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

Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay

Lecture 17, 18: Predicate Calculus

15th and 16th Feb, 2011

Predicate Calculus: well known examples

- Man is mortal : rule

  $\forall x[man(x) \rightarrow mortal(x)]$

- shakespeare is a man

  man(shakespeare)

- To infer shakespeare is mortal

  mortal(shakespeare)

Inferencing: Forward Chaining

- *man(x)* $\rightarrow$ *mortal(x)*
  - *Dropping the quantifier, implicitly Universal quantification assumed*
  - *man(shakespeare)*
- Goal mortal(shakespeare)
  - Found in one step
  - x = shakespeare, unification

Backward Chaining

- $man(x) \rightarrow mortal(x)$
- Goal mortal(shakespeare)
  - x = shakespeare
  - Travel back over and hit the fact asserted
  - man(shakespeare)

Factors influencing Forward and Backward chaining

- Is the goal precisely known?

- Fan-in and Fan-out of rules.

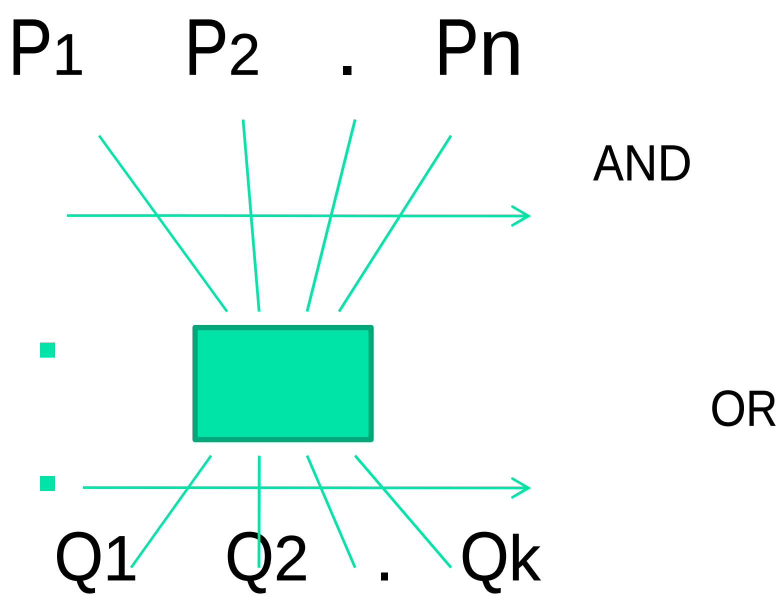
Rule Structure

$R_1 : \quad P_1 \wedge P_2 \wedge P_3 \wedge \ldots\ldots \wedge P_n \rightarrow Q_1$

