

CS217: Artificial Intelligence and Machine Learning (associated lab: CS240)

Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay

*Week7 of 17feb25, Predicate Calculus,
Search in Chess*

Main points covered: week6 of
10feb25

Important points associated with FFNN BP

Local Minima

Momentum Factor

Symmetry Breaking

Hilbert's formalization of propositional calculus

1. Elements are *propositions* : Capital letters
2. Operator is only one : \rightarrow (called implies)
3. Special symbol F (called 'false')
4. Two other symbols : '(' and ')'
5. Well formed formula is constructed according to the grammar

$$WFF \rightarrow P/F/WFF \rightarrow WFF$$

6. Inference rule : only one

Given $A \rightarrow B$ and

A

write B

known as MODUS PONENS

7. Axioms : Starting structures

$$A1: \quad (A \rightarrow (B \rightarrow A))$$

$$A2: \quad ((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$$

$$A3 \quad (((A \rightarrow F) \rightarrow F) \rightarrow A)$$

This formal system defines the propositional calculus

A very useful theorem (Actually a meta theorem, called deduction theorem)

Statement

If

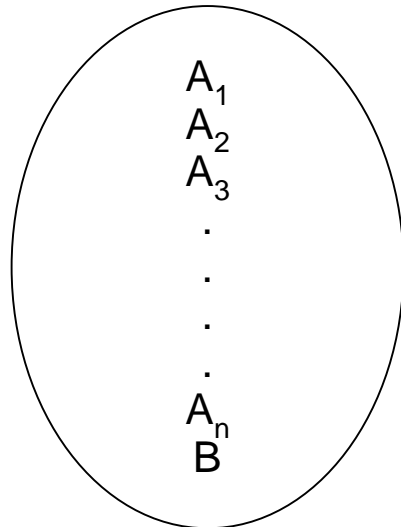
$$A_1, A_2, A_3 \dots\dots\dots A_n \vdash B$$

then

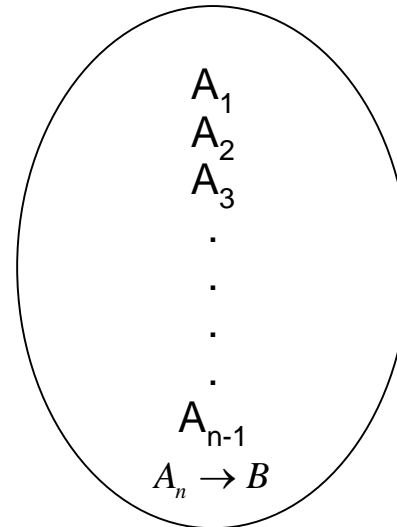
$$A_1, A_2, A_3, \dots\dots\dots A_{n-1} \vdash A_n \rightarrow B$$

\vdash is read as 'derives'

