.

\[
\begin{align*}
\text{KB} & \quad a \lor b \lor c \\
\Rightarrow & \quad a \lor b \lor d \\
\text{Observe} & \quad a
\end{align*}
\]
DPLL: Search Heuristics

- Early termination: look only for one model
- Unit propagation: clauses with only one symbol
- Pure literal elimination (obsolete)

DPLL is sound and complete
DPLL: Search Heuristics

- Early termination: look only for one model
- Unit propagation: clauses with only one symbol
- Pure literal elimination (obsolete)

DPLL is sound and complete
DPLL Example

Clauses

1. \(a \lor b\)
2. \(\neg a \lor \neg b\)
3. \(a \lor \neg c\)
4. \(c \lor d \lor e\)
5. \(d \lor \neg e\)
6. \(\neg d \lor \neg f\)
7. \(f \lor e\)
8. \(\neg f \lor \neg e\)

Steps

- \(a = F\) (b is pure)
- \(a = F, b = T\) (3 is unit)
- \(a = F, b = T, c = F\)
- \(a = F, b = T, c = F, d = F\) (4 is unit)
- \(a = F, b = T, c = F, d = F, e = T\) (unsatisfied, early termination, backtrace)
- \(a = F, b = T, c = F, d = T\) (6 is unit)
- \(a = F, b = T, c = F, d = T, f = F\) (e is pure)
- \(a = F, b = T, c = F, d = T, f = F, e = T\) (SAT!)
DPLL Example

Steps

Clauses

1. $a \lor b$
2. $\neg a \lor \neg b$
3. $a \lor \neg c$
4. $c \lor d \lor e$
5. $d \lor \neg e$
6. $\neg d \lor \neg f$
7. $f \lor e$
8. $\neg f \lor \neg e$

- $a = F$ (b is pure)
  - $a = F, b = T$ (3 is unit)
  - $a = F, b = T, c = F$
  - $a = F, b = T, c = F, d = F$ (4 is unit)
  - $a = F, b = T, c = F, d = F, e = T$
  (unsatisfied, early termination, backtrace)
  - $a = F, b = T, c = F, d = T$ (6 is unit)
  - $a = F, b = T, c = F, d = T, f = F$ (e is pure)
  - $a = F, b = T, c = F, d = T, f = F, e = T$ (SAT!)
DPLL Example

Clauses

1. \(a \lor b\)
2. \(\neg a \lor \neg b\)
3. \(a \lor \neg c\)
4. \(c \lor d \lor e\)
5. \(d \lor \neg e\)
6. \(\neg d \lor \neg f\)
7. \(f \lor e\)
8. \(\neg f \lor \neg e\)

Steps

- \(a = F\) (\(b\) is pure)
- \(a = F, b = T\) (3 is unit)
  - \(a = F, b = T, c = F\)
  - \(a = F, b = T, c = F, d = F\) (4 is unit)
  - \(a = F, b = T, c = F, d = F, e = T\) (unsatisfied, early termination, backtrace)
- \(a = F, b = T, c = F, d = T\) (6 is unit)
  - \(a = F, b = T, c = F, d = T, f = F\) (e is pure)
  - \(a = F, b = T, c = F, d = T, f = F, e = T\) (SAT!)
DPLL Example

Steps

Clauses

1. $a \vee b$
2. $\neg a \vee \neg b$
3. $a \vee \neg c$
4. $\neg c \vee d \vee e$
5. $d \vee \neg e$
6. $\neg d \vee \neg f$
7. $f \vee e$
8. $\neg f \vee \neg e$

$\triangleright$ $a = F$ (b is pure)
$\triangleright$ $a = F, b = T$ (3 is unit)
$\triangleright$ $a = F, b = T, c = F$
$\triangleright$ $a = F, b = T, c = F, d = F$ (4 is unit)
$\triangleright$ $a = F, b = T, c = F, d = F, e = T$

(unsatisfied, early termination, backtrace)
$\triangleright$ $a = F, b = T, c = F, d = T$ (6 is unit)
$\triangleright$ $a = F, b = T, c = F, d = T, f = F$ (e is pure)
$\triangleright$ $a = F, b = T, c = F, d = T, f = F, e = T$ (SAT!)
DPLL Example