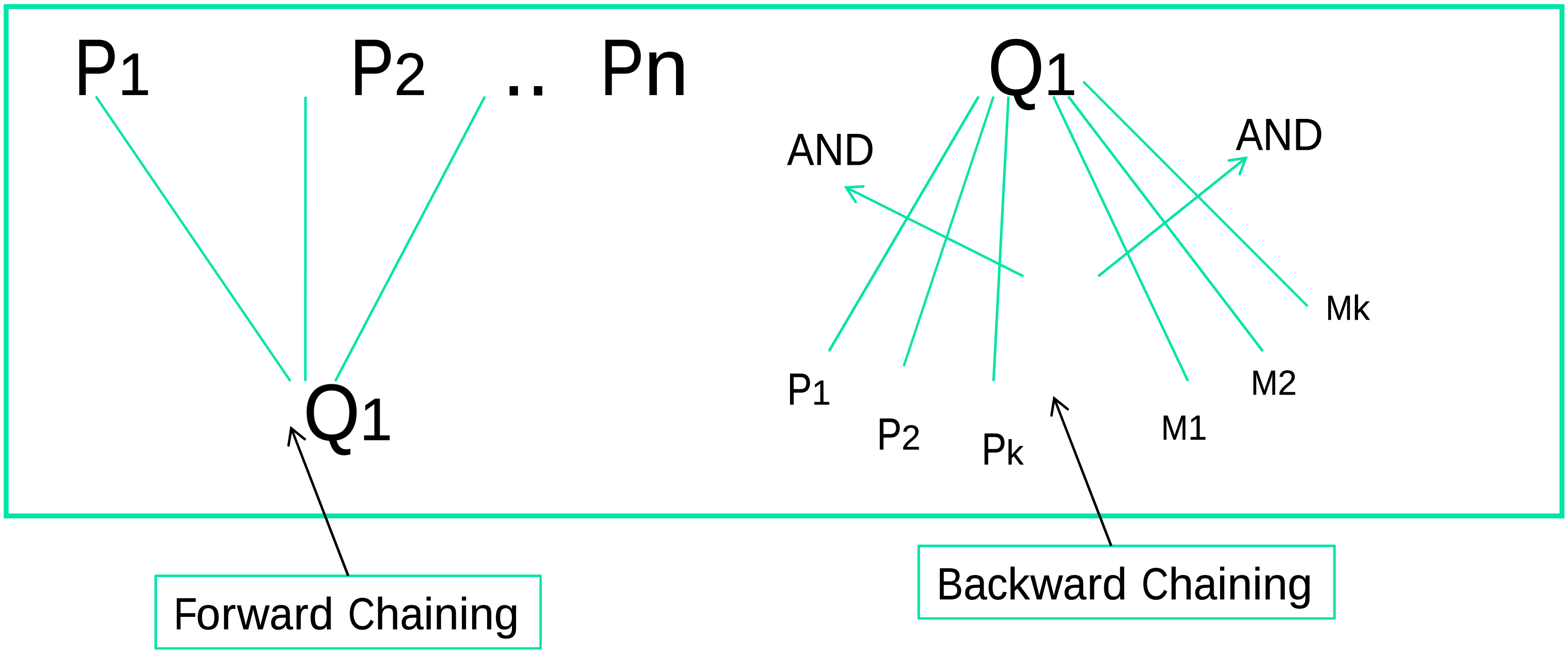
$R_2 : \quad P_1 \wedge P_2 \wedge P_3 \wedge \ldots\ldots \wedge P_n \rightarrow Q_2$

.

$R_k : \quad P_1 \wedge P_2 \wedge P_3 \wedge \ldots\ldots \wedge P_n \rightarrow Q_k$

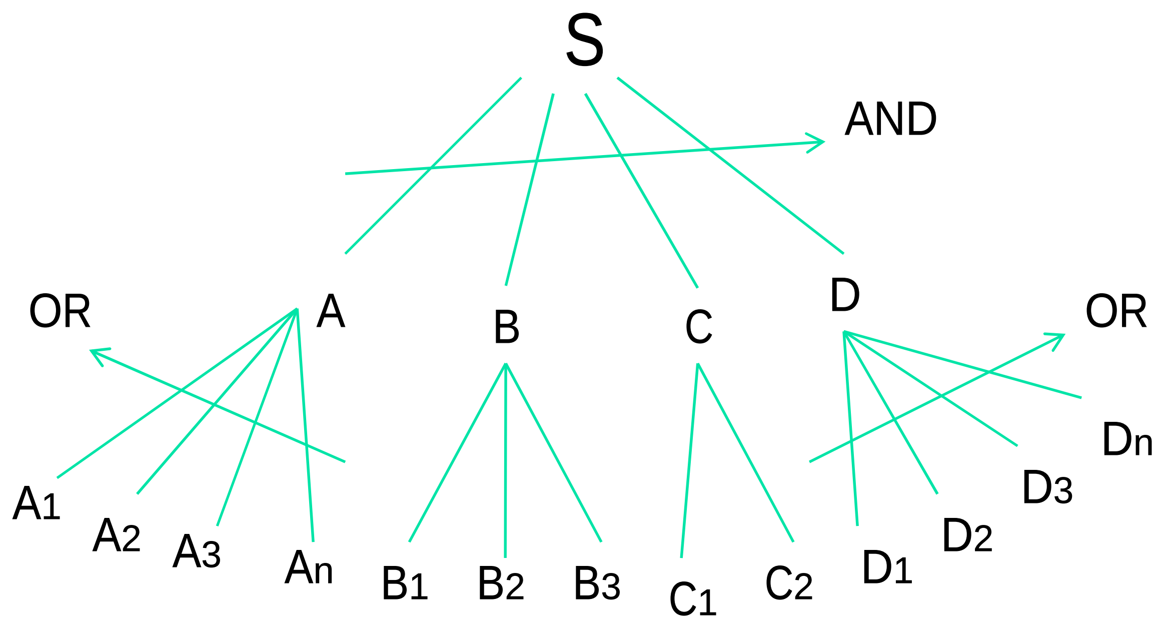


Pictorial Representation of Forward and Backward chaining



- If Fan-out is less Forward chaining is preferable ?