Given

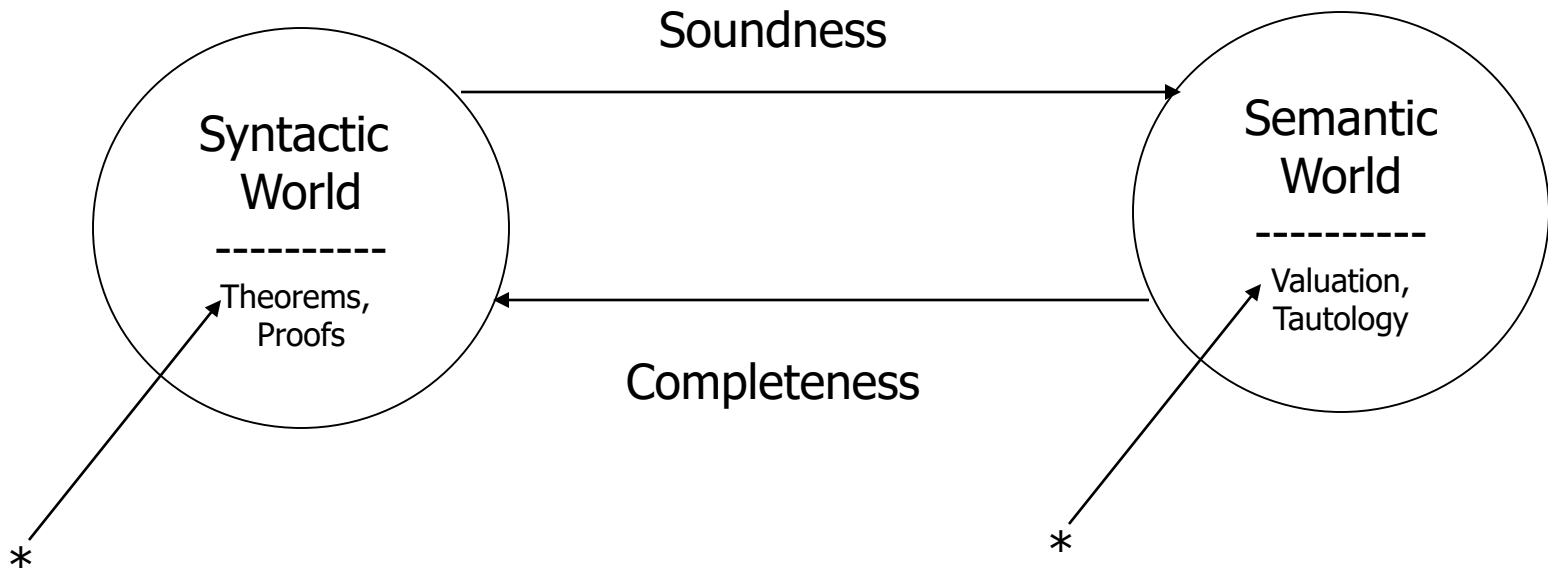


Picture 1



Picture 2

Soundness, Completeness & Consistency



Consistency

The System should not be able to
prove both P and $\sim P$, *i.e.*, should not be
able to derive