Clauses

- 1. \( a \lor b \)
- 2. \( \neg a \lor \neg b \)
- 3. \( a \lor \neg c \)
- 4. \( \neg c \lor d \lor e \)
- 5. \( d \lor \neg e \)
- 6. \( \neg d \lor \neg f \)
- 7. \( f \lor e \)
- 8. \( \neg f \lor \neg e \)

Steps

- \( a = F \) (\( b \) is pure)
- \( a = F, b = T \) (3 is unit)
- \( a = F, b = T, c = F \)
- \( a = F, b = T, c = F, d = F \) (4 is unit)
- \( a = F, b = T, c = F, d = F, e = T \)
  (unsatisfied, early termination, backtrace)
- \( a = F, b = T, c = F, d = T \) (6 is unit)
- \( a = F, b = T, c = F, d = T, f = F \) (\( e \) is pure)
- \( a = F, b = T, c = F, d = T, f = F, e = T \) (SAT!)
DPLL Example

Steps

Clauses

1. \(a \lor b\)
2. \(\neg a \lor \neg b\)
3. \(a \lor \neg c\)
4. \(c \lor d \lor \neg e\)
5. \(d \lor \neg e\)
6. \(\neg d \lor \neg f\)
7. \(f \lor e\)
8. \(\neg f \lor \neg e\)

- \(a = F\) (\(b\) is pure)
- \(a = F, b = T\) (3 is unit)
- \(a = F, b = T, c = F\)
- \(a = F, b = T, c = F, d = F\) (4 is unit)
- \(a = F, b = T, c = F, d = F, e = T\) (unsatisfied, early termination, backtrace)
- \(a = F, b = T, c = F, d = T\) (6 is unit)
- \(a = F, b = T, c = F, d = T, f = F\) (\(e\) is pure)
- \(a = F, b = T, c = F, d = T, f = F, e = T\) (SAT!)
DPLL Example

**Steps**

- **a = F** *(b is pure)*
- **a = F, b = T** *(3 is unit)*
- **a = F, b = T, c = F***
- **a = F, b = T, c = F, d = F** *(4 is unit)*
- **a = F, b = T, c = F, d = F, e = T** *(unsatisfied, early termination, backtrace)*
- **a = F, b = T, c = F, d = T** *(6 is unit)*
- **a = F, b = T, c = F, d = T, f = F** *(e is pure)*
- **a = F, b = T, c = F, d = T, f = F, e = T** *(SAT!)*

**Clauses**

- **1.** \( a \lor b \)
- **2.** \( \neg a \lor \neg b \)
- **3.** \( a \lor \neg c \)
- **4.** \( \neg c \lor d \lor e \)
- **5.** \( d \lor \neg e \)
- **6.** \( \neg d \lor \neg f \)
- **7.** \( f \lor e \)
- **8.** \( \neg f \lor \neg e \)
DPLL Example

Steps

Clauses

1. \( a \lor b \)
2. \( \neg a \lor \neg b \)
3. \( a \lor \neg c \)
4. \( c \lor d \lor \neg e \)
5. \( d \lor \neg e \)
6. \( \neg d \lor \neg f \)
7. \( f \lor e \)
8. \( \neg f \lor \neg e \)

- \( a = F \) (\( b \) is pure)
- \( a = F, b = T \) (3 is unit)
- \( a = F, b = T, c = F \)
- \( a = F, b = T, c = F, d = F \) (4 is unit)
- \( a = F, b = T, c = F, d = F, e = T \) (unsatisfied, early termination, backtrace)
- \( a = F, b = T, c = F, d = T \) (6 is unit)
- \( a = F, b = T, c = F, d = T, f = F \) (\( e \) is pure)
- \( a = F, b = T, c = F, d = T, f = F, e = T \) (SAT!)
DPLL Example

