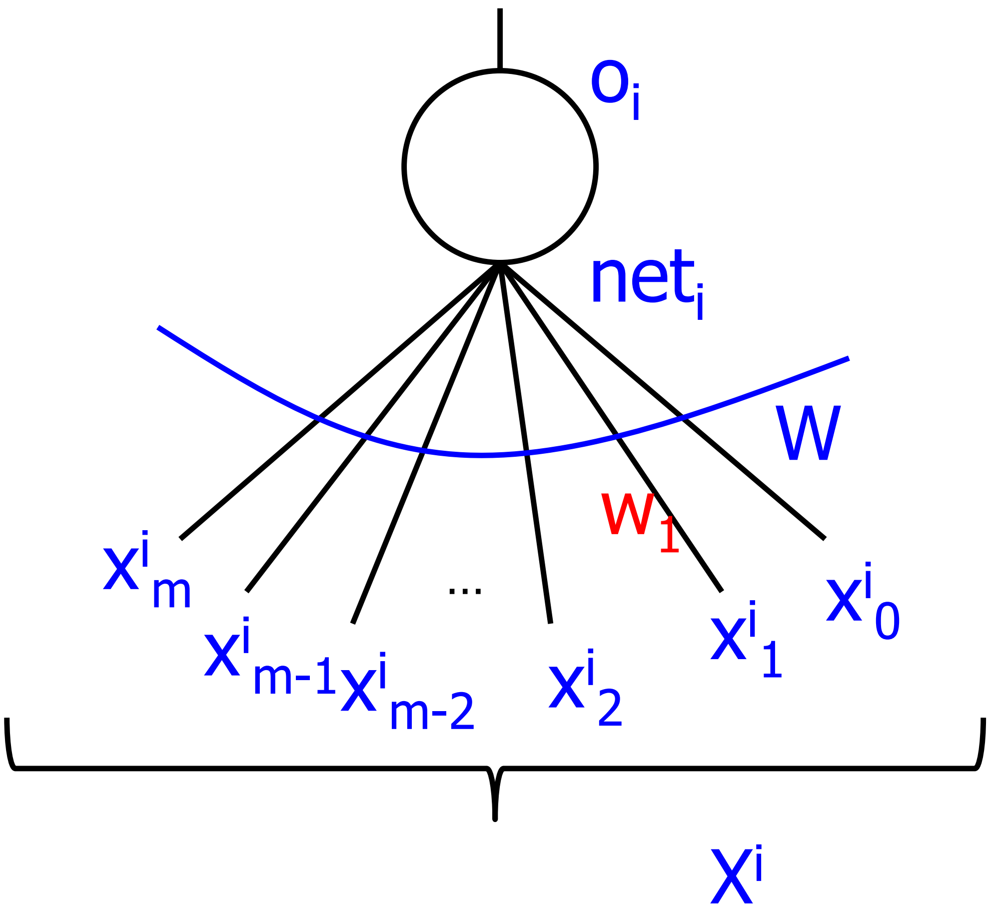
CS217: Artificial Intelligence and Machine Learning
(associated lab: CS240)

Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay

Week6 of 10feb25, BP finishing, Logic

Main points covered: week5 of 3feb25

Sigmoid neuron



$$o^i = \frac{1}{1+e^{-net^i}}$$

$$net_i = W.X^i = \sum_{j=0}^{m} w_j x_j^i$$

Softmax

$$\sigma(\bar{Z})_i = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}}$$

- σ is the **softmax** function
- Z is the input vector of size K
- The RHS gives the i^{th} component of the output vector
- Input to softmax and output of softmax are of the same dimension

Cross Entropy Function

$$H(P,Q) = -\sum_{x=1,N} \sum_{k=1,C} P(x,k) \log_2 Q(x,k)$$

x varies over *N* data instances, *c* varies over *C* classes
P is target distribution; *Q* is observed distribution

Sigmoid and Softmax: weight change rule

With Cross Entropy Loss, the change in any weight is

learning rate *

diff between target and observed outputs *

input at the connection

General Backpropagation Rule

- General weight updating rule:

