Resolution Theorem Proving

\[ \text{KB} \models Q \leftarrow \text{Query in prop. logic.} \]

\[ \text{prop. formula} \rightarrow \text{KBA}^{-1}Q \text{ has a model?} \]

\[ \text{in CNF=} \]

\[ \begin{align*}
\text{Yes} & \rightarrow \text{KB} \not\models Q \\
\text{No} & \rightarrow \text{KB} \models Q
\end{align*} \]

we saw \[ \rightarrow \text{DPLL finds a model of KB} \]

Today \[ \rightarrow \text{Find proofs of KB} \models Q \]

Def: A proof of Q from KB is a sequence of logical inference steps/rules applied to our starting knowledge.
\[ M \models \text{KB} \\Longleftrightarrow \]

Then we can extend \( M \) to

If \( \text{KB} \) has a model \( M \), \( \text{KB} \cup \Gamma \) has no model.

Instance of \( \neg \text{KB} \).

set of clauses that have no

on \( p \) in \( \text{KB} \). Call the result

perform all the resolutions possible

Take \( p \in \text{KB} \).

We prove that for \( \Gamma \),

\( \text{KB} \cup \Gamma \models \text{false} \).

If \( \text{KB} \cup \Gamma \not\models \text{false} \) then

Assume that we proved

Induction Step:
### Resolution Algorithm

1. **Input:** A set of clauses \( \{ C_1, C_2, \ldots, C_m \} \) in \( \text{CNF} \) form.

2. **Resolution Step:** Choose a pair of clauses \( C_i \) and \( C_j \) where \( C_i \cap C_j \neq \emptyset \) and \( \exists \text{ a literal } l = \overline{l} \text{ such that } C_i \cap C_j = \{ l, \overline{l} \} \).

3. **Combine:** Derive the new clause \( C_{i,j} \) using the resolution rule:
   \[
   C_{i,j} = C_i \lor C_j - \{ l, \overline{l} \}
   \]
   if \( l \) is a literal in \( C_i \), then add \( C_{i,j} \) to \( \{ C_1, C_2, \ldots, C_m \} \).

4. **Output:** The set of clauses after the resolution process.

   - **Resolution Rule:**
     \[
     \text{If } C_i \cap C_j \neq \emptyset \text{ and } \exists \text{ a literal } l = \overline{l} \text{ such that } C_i \leftrightarrow \{ l, \overline{l} \} \implies C_{i,j} = C_i \lor C_j - \{ l, \overline{l} \}
     \]

   - **Termination:** The derivation is complete if there are no more pairs of clauses that can be resolved.

### Satisfiable Form of Clauses

- **TRUE:**
  \[
  \begin{align*}
  p &\lor \overline{p} \\
  \end{align*}
  \]
- **FALSE:**
  \[
  \begin{align*}
  \overline{p} &\lor \overline{p} \\
  \end{align*}
  \]
Proof: 

\[ \text{If } \models L(q) \text{ then } KBA \vdash q \text{ is } \text{true.} \]

Given: 

\[ KBA \vdash q \implies \text{false.} \]

Let \( n = 1 \). We prove that 

\[ KBA \vdash q \implies \text{false.} \]

Vocabulary (set of prop. symbols) of \( KBA \vdash q \) (assume \( L(KB) = \{ p \} \))

\[ KB \vdash q = KBA \vdash q \implies \text{false.} \]

\[ \text{If an interpretation } I \]

\[ \models p, \text{ then } I \not\models KBA \vdash q \]

\[ \text{there is a clause } C \text{ in } 
   \text{KBA} \vdash q \text{ s.t. } C = \{ \neg p \}. \]

Similarly for \( I \models \neg p \), there is 

\[ C' = \{ p \} \text{ in } KBA \vdash q \]

\[ \text{resolution will resolve } 
   C, C' \text{ and give } \{ \}. \]

---

Take \( C_1, C_2 \) the first in resolution alg.: s.t. \( M \models C_1 \lor C_2 \)

but \( M \not\models C_3 \) \( (C_3 \text{ resolved of } C_1, C_2) \).

\[ C_1 = p \lor C_1', \quad C_2 = \neg p \lor C_2' \]

for some \( C_1', C_2' \) clauses.

\[ M \models C_1 \implies M \not\models p \text{ or } M \models C_1'. \]

\[ M \models C_2 \implies M \not\models \neg p \text{ or } M \models C_2'. \]

\[ M \not\models p \text{ or } M \not\models \neg p. \]

\[ \implies M \models C_1' \lor C_2'. \]

\[ \implies M \models C_3. \]