- **Clauses**
  - 1. \( a \lor b \)
  - 2. \( \neg a \lor \neg b \)
  - 3. \( a \lor \neg c \)
  - 4. \( c \lor d \lor e \)
  - 5. \( d \lor \neg e \)
  - 6. \( d \lor \neg f \)
  - 7. \( f \lor e \)
  - 8. \( f \lor \neg e \)

- **Steps**
  - \( a = F \) (\( b \) is pure)
  - \( a = F, b = T \) (3 is unit)
  - \( a = F, b = T, c = F \)
  - \( a = F, b = T, c = F, d = F \) (4 is unit)
  - \( a = F, b = T, c = F, d = F, e = T \)
    (unsatisfied, early termination, backtrace)
  - \( a = F, b = T, c = F, d = T \) (6 is unit)
  - \( a = F, b = T, c = F, d = T, f = F \) (e is pure)
  - \( a = F, b = T, c = F, d = T, f = F, e = T \) (SAT!)
DPLL Example

Clauses

1. \(a \lor b\)
2. \(\neg a \lor \neg b\)
3. \(a \lor c\)
4. \(c \lor d \lor e\)
5. \(d \lor \neg e\)
6. \(d \lor \neg f\)
7. \(f \lor e\)
8. \(f \lor \neg e\)

Steps

- \(a = F\) (\(b\) is pure)
- \(a = F, b = T\) (3 is unit)
- \(a = F, b = T, c = F\)
- \(a = F, b = T, c = F, d = F\) (4 is unit)
- \(a = F, b = T, c = F, d = F, e = T\)
- \(a = F, b = T, c = F, d = T, f = F\) (\(e\) is pure)
- \(a = F, b = T, c = F, d = T, f = F, e = T\) (SAT!)
Outline: Propositional Logic

- Basics of Propositional Logic Review
  - Semantics and Deduction
- Generate and Test Review
  - Checking Satisfiability (SAT) using DPLL
- Syntactic Manipulation:
  - Entailment and Refutation using Resolution
Resolution Theorem Proving

Resolution theorem (converse of deduction theorem):

\[ KB \models Q \iff KB \land \neg Q \models False \]

Given query \( Q \) and a database of facts \( KB \)
- Add \( \neg Q \) to \( KB \)
- Convert \( KB \) into CNF
- Run theorem prover, if we prove contradiction, return True, otherwise return False.
Resolution Theorem Proving

- Resolution theorem (converse of deduction theorem):

\[ KB \models Q \iff KB \land \neg Q \models \text{False} \]

- Given query \( Q \) and a database of facts \( KB \)
  - Add \( \neg Q \) to \( KB \)
  - Convert \( KB \) into CNF
  - Run theorem prover, if we prove contradiction, return True, otherwise return False.
Resolution Rule

Resolution rule

\[ x \lor a_1 \lor \cdots \lor a_m, \neg x \lor b_1 \lor \cdots \lor b_n \]

\[ a_1 \lor \cdots \lor a_m \lor b_1 \lor \cdots \lor b_n \]

Intuition:

\[ \neg x \rightarrow \alpha, \ x \rightarrow \beta \models \alpha \lor \beta \]
Resolution Rule

- Resolution rule

\[
\begin{array}{c}
\neg x \lor a_1 \lor \cdots \lor a_m, \\
\neg x \lor b_1 \lor \cdots \lor b_n
\end{array}
\]

\[
\begin{array}{c}
a_1 \lor \cdots \lor a_m \lor b_1 \lor \cdots \lor b_n
\end{array}
\]

- Intuition:

\[
\neg x \rightarrow \alpha, \ x \rightarrow \beta \models \alpha \lor \beta
\]
Propositional Resolution

Resolution Algorithm

1. While there are unresolved $C_1$, $C_2$
   
   1. Select $C_1$, $C_2$ in KB
   
   2. If $C_1$, $C_2$ are resolvable, resolve them into new clause $C_3$ and add it to KB
   
   3. If $C_3 = \emptyset$ we got a contradiction

2. STOP

Theorem Proving VS. Model Checking (Difference?)