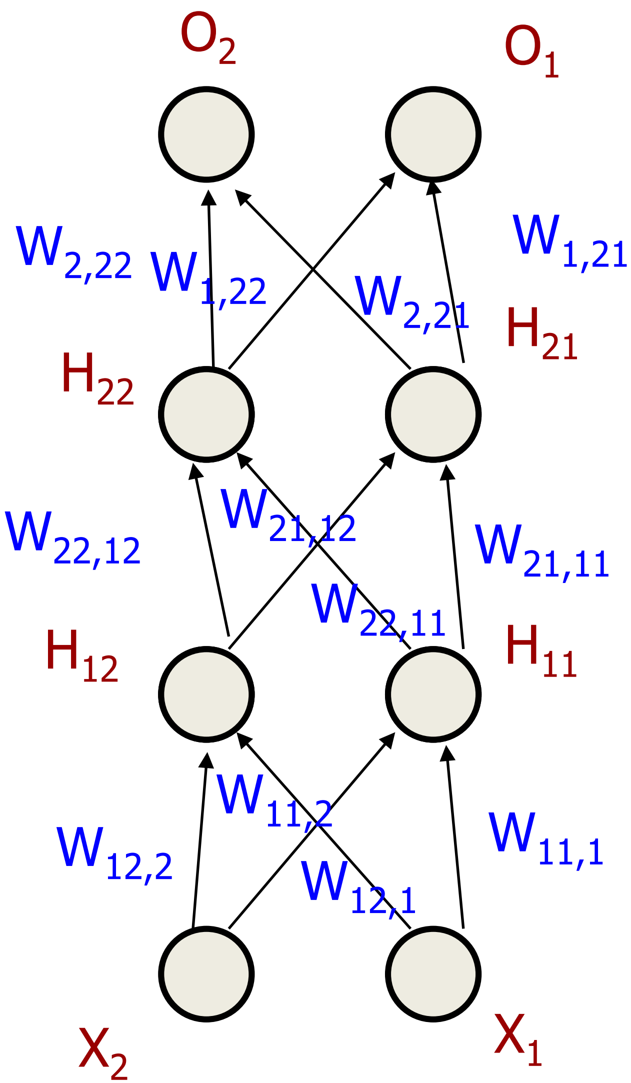
$$\Delta w_{ji} = \eta \delta j o_i$$

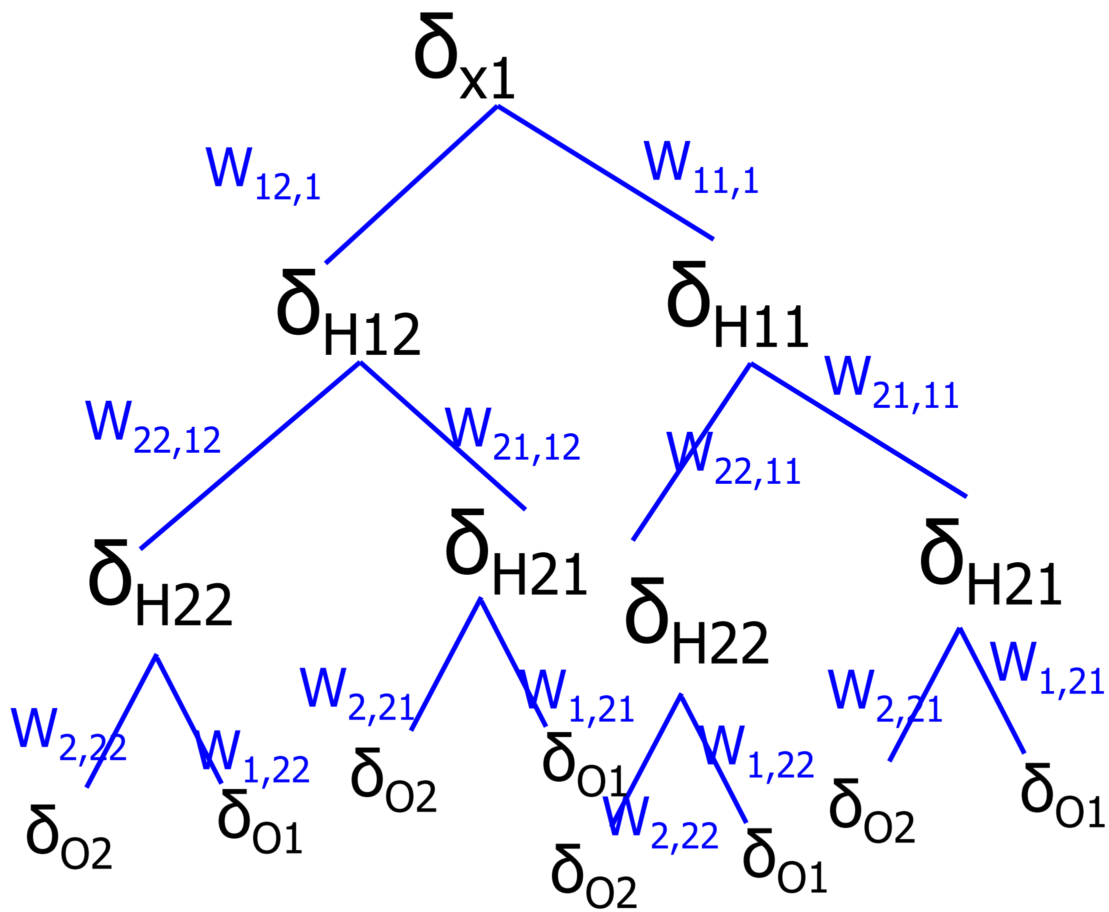
- Where

$$\delta_j = (t_j - o_j) o_j (1 - o_j) \quad \text{for outermost layer}$$

$$= \sum_{k \in \text{next