F

End of main points

Predicate calculus

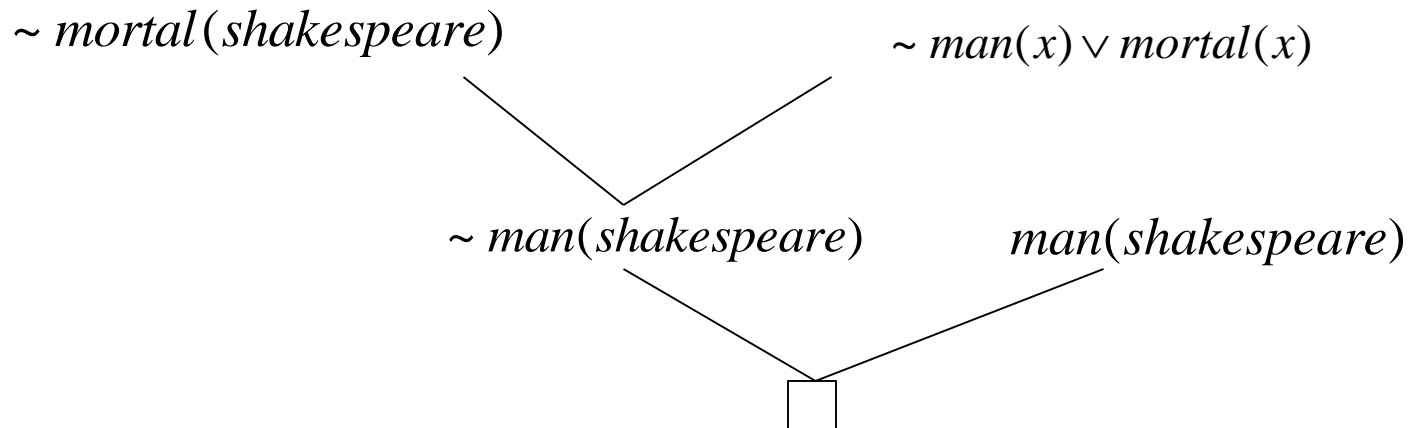
Insight into resolution

Resolution - Refutation

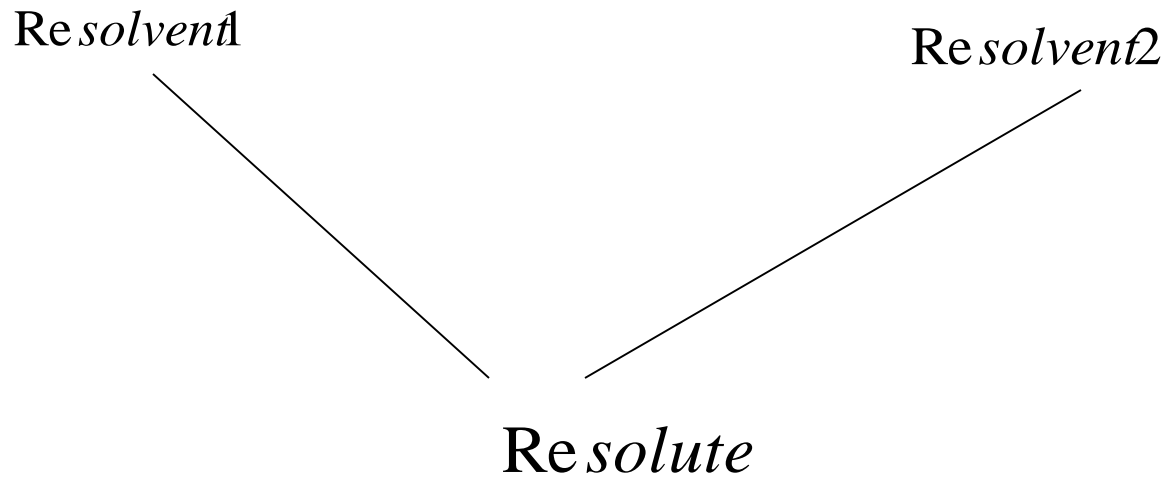
- $man(x) \rightarrow mortal(x)$
 - *Convert to clausal form*
 - $\sim man(shakespeare) \vee mortal(x)$
- **Clauses in the knowledge base**
 - $\sim man(shakespeare) \vee mortal(x)$
 - $man(shakespeare)$
 - $mortal(shakespeare)$

Resolution – Refutation contd

- *Negate the goal*
 - $\sim \text{man}(\text{shakespeare})$
- 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

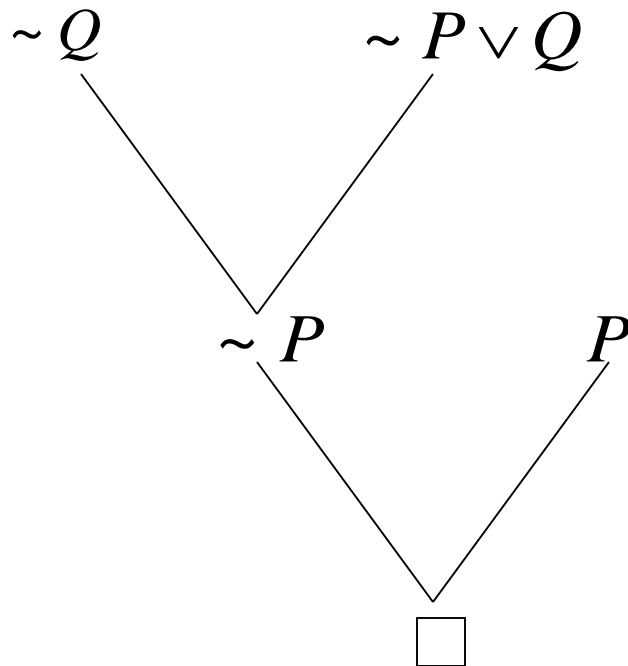
- Q , Match $R.H.S$ of $P \rightarrow Q$
- assert P
- Check if P exists

