CS344: Introduction to Artificial Intelligence
(associated lab: CS386)

Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay

Lecture–30, 31: Predicate Calculus; Interpretation
26th and 29th March, 2012

(Course seminars and discussion on g(n)=g*(n) on 27th)
Himalayan Club example

- Introduction through an example (Zohar Manna, 1974):
  - Problem: A, B and C belong to the Himalayan club. Every member in the club is either a mountain climber or a skier or both. A likes whatever B dislikes and dislikes whatever B likes. A likes rain and snow. No mountain climber likes rain. Every skier likes snow. Is there a member who is a mountain climber and not a skier?

- Given knowledge has:
  - Facts
  - Rules
Let $mc$ denote mountain climber and $sk$ denotes skier. Knowledge representation in the given problem is as follows:

1. $\text{member}(A)$
2. $\text{member}(B)$
3. $\text{member}(C)$
4. $\forall x[\text{member}(x) \rightarrow (mc(x) \lor sk(x))]$
5. $\forall x[mc(x) \rightarrow \sim \text{like}(x, \text{rain})]$
6. $\forall x[sk(x) \rightarrow \text{like}(x, \text{snow})]$
7. $\forall x[\text{like}(B, x) \rightarrow \sim \text{like}(A, x)]$
8. $\forall x[\sim \text{like}(B, x) \rightarrow \text{like}(A, x)]$
9. $\text{like}(A, \text{rain})$
10. $\text{like}(A, \text{snow})$
11. Question: $\exists x[\text{member}(x) \land mc(x) \land \sim sk(x)]$

We have to infer the 11\textsuperscript{th} expression from the given 10.

Done through Resolution Refutation.
Club example: Inferencing

1. \textit{member}(A)
2. \textit{member}(B)
3. \textit{member}(C)
4. \( \forall x[\textit{member}(x) \rightarrow (\textit{mc}(x) \lor \textit{sk}(x))] \)
   - Can be written as
   \( \sim \textit{member}(x) \lor \textit{mc}(x) \lor \textit{sk}(x) \) \( (\textit{mc}(x) \lor \textit{sk}(x)) \)
5. \( \forall x[\textit{sk}(x) \rightarrow \textit{lk}(x, \textit{snow})] \)
   - \( \sim \textit{sk}(x) \lor \textit{lk}(x, \textit{snow}) \)
6. \( \forall x[\textit{mc}(x) \rightarrow \sim \textit{lk}(x, \textit{rain})] \)
   - \( \sim \textit{mc}(x) \lor \sim \textit{lk}(x, \textit{rain}) \)
7. \( \forall x[\textit{like}(A, x) \rightarrow \sim \textit{lk}(B, x)] \)
   - \( \sim \textit{like}(A, x) \lor \sim \textit{lk}(B, x) \)
8. $\forall x[\sim lk(A, x) \rightarrow lk(B, x)]$
   \[ \neg lk(A, x) \lor lk(B, x) \]

9. $lk(A, \text{rain})$

10. $lk(A, \text{snow})$

11. $\exists x[\text{member}(x) \land mc(x) \land \sim sk(x)]$
   \[ \neg \forall x[\sim \text{member}(x) \lor \sim mc(x) \lor sk(x)] \]
Now standardize the variables apart which results in the following

1. \(member(A)\)
2. \(member(B)\)
3. \(member(C)\)
4. \(\sim member(x_1) \lor mc(x_1) \lor sk(x_1)\)
5. \(\sim sk(x_2) \lor lk(x_2, snow)\)
6. \(\sim mc(x_3) \lor \sim lk(x_3, rain)\)
7. \(\sim like(A, x_4) \lor \sim lk(B, x_4)\)
8. \(lk(A, x_5) \lor lk(B, x_5)\)
9. \(lk(A, rain)\)
10. \(lk(A, snow)\)
11. \(\sim member(x_6) \lor \sim mc(x_6) \lor sk(x_6)\)
\(~ \text{like}(A, x_4) \lor \sim \text{lk}(B, x_4)\)
Insight into resolution
Resolution - Refutation

- $\text{man}(x) \rightarrow \text{mortal}(x)$
  - Convert to clausal form
  - $\sim\text{man}(\text{shakespeare}) \lor \text{mortal}(x)$

- Clauses in the knowledge base
  - $\sim\text{man}(\text{shakespeare}) \lor \text{mortal}(x)$
  - $\text{man}(\text{shakespeare})$
  - $\text{mortal}(\text{shakespeare})$
Resolution – Refutation contd

- *Negate the goal*
  - ~man(shakespeare)

- *Get a pair of resolvents*

\[
\neg \text{mortal}(\text{shakespeare}) \quad \neg \text{man}(\text{shakespeare}) \lor \text{mortal}(\text{shakespeare})
\]

\[
\neg \text{man}(\text{shakespeare}) \quad \neg \text{man}(\text{shakespeare})
\]
Resolution Tree

Resolvent 1

Resolvent 2

Resolvent

Resolvent

Resolute
Search in resolution

- Heuristics for Resolution Search
  - Goal Supported Strategy
    - Always start with the negated goal
  - Set of support strategy
    - Always one of the resolvents is the most recently produced resolute
Inferencing in Predicate Calculus

- **Forward chaining**
  - Given P, $P \rightarrow Q$, to infer Q
  - P, match *L.H.S* of
  - Assert Q from *R.H.S*

- **Backward chaining**
  - Q, Match *R.H.S* of $P \rightarrow Q$
  - assert P
  - Check if P exists

- **Resolution – Refutation**
  - Negate goal
  - Convert all pieces of knowledge into clausal form (disjunction of literals)
  - See if contradiction indicated by null clause can be derived
1. \( P \)

2. \( P \rightarrow Q \) converted to \( \sim P \lor Q \)

3. \( \sim Q \)

Draw the resolution tree (actually an inverted tree). Every node is a clausal form and branches are intermediate inference steps.
Theoretical basis of Resolution

- Resolution is proof by contradiction
- \textit{resolvent1 .AND. resolvent2} $\Rightarrow$ \textit{resolute} is a tautology
Tautologiness of Resolution

- Using Semantic Tree

\[(P \lor Q)^{\land} (\neg P \lor Q)\]

\[\neg Q\]

\[P \lor Q\]

\[\neg P \lor Q\]

\[\neg P\]

\[Q\]
Theoretical basis of Resolution (cont ...)

- Monotone Inference
  - Size of Knowledge Base goes on increasing as we proceed with resolution process since intermediate resolvents added to the knowledge base

- Non-monotone Inference
  - Size of Knowledge Base does not increase
  - Human beings use non-monotone inference
Terminology

- Pair of clauses being resolved is called the Resolvents. The resulting clause is called the Resolute.
- Choosing the correct pair of resolvents is a matter of search.