layer}} (w_{kj} \delta_k) o_j (1 - o_j) \quad \text{for hidden layers}$$

Vanishing/Exploding Gradient

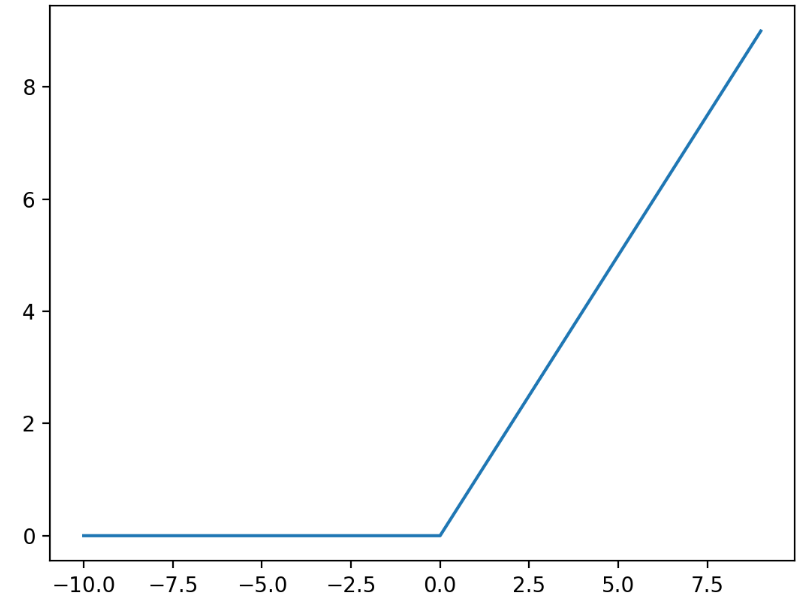




$$\delta_{x1}=W_{11,1}\delta_{H11}+W_{12,1}\delta_{H12}$$

RELU and Vanishing Gradient

y=relu(x)=max(0,x)