- 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

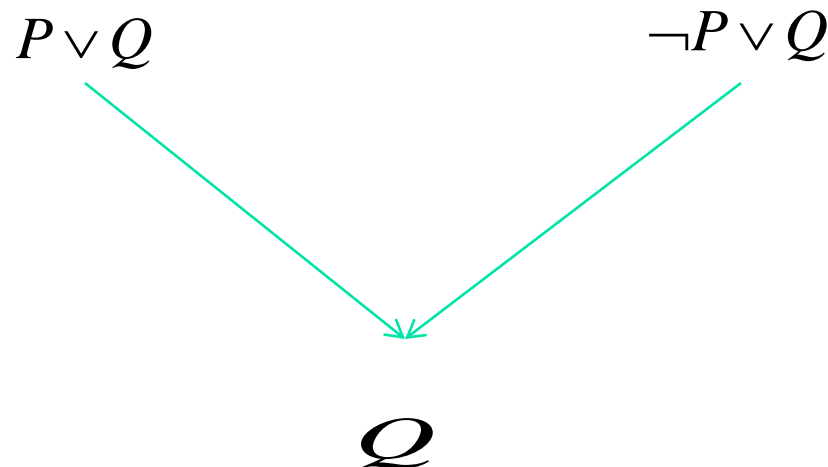
1. P
2. $P \rightarrow Q$ converted to $\sim P \vee Q$
3. $\sim Q$

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



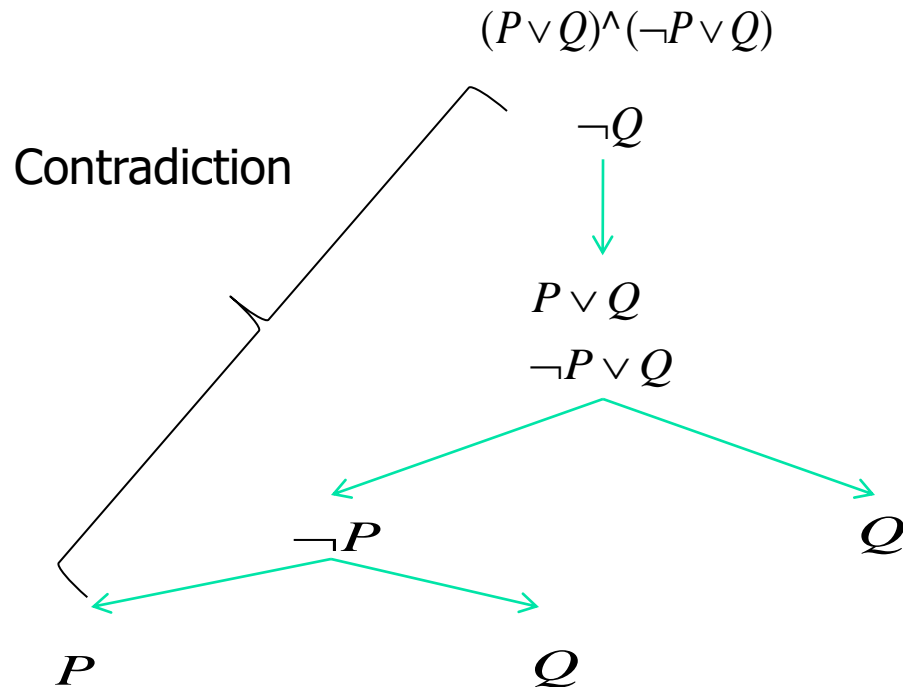
Theoretical basis of Resolution

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



Tautologiness of Resolution

- Using Semantic Tree (assume the RHS of \rightarrow is false)



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

Back to Himalayan Club

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 11th 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