Important Data structure: AND-OR Graph



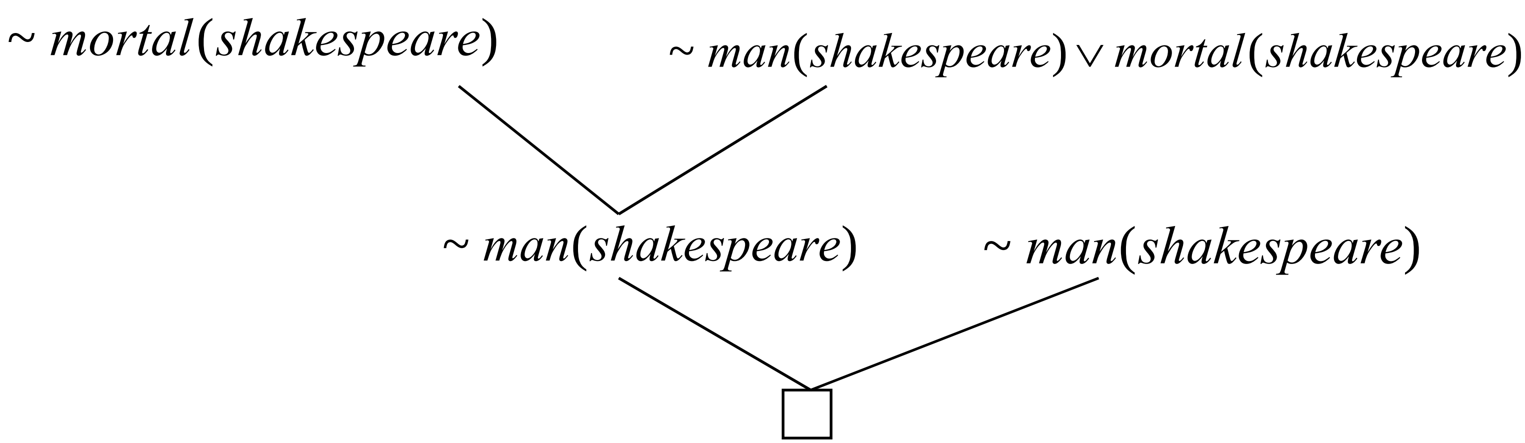
- Structure of AND-OR Graph decides the direction of inferencing.

Resolution - Refutation

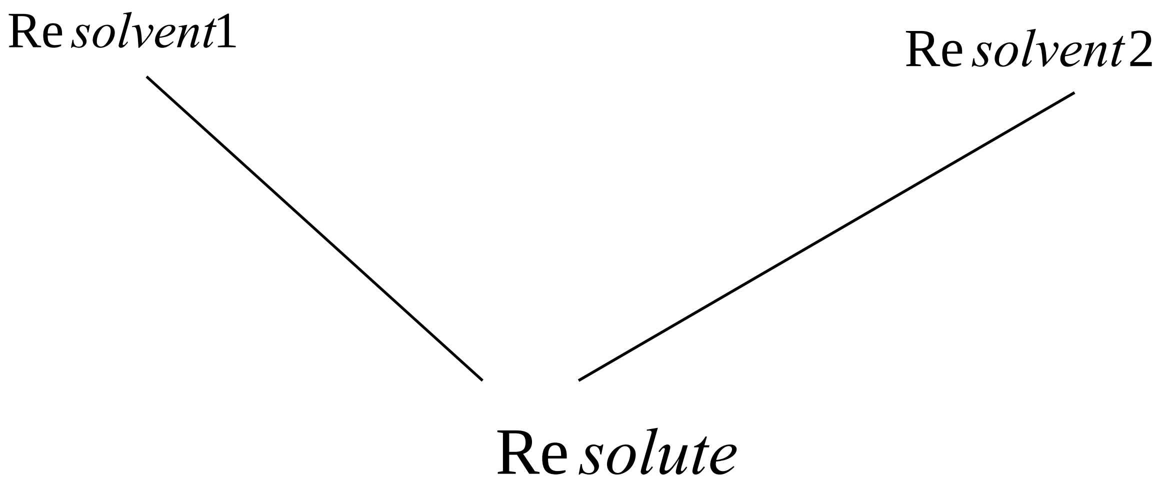
- *man(x) → mortal(x)*
  - *Convert to clausal form*
  - *~man(shakespeare) \/ mortal(x)*
- Clauses in the knowledge base
  - *~man(shakespeare) \/ mortal(x)*
  - *man(shakespeare)*
  - *mortal(shakespeare)*