dy/dx

= 0 for x<o

= 1 for x>0

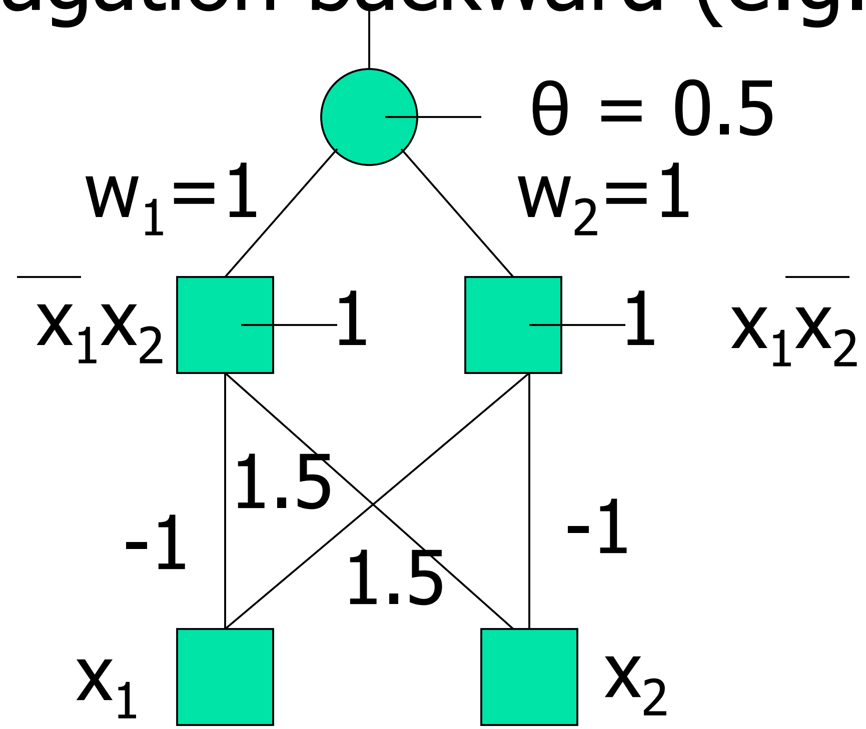
= 0 (forced to be 0 at x=0, though does not exit)

End of main points

Important concepts associated with FFNN-BP

How does BP work?

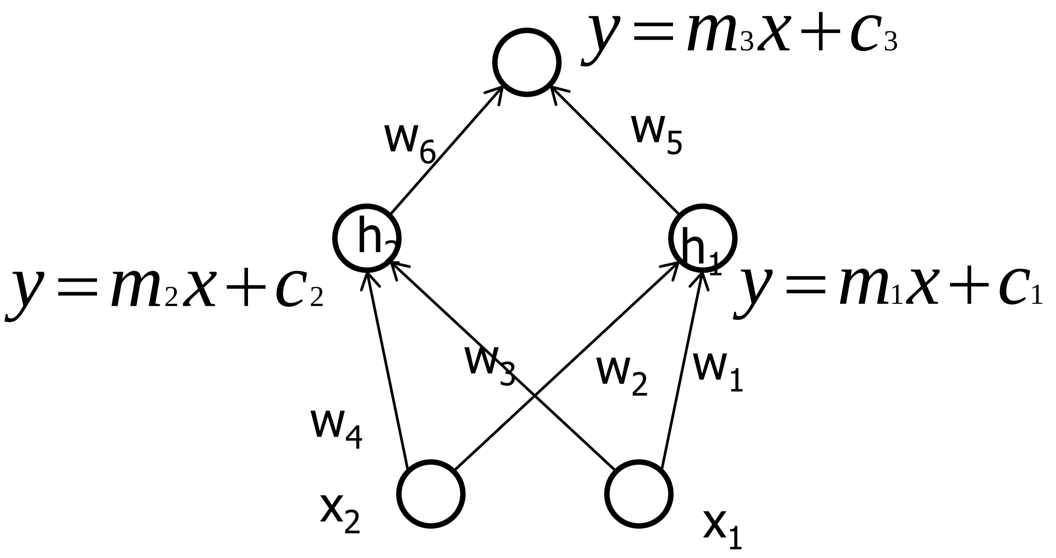
- Input propagation forward and error propagation backward (e.g. XOR)



Work it out !

1) In the XOR network, if the activation function of the hidden layer neurons is changed from sigmoid to the ReLU function how will the weight update rule change for minimizing the 'total sum-squared error' of the network?
2) Suppose we have two neurons each in both the hidden and the output layer. Softmax is used at the output. Find out the weight update expressions for the following two cases:
   a) The hidden layer uses ReLU activation.
   b) The hidden layer uses sigmoid activation.