– $\sim member(x) \vee mc(x) \vee sk(x)$

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

– $\sim like(A, x) \vee \sim lk(B, x)$

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

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

$$9. \quad lk(A, rain)$$

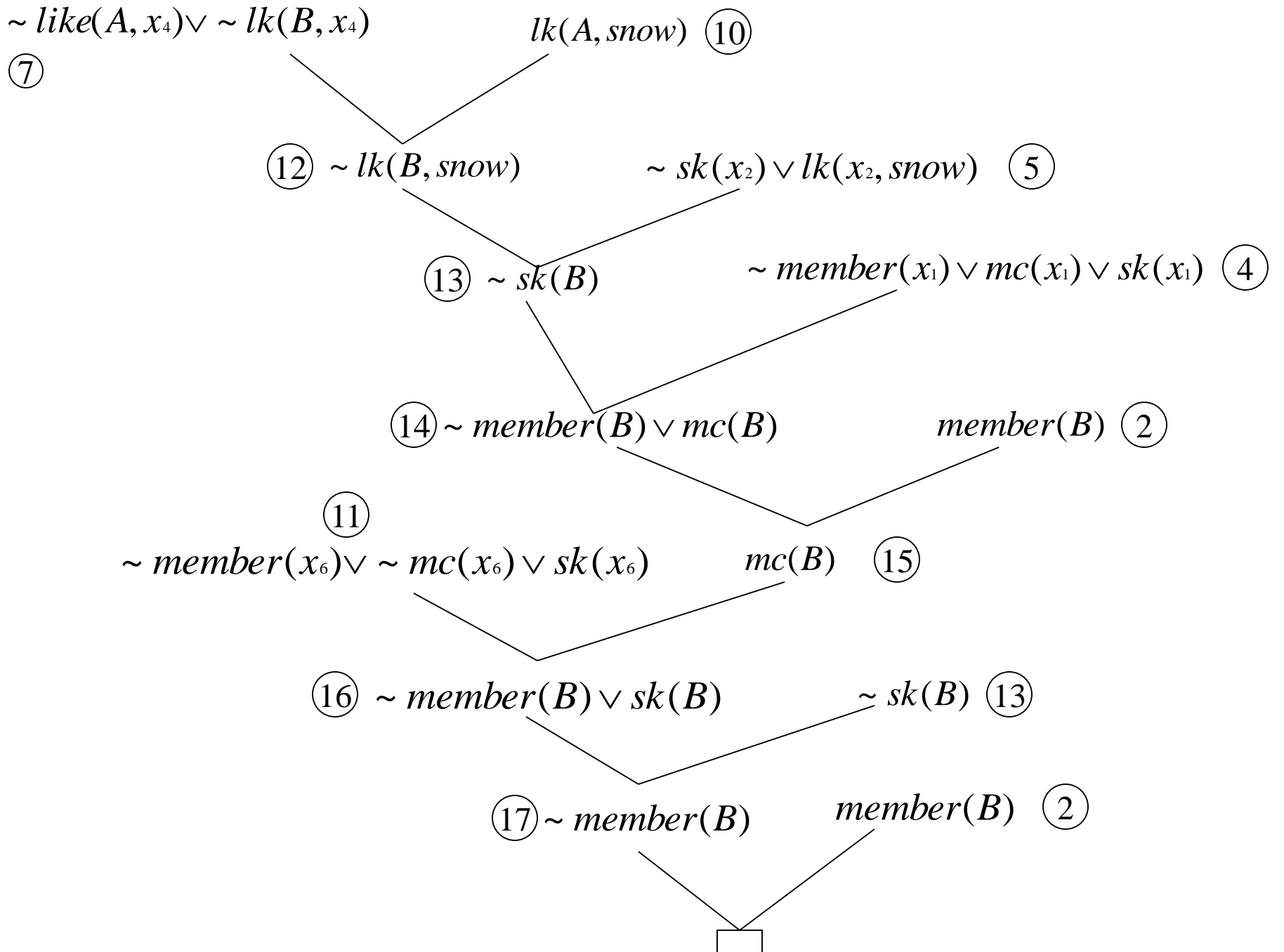
$$10. \quad lk(A, snow)$$

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

$$- \quad \text{Negate} - \quad \forall x[\sim member(x) \vee \sim mc(x) \vee sk(x)]$$