Resolution – Refutation contd

- *Negate the goal*
  - *~man(shakespeare)*
- Get a pair of resolvents



Resolution Tree



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

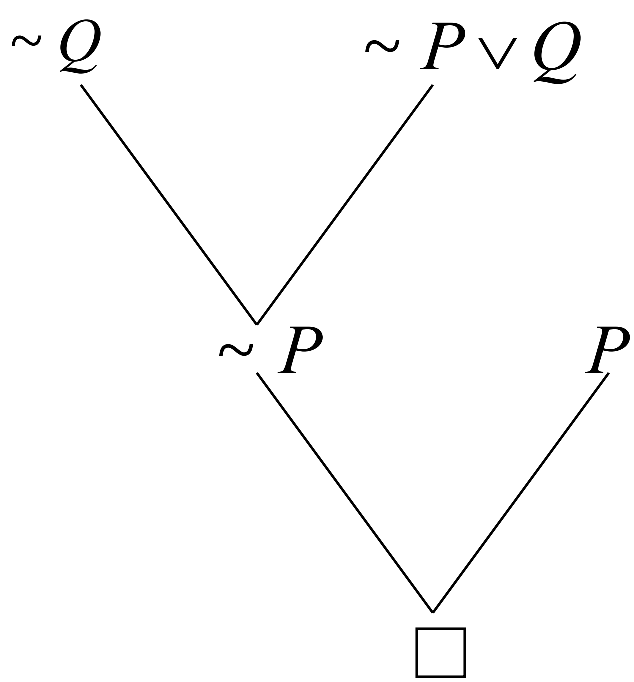
- Forward chaining
  - Given P, $P \rightarrow Q$, to infer Q
  - P, match $L.H.S$ of
  - Assert Q from $R.H.S$
- Backward chaining
  - Q, Match $R.H.S$ of $P \rightarrow Q$
  - assert P
  - Check if P exists
- Resolution – Refutation
  - Negate goal
  - Convert all pieces of knowledge into clausal form (disjunction of literals)
  - See if contradiction indicated by null clause $\square$ can be derived