Can Linear Neurons Work?



$$h_1 = m_1(w_1 x_1 + w_2 x_2) + c_1$$

$$h_2 = m_2(w_3 x_1 + w_4 x_2) + c_2$$

$$Out = (w_5 h_1 + w_6 h_2) + c_3$$

$$= k_1 x_1 + k_2 x_2 + k_3$$

Note: The whole structure shown in earlier slide is reducible to a single neuron with given behavior

$$Out = k_1 x_1 + k_2 x_2 + k_3$$

Claim: A neuron with linear I-O behavior can't compute X-OR.

Proof: Considering all possible cases:

[assuming 0.1 and 0.9 as the lower and upper thresholds]

For (0,0), Zero class:
$$m(w_1.0 + w_2.0 - \theta) + c < 0.1$$
$$\Rightarrow c - m.\theta < 0.1$$

For (0,1), One class:
$$m(w_1.1 + w_2.0 - \theta) + c > 0.9$$
$$\Rightarrow m.w_1 - m.\theta + c > 0.9$$

For (1,0), One class: $m.w_2 - m.\theta + c > 0.9$

For (1,1), Zero class: $m.(w_1 + w_2) - m\ .\theta + c < 0.1$

These equations are inconsistent. Because when we add these inequalities after adjusting for sign, we get 0>1.6!

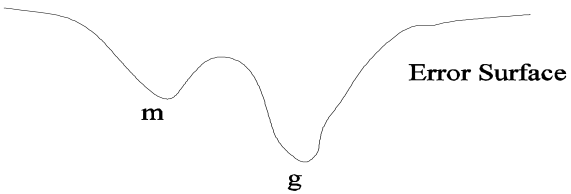
Hence X-OR can't be computed.

Observations:

1. A linear neuron can't compute X-OR.
2. A multilayer FFN with linear neurons is collapsible to a single linear neuron, hence **no a additional power due to hidden layer.**
3. Non-linearity is essential for power.

Local Minima

Due to the Greedy nature of BP, it can get stuck in local minimum *m* and will never be able to reach the global minimum *g* as the error can only decrease by weight change.



m- local minima, g- global minima

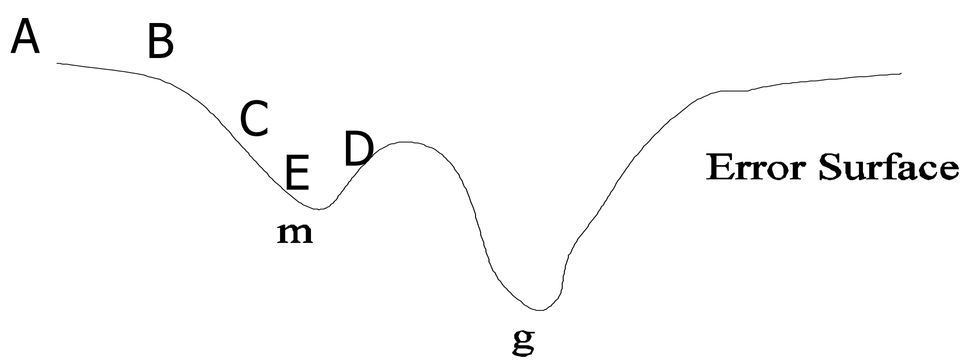
Figure- Getting Stuck in local minimum

Momentum factor

1. Introduce momentum factor.

- Accelerates the movement out of the trough.
- Dampens oscillation inside the trough.
- Choosing β : If β is large, we may jump over the minimum.

$$(\Delta w_{ji})_{nth-iteration} = \eta \delta_j O_i + \beta (\Delta w_{ji})_{(n-1)th-iteration}$$