- 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. $\sim member(x_6) \vee \sim mc(x_6) \vee sk(x_6)$



Well known examples in Predicate Calculus

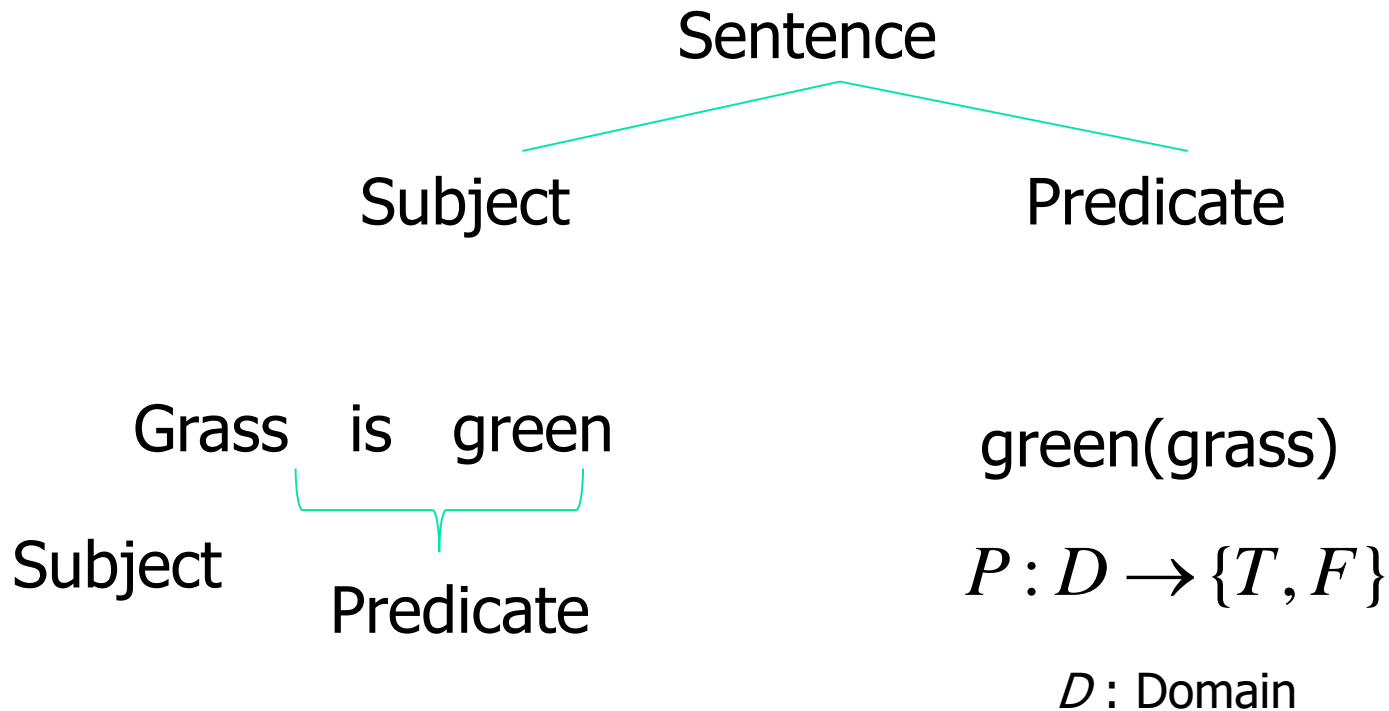
- Man is mortal : rule

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

- shakespeare is a man
man(shakespeare)
- To infer shakespeare is mortal
mortal(shakespeare)