1. P

2. $P \rightarrow Q$ converted to $\sim P \vee Q$

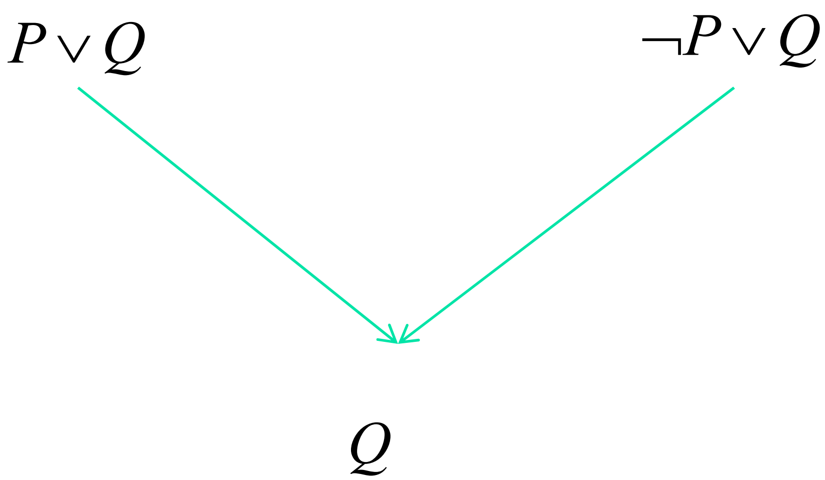
3. $\sim Q$

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



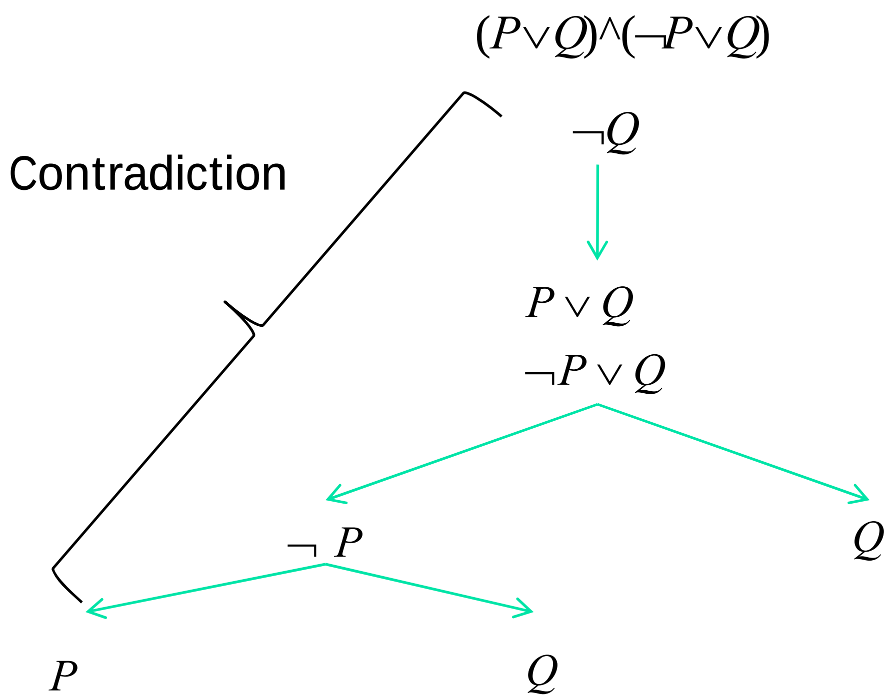
Theoretical basis of Resolution

- Resolution is proof by contradiction
- ***resolvent1 .AND. resolvent2 => resolute*** is a tautology



Tautologiness of Resolution

- Using Semantic Tree



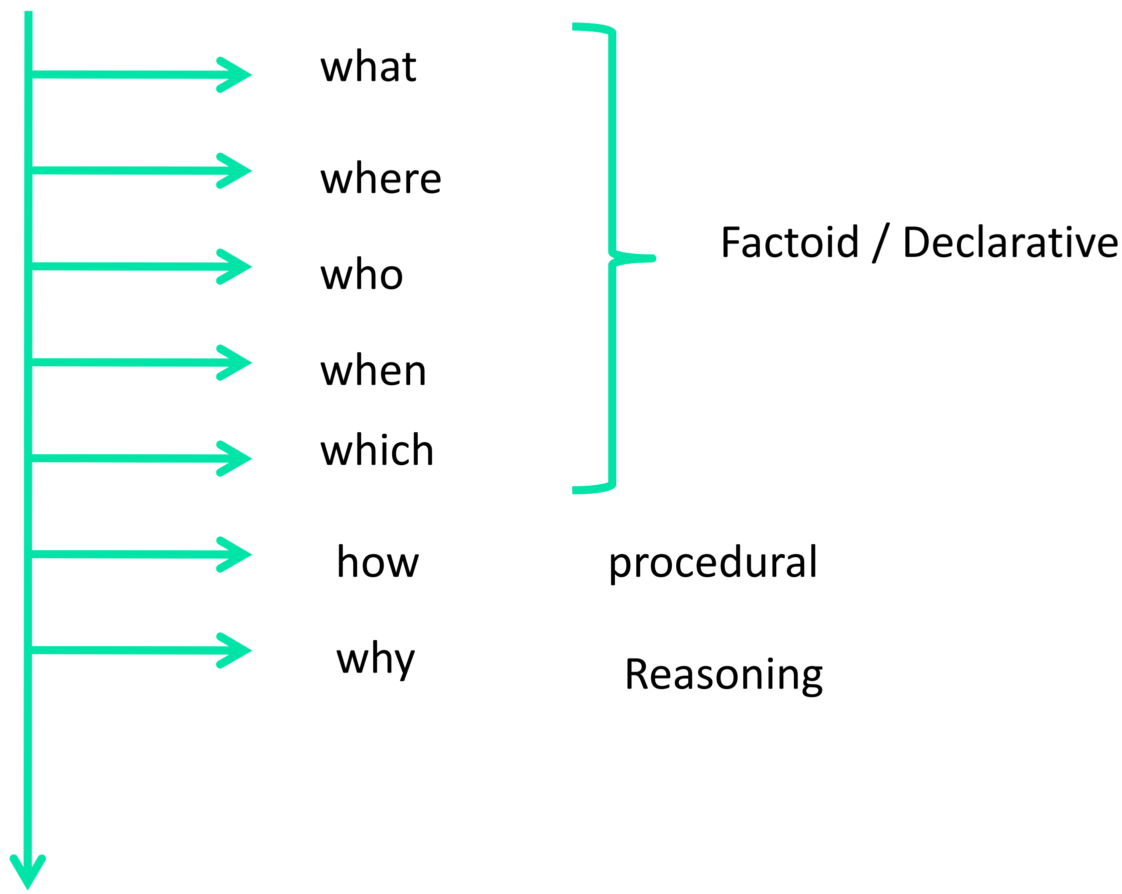
Theoretical basis of Resolution (cont ...)