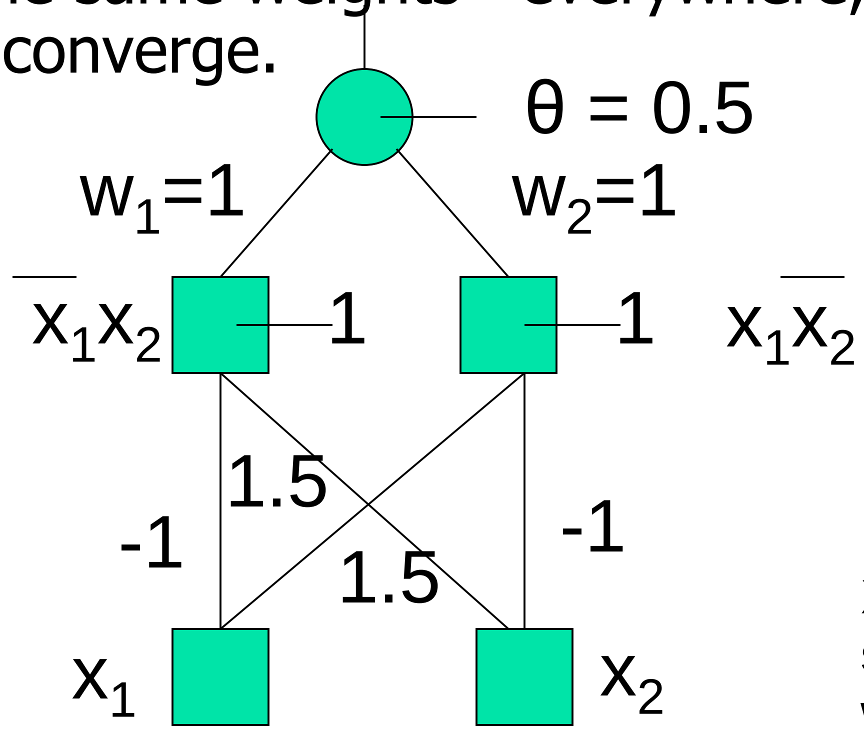


m- local minima, g- global minima

Figure- Getting Stuck in local minimum

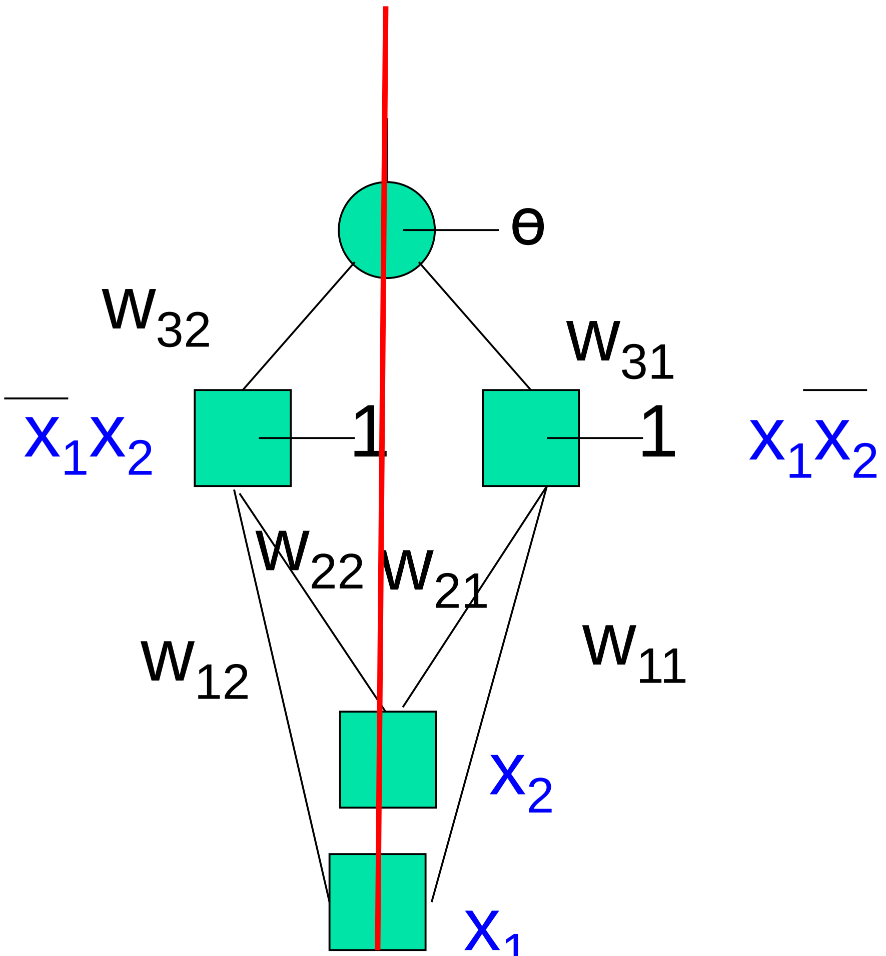
Symmetry breaking

- If mapping demands different weights, but we start with the same weights everywhere, then BP will never converge.