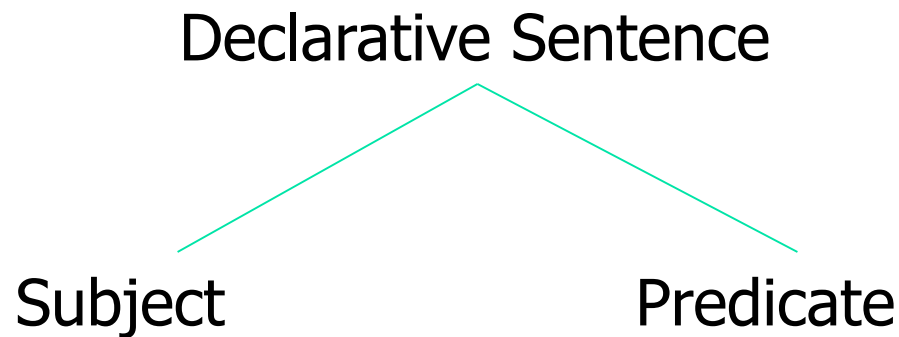
Predicate Calculus: origin

- Predicate calculus originated in language



Predicate Calculus: only for declarative sentences

- Is grass green? (Interrogative)
- Oh, grass is green! (Exclamatory)



- Grass which is supple is green

$$\forall x(\text{grass}(x) \wedge \text{supple}(x) \rightarrow \text{green}(x))$$

Predicate Calculus: more expressive power than propositional calculus

- 2 is even and is divisible by 2: P1
- 4 is even and is divisible by 2: P2
- 6 is even and is divisible by 2: P3

Generalizing,

$$\forall x((Integer(x) \wedge even(x) \Rightarrow divides(2, x))$$

Predicate Calculus: finer than propositional calculus

1. Finer Granularity (Grass is green, ball is green, leaf is green (green(x)))
2. Succinct description for infinite number of statements which would need ∞ number of properties

3 place predicate

Example: x gives y to z give(x,y,z)

4 place predicate

Example: x gives y to z through w give(x,y,z,w)

Double causative in Hindi giving rise to higher place predicates

- जॉन ने खाना खाया
John ne khana khaya
John <CM> food ate
John ate food
eat(John, food)
- जॉन ने जैक को खाना खिलाया
John ne Jack ko khana khilaya
John <CM> Jack <CM> food fed
John fed Jack
eat(John, Jack, food)
- जॉन ने जैक को जिल के द्वारा खाना खिलाया
John ne Jack ko Jill ke dvara khana khilaya
John <CM> Jack <CM> Jill <CM> food made-to-eat
John fed Jack through Jill
eat(John, Jack, Jill, food)

PC primitive: N-ary Predicate

$$P(a_1, \dots, a_n)$$

$$P : D^n \rightarrow \{T, F\}$$

- Arguments of predicates can be variables and constants
- Ground instances : Predicate all whose arguments are constants

N-ary Functions

$$f : D^n \rightarrow D$$

president(India) : Droupadi Murmu

- Constants & Variables : Zero-order objects
- Predicates & Functions : First-order objects

Prime minister of Fiji is older than the president of Fiji

older(prime_minister(Fiji), president(Fiji))

Operators

$$\wedge \vee \sim \oplus \forall \rightarrow \exists$$

- Universal Quantifier
- Existential Quantifier

All men are mortal

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

Some men are rich

$$\exists x[man(x) \wedge rich(x)]$$

Tautologies

$$\sim \forall x(p(x)) \rightarrow \exists x(\sim p(x))$$

$$\sim \exists x(p(x)) \rightarrow \forall x(\sim p(x))$$

- 2nd tautology in English:
 - *Not a single man in this village is educated implies all men in this village are uneducated*
- Tautologies are important instruments of logic, but uninteresting statements!