- 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.

Wh-Questions and Knowledge



Fixing Predicates

- Natural Sentences

<Subject> <verb> <object>

Verb(subject,object)

predicate(subject)

Examples

- Ram is a boy
  - Boy(Ram)?
  - Is_a(Ram,boy)?

- Ram Playes Football
  - Plays(Ram,football)?
  - Plays_football(Ram)?

Knowledge Representation of Complex Sentence

- *"In every city there is a thief who is beaten by every policeman in the city"*

$$\forall x[\text{city}(x) \rightarrow \{\exists y((\text{thief}(y) \wedge \text{lives\_in}(y,x)) \wedge \forall z(\text{poleceman}(z,x) \rightarrow \text{beaten\_by}(z,y)))\}]$$

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

Example contd.

- Let *mc* denote mountain climber and *sk* denotes skier. Knowledge representation in the given problem is as follows:
  1. *member(A)*
  2. *member(B)*
  3. *member(C)*
  4. $\forall x[member(x) \rightarrow (mc(x) \vee sk(x))]$
  5. $\forall x[mc(x) \rightarrow \sim like(x,rain)]$
  6. $\forall x[sk(x) \rightarrow like(x, snow)]$
  7. $\forall x[like(B, x) \rightarrow \sim like(A, x)]$
  8. $\forall x[\sim like(B, x) \rightarrow like(A, x)]$
  9. *like(A, rain)*
  10. *like(A, snow)*
  11. *Question:* $\exists x[member(x) \wedge mc(x) \wedge \sim sk(x)]$
- We have to infer the 11^{th} expression from the given 10.
- Done through Resolution Refutation.