$C_1 : p_1 \lor C'_1$

$C_2 : \neg p_1 \lor C'_2$

$\therefore C_3 : C'_1 \lor C'_2$
Propositional Resolution

- Resolution Algorithm
  1. While there are unresolved $C_1$, $C_2$
     1. Select $C_1$, $C_2$ in KB
     2. If $C_1$, $C_2$ are resolvable, resolve them into new clause $C_3$ and add it to KB
     3. If $C_3 = \{\}$ we got a contradiction
  2. STOP

- Theorem Proving VS. Model Checking (Difference?)

$$C_1 : p_1 \lor C'_1$$
$$C_2 : \neg p_1 \lor C'_2$$

$$\frac{C_1 \land C_2}{C_3 : C'_1 \lor C'_2}$$
Resolution Example: Constructive Dilemma

- Problem: “Constructive Dilemma”
  - KB: $p \rightarrow q$, $r \rightarrow s$, $p \lor r$
  - Theorem to prove: $q \lor s$

\[
(p \rightarrow q) \land (r \rightarrow s) \land (p \lor r) \models q \lor s
\]

- Convert to CNF
  - KB: $\neg p \lor q$, $\neg r \lor s$, $p \lor r$
  - Negated query: $\neg q$, $\neg s$
Resolution Example: Constructive Dilemma

Problem: “Constructive Dilemma”
- KB: $p \rightarrow q$, $r \rightarrow s$, $p \lor r$
- Theorem to prove: $q \lor s$

$$(p \rightarrow q) \land (r \rightarrow s) \land (p \lor r) \models q \lor s$$

Convert to CNF
- KB: $\neg p \lor q$, $\neg r \lor s$, $p \lor r$
- Negated query: $\neg q$, $\neg s$
Resolution Example (cont.)

Formulas:
1. $\neg p \lor q$
2. $\neg r \lor s$
3. $p \lor r$
4. $\neg q$
5. $\neg s$

Steps
6. $\neg p$ (1, 4)
7. $r$ (3, 6)
8. $s$ (2, 7)
9. (empty) (5, 8) Contradiction!

Conclude: $(p \rightarrow q) \land (r \rightarrow s) \land (p \lor r) \models_{\text{resolution}} q \lor s$
Resolution Example (cont.)

Formulas:

1. ¬p ∨ q
2. ¬r ∨ s
3. p ∨ r
4. ¬q
5. ¬s

Steps

6. ¬p (1, 4)
7. r (3, 6)
8. s (2, 7)
9. (empty) (5, 8) \textbf{Contradiction!}

Conclude: \((p \rightarrow q) \land (r \rightarrow s) \land (p \lor r) \vdash \text{resolution} \ q \lor s\)
Resolution Example (cont.)

Formulas:

1. \( \neg p \lor q \)
2. \( \neg r \lor s \)
3. \( p \lor r \)
4. \( \neg q \)
5. \( \neg s \)

Steps

6. \( \neg p \) (1, 4)
7. \( r \) (3, 6)
8. \( s \) (2, 7)
9. (empty) (5, 8) Contradiction!

Conclude: \( (p \rightarrow q) \land (r \rightarrow s) \land (p \lor r) \models_{\text{resolution}} q \lor s \)
Resolution Example (cont.)

- **Formulas:**
  1. \( \neg p \lor q \)
  2. \( \neg r \lor s \)
  3. \( p \lor r \)
  4. \( \neg q \)
  5. \( \neg s \)

- **Steps**
  6. \( \neg p \ (1, 4) \)
  7. \( r \ (3, 6) \)
  8. \( s \ (2, 7) \)
  9. (empty) (5, 8) **Contradiction!**

Conclude: \((p \rightarrow q) \land (r \rightarrow s) \land (p \lor r) \vdash_{\text{resolution}} q \lor s\)
Resolution Example (cont.)

- Formulas:
  1. \( \neg p \lor q \)
  2. \( \neg r \lor s \)
  3. \( p \lor r \)
  4. \( \neg q \)
  5. \( \neg s \)

- Steps
  6. \( \neg p \ (1, 4) \)
  7. \( r \ (3, 6) \)
  8. \( s \ (2, 7) \)
  9. (empty) \( (5, 8) \) Contradiction!