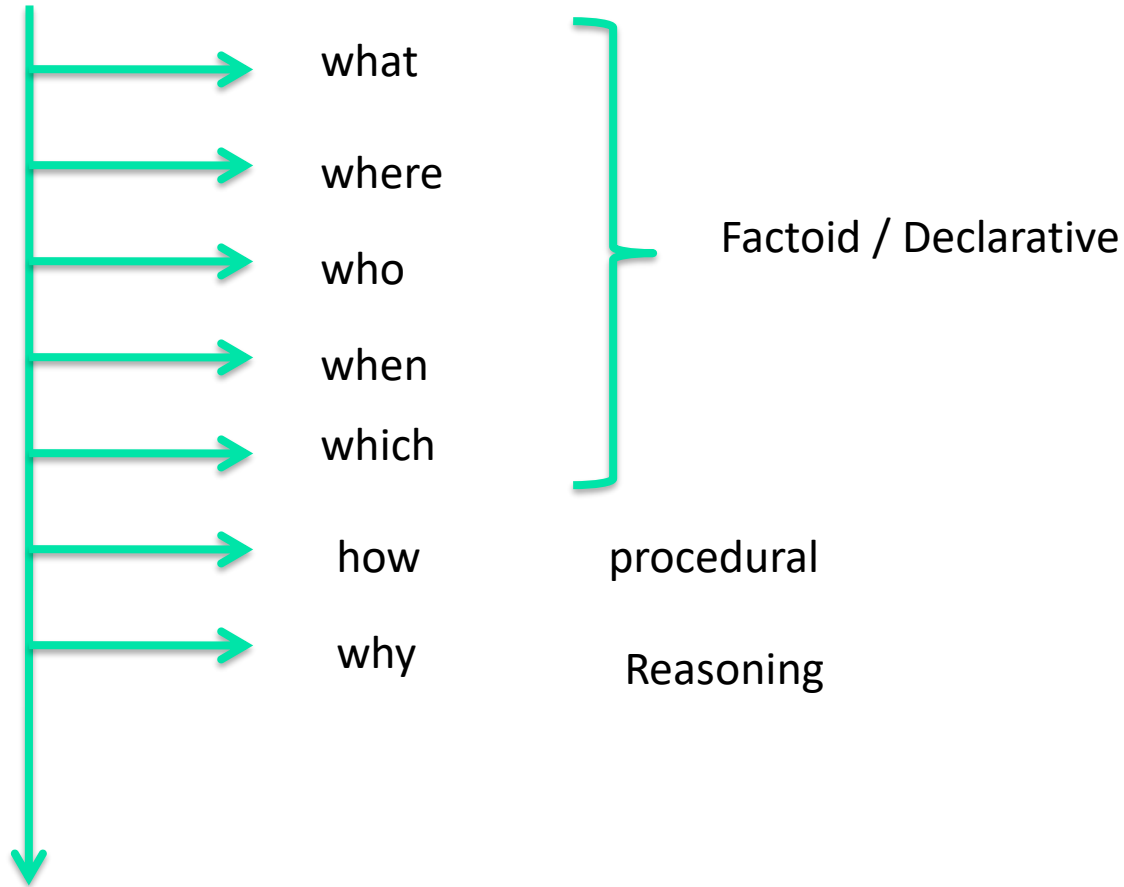
Inferencing: Forward Chaining

- $man(x) \rightarrow mortal(x)$
 - *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
 - $man(shakespeare)$

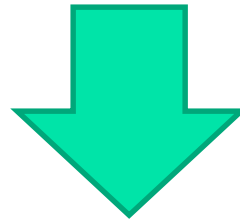
Wh-Questions and Knowledge



Fixing Predicates

- Natural Sentences

<Subject> <verb> <object>



Verb(subject,object)



predicate(subject)

Examples

- Ram is a boy
 - Boy(Ram)?
 - Is_a(Ram,boy)?
- Ram Plays 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"*

Knowledge Representation of Complex Sentence

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

$\forall x[\text{city}(x) \rightarrow \{\exists y((\text{thief}(y) \wedge \text{lives_in}(y, x)) \wedge \forall z(\text{policeman}(z, x) \rightarrow \text{beaten_by}(z, y)))\}]$

Interpretation in Logic

- Logical expressions or formulae are “FORMS” (placeholders) for whom contents are created through interpretation.

- Example:

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

- 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

$D = \{\text{strings}\}$

$a = b = \lambda$

$P(x)$ 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”