Club example: Inferencing

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

8. $\forall x[\sim lk(A,x) \rightarrow lk(B,x)]$

- $lk(A,x) \vee lk(B,x)$

9. $lk(A, rain)$

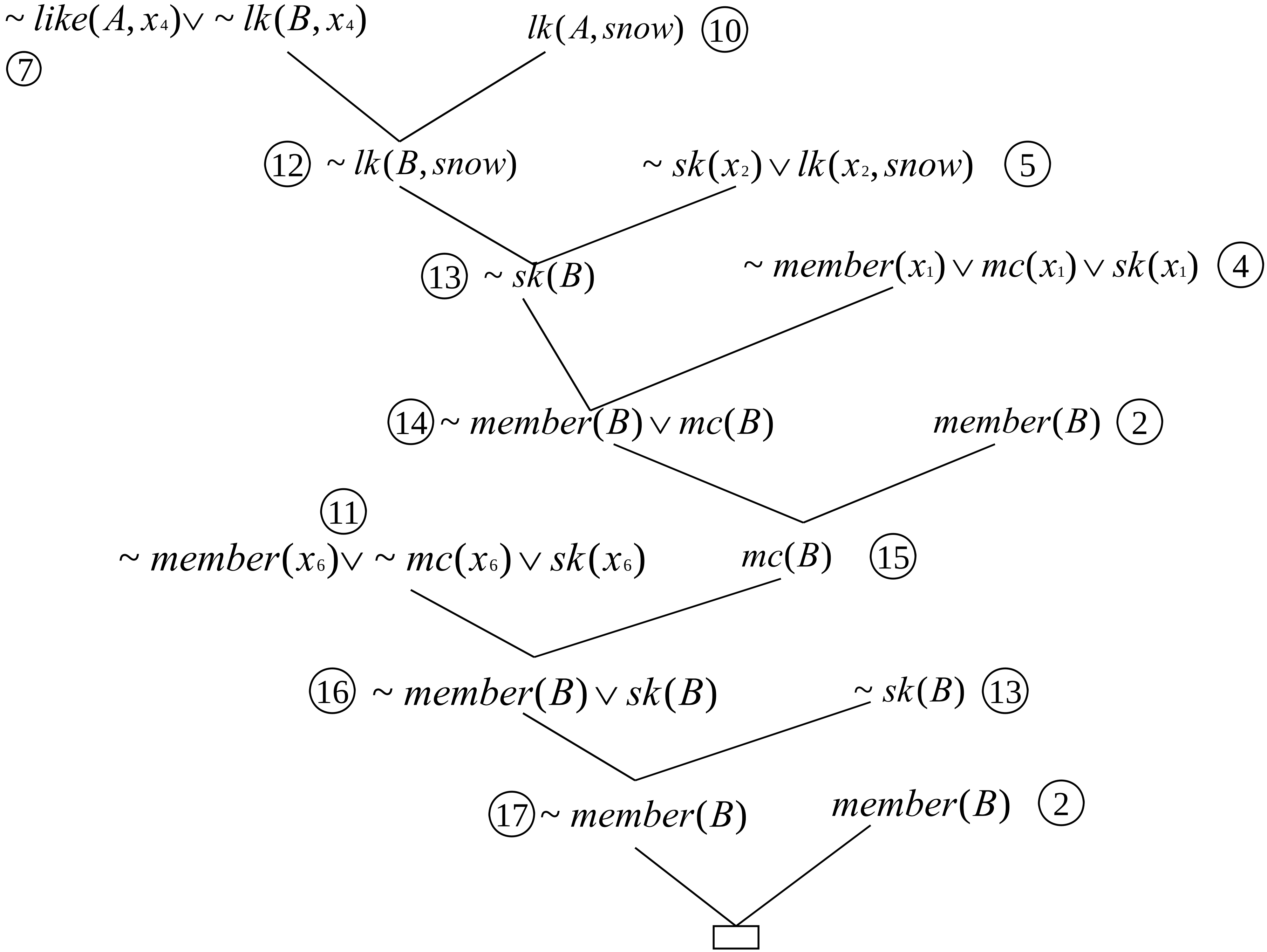
10. $lk(A, snow)$

11. $\exists x[member(x) \wedge mc(x) \wedge \sim sk(x)]$

- Negate– $\forall x[\sim member(x) \vee \sim mc(x) \vee sk(x)]$

- Now standardize the variables apart which results in the following

1. $member(A)$
2. $member(B)$
3. $member(C)$
4. $\sim member(x_1) \vee mc(x_1) \vee sk(x_1)$
5. $\sim sk(x_2) \vee lk(x_2, snow)$
6. $\sim mc(x_3) \vee \sim lk(x_3, rain)$
7. $\sim like(A, x_4) \vee \sim lk(B, x_4)$
8. $lk(A, x_5) \vee lk(B, x_5)$
9. $lk(A, rain)$
10. $lk(A, snow)$
11. $\sim member(x_6) \vee \sim mc(x_6) \vee sk(x_6)$


~ like(A, x4) ∨ ~ lk(B, x4)
7
lk(A, snow)
10
12
~ lk(B, snow)
~ sk(x2) ∨ lk(x2, snow)
5
13
~ sk(B)
~ member(x1) ∨ mc(x1) ∨ sk(x1)
4
14
~ member(B) ∨ mc(B)
member(B)
2
11
~ member(x6) ∨ ~ mc(x6) ∨ sk(x6)
mc(B)
15
16
~ member(B) ∨ sk(B)
~ sk(B)
13
17
~ member(B)
member(B)
2

Interpretation in Logic

- Logical expressions or formulae are "FORMS" (placeholders) for whom contents are created through interpretation.
- Example:

$$\exists F\left[\left\{F(a)=b\right\}\wedge\forall x\left\{P(x)\rightarrow\left(F(x)=g\left(x,F\left(h(x)\right)\right)\right)\right\}\right]$$

- This is a Second Order Predicate Calculus formula.
- Quantification on 'F' which is a function.

Examples

- Interpretation:1

  $D=N$ (natural numbers)

  $a = 0$ and $b = 1$

  $x \in N$

  $P(x)$ stands for $x > 0$

  $g(m,n)$ stands for $(m \times n)$

  $h(x)$ stands for $(x - 1)$

- Above interpretation defines **Factorial**

Examples (contd.)

- Interpretation:2

  D={strings)

  $a = b = \lambda$

  P(x) stands for “x is a non empty string”

  g(m, n) stands for “append head of m to n”

  h(x) stands for *tail(x)*

- Above interpretation defines “reversing a string”