Conclude: \( (p \rightarrow q) \land (r \rightarrow s) \land (p \lor r) \models \text{resolution} \ q \lor s \)
Resolution Example (cont.)

Formulas:

1. \(\neg p \lor q\)
2. \(\neg r \lor s\)
3. \(p \lor r\)
4. \(\neg q\)
5. \(\neg s\)

Steps

6. \(\neg p\) (1, 4)
7. \(r\) (3, 6)
8. \(s\) (2, 7)
9. (empty) (5, 8) Contradiction!

Conclude: \((p \rightarrow q) \land (r \rightarrow s) \land (p \lor r)\) \(\vdash_{\text{resolution}} q \lor s\)
Properties of Resolution

- **Theorem: Resolution is sound**
  - Resolving clauses in KB generates valid consequences of KB

- **Theorem: Resolution is refutation complete**
  - Resolution of KB with $\neg Q$ yields the empty clause iff KB $\Vdash Q$

\[(p \rightarrow q) \land (r \rightarrow s) \land (p \lor r) \Vdash q \lor s\]
Properties of Resolution

- **Theorem: Resolution is sound**
  - Resolving clauses in KB generates valid consequences of KB

- **Theorem: Resolution is refutation complete**
  - Resolution of KB with \( \neg Q \) yields the empty clause iff KB \( \models Q \)

\[(p \rightarrow q) \land (r \rightarrow s) \land (p \lor r) \models q \lor s\]
Properties of Resolution

- **Will resolution only in KB generate Q?**
- Resolution does not always generate Q

\[
KB = \{\{a, b\}, \{\neg a, b\}, \{b, c\}\}
\]
\[
Q = b \lor \neg c = \{b, \neg c\}
\]

- Theorem: Resolution always generates a clause that subsumes Q iff KB ⊨ Q.
  - Example: Resolving KB generates b
Properties of Resolution

- Will resolution only in KB generate $Q$?
- Resolution does not always generate $Q$

$$KB = \{\{a, b\}, \{\neg a, b\}, \{b, c\}\}$$
$$Q = b \lor \neg c = \{b, \neg c\}$$

- Theorem: Resolution always generates a clause that subsumes $Q$ iff $KB \models Q$.
  - Example: Resolving KB generates $b$
Exercise

- Use resolution refutation prove
  - KB: $P \rightarrow Q, \neg P \rightarrow R$
  - Query: $\neg Q \rightarrow \neg R$
Solution

- Convert to CNF

  KB: \( \neg P \lor Q, \ P \lor R \), Negated query: \( \neg Q, \ R \)

- Formulas:
  - 1. \( \neg P \lor Q \)
  - 2. \( P \lor R \)
  - 3. \( \neg Q \)
  - 4. \( R \)
  - 5. \( P \lor Q \)

- Steps:
  - 5. \( \neg P \ (1,3) \)
  - 6. \( R \ (2, 5) \)
  - 7. \( Q \lor R \ (1, 2) \)
  - Terminate without producing empty clause!

  \[ \text{KB} \not\models \neg Q \rightarrow \neg R \]
Solution

- Convert to CNF
  KB: \( \neg P \lor Q, P \lor R \), Negated query: \( \neg Q, R \)

- Formulas:
  1. \( \neg P \lor Q \)
  2. \( P \lor R \)
  3. \( \neg Q \)
  4. \( R \)
  5. \( P \lor Q \)

- Steps:
  5. \( \neg P \) (1,3)
  6. \( R \) (2, 5)
  7. \( Q \lor R \) (1, 2)

- Terminate without producing empty clause!

\[ KB \not\models \neg Q \rightarrow \neg R \]
Solution

- Convert to CNF
  
  KB: $\neg P \lor Q$, $P \lor R$, Negated query: $\neg Q$, $R$

- Formulas:
  1. $\neg P \lor Q$
  2. $P \lor R$
  3. $\neg Q$
  4. $R$
  5. $P \lor Q$

- Steps:
  5. $\neg P$ (1,3)
  6. $R$ (2, 5)
  7. $Q \lor R$ (1, 2)

  Terminate without producing empty clause!

  $\text{KB} \not\models \neg Q \rightarrow \neg R$
Solution

- Convert to CNF
  - KB: \( \neg P \lor Q, \ P \lor R \), Negated query: \( \neg Q, \ R \)

- Formulas:
  1. \( \neg P \lor Q \)
  2. \( P \lor R \)
  3. \( \neg Q \)
  4. \( R \)
  5. \( P \lor Q \)

- Steps:
  5. \( \neg P \) (1,3)
  6. \( R \) (2, 5)
  7. \( Q \lor R \) (1, 2)

- Terminate without producing empty clause!

\[
\text{KB} \not\models \neg Q \rightarrow \neg R
\]
Solution

- Convert to CNF
  - KB: \( \neg P \lor Q, P \lor R \), Negated query: \( \neg Q, R \)

- Formulas:
  1. \( \neg P \lor Q \)
  2. \( P \lor R \)
  3. \( \neg Q \)
  4. \( R \)
  5. \( P \lor Q \)

- Steps:
  5. \( \neg P \) (1,3)
  6. \( R \) (2, 5)
  7. \( Q \lor R \) (1, 2)
  Terminate without producing empty clause!

\[ \text{KB} \nvDash \neg Q \rightarrow \neg R \]
Resolution Strategies

- **Unit Preference**: resolve unit clauses (one literal clause) first
  - Generate shorter clause (our target is a zero length clause, i.e. contradiction)

- **Set of Support**: Choose a resolution involving the negated goal or any clause derived from the negated goal
  - Target is to produce contradiction from negated query, so these are more relevant
  - If contradiction exists, we can find one using this strategy

- **Other improvements**
  - Remove subsumed clauses
    - \{p\} subsumes \{p, q\}
    - \{p, q\} subsumes \{p, q, r\}
    - \{\neg p\} does not subsume \{p, q\}
  - Contract same literals
    - \{p, p, q\} becomes \{p, q\}
Resolution Strategies

- **Unit Preference**: resolve unit clauses (one literal clause) first
  - Generate shorter clause (our target is a zero length clause, i.e. contradiction)

- **Set of Support**: Choose a resolution involving the negated goal or any clause derived from the negated goal
  - Target is to produce contradiction from negated query, so these are more relevant
  - If contradiction exists, we can find one using this strategy

- **Other improvements**
  - Remove subsumed clauses
    - \{p\} subsumes \{p, q\}
    - \{p, q\} subsumes \{p, q, r\}
    - \{¬p\} does not subsume \{p, q\}
  - Contract same literals
    - \{p, p, q\} becomes \{p, q\}
Resolution Strategies

- **Unit Preference**: resolve unit clauses (one literal clause) first
  - Generate shorter clause (our target is a zero length clause, i.e. contradiction)

- **Set of Support**: Choose a resolution involving the negated goal or any clause derived from the negated goal
  - Target is to produce contradiction from negated query, so these are more relevant
  - If contradiction exists, we can find one using this strategy

- Other improvements
  - **Remove subsumed clauses**
    - \{p\} subsumes \{p, q\}
    - \{p, q\} subsumes \{p, q, r\}
    - \{¬p\} does not subsume \{p, q\}
  - **Contract same literals**
    - \{p, p, q\} becomes \{p, q\}
Resolution vs. SAT

- Theorem proving vs Model checking
- SAT solvers can find models
- Resolution sometimes better at finding contradictions
- With resolution it is easier to explain and provide a proof
Resolution in Prop. Logc and FOL

- **Similarities**
  - Both use clausal form

- **Differences**
  - Unification is needed in FOL
  - Always terminates in prop. logic, but does not guarantee to terminate in FOL
Summary

- SAT checking using DPLL (instantiate, propagate, backtrack)
- Entailment/SAT checking using resolution (create more and more clauses until KB is saturated)
- Formal verification uses mainly SAT checking such as DPLL, but also sometimes resolution.