XOR n/w: if we s started with identical weight everywhere, BP will not converge

Symmetry breaking: understanding with proper diagram



Symmetry About The red Line should Be broken

Formal Logic

Theory of CS

- Theory A
  - Logic
- Theory B
  - Algorithm an Complexity

Concepts, Axioms, Rule

- Some foundational questions for Mechanization or Automation of Knowledge Representation and Reasoning:
  - What are symbols and concepts (well formed formulae)
  - What are the self evident and ground truths in the system (axiomatization)
  - What is the validity of the inference (soundness and consistency)
  - Is the inference system powerful enough to capture reality (completeness)
  - Can it be implemented in Turing machine (derivability and complexity)

Case study: Propositional calculus

Propositions

- Stand for facts/assertions
- Declarative statements
  - As opposed to interrogative statements (questions) or imperative statements (request, order)

Operators

AND ($\wedge$), OR ($\vee$), NOT ($\neg$), IMPLICATION (=>)

=> and ¬ form a minimal set (can express other operations)

- Prove it.

Tautologies are formulae whose truth value is always T, whatever the assignment is

Model

In propositional calculus any formula with n propositions has 2^n models (assignments)

- Tautologies evaluate to T in all models.

Examples:

1) $$P \vee \neg P$$

2) $$\neg(P \wedge Q) \Leftrightarrow (\neg P \vee \neg Q)$$

- De Morgan with AND

Example

- Prove $\sim(P \wedge Q) \rightarrow (\sim P \vee \sim Q)$ is a Tautology.

Q	P	$L = \sim(P \wedge Q)$	$R = \sim P \vee \sim Q$	$L \rightarrow R$
T	T	F	F	T
T	F	T	T	T
F	T	T	T	T
F	F	T	T	T

Formal Systems

- Rule governed
- Strict description of structure and rule application
- Constituents
  - Symbols
  - Well formed formulae
  - Inference rules
  - Assignment of semantics
  - Notion of proof
  - Notion of soundness, completeness, consistency, decidability etc.

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: $(A \to (B \to A))$

A2: $((A \to (B \to C)) \to ((A \to B) \to (A \to C)))$

A3 $(((A \to F) \to F) \to A)$

This formal system defines the propositional calculus

Notion of proof

1. Sequence of well formed formulae
2. Start with a set of hypotheses
3. The expression to be proved should be the last line in the sequence
4. Each intermediate expression is either one of the hypotheses or one of the axioms or the result of modus ponens
5. An expression which is proved only from the axioms and inference rules is called a THEOREM within the system

Example of proof

From P and $P \to Q$ and $Q \to R$ prove R

H1: P

H2: $P \to Q$

H3: $Q \to R$

i)	P	H1
ii)	$P \to Q$	H2
iii)	Q	MP, (i), (ii)
iv)	$Q \to R$	H3
v)	R	MP, (iii), (iv)

Prove that $(P \to P)$ is a THEOREM

i) $P \to (P \to P)$ A1 : P for A and B

ii) $P \to ((P \to P) \to P)$ A1: P for A and $(P \to P)$ for B

iii) $[(P \to ((P \to P) \to P)) \to ((P \to (P \to P)) \to (P \to P))]$

A2: with P for A, $(P \to P)$ for B and P for C

iv) $(P \to (P \to P)) \to (P \to P)$ MP, (ii), (iii)

v) $(P \to P)$ MP, (i), (iv)

Shorthand

1. $\neg P$ is written as $P \to F$ and called '*NOT P*'

2. $((P \to F) \to Q)$ is written as $(P \vee Q)$ and called

   '*P OR Q*'

3. $((P \to (Q \to F)) \to F)$ is written as $(P \wedge Q)$ and called

   '*P AND Q*'

Exercise: (Challenge)

- Prove that $A \to \neg(\neg(A))$

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

<u>Statement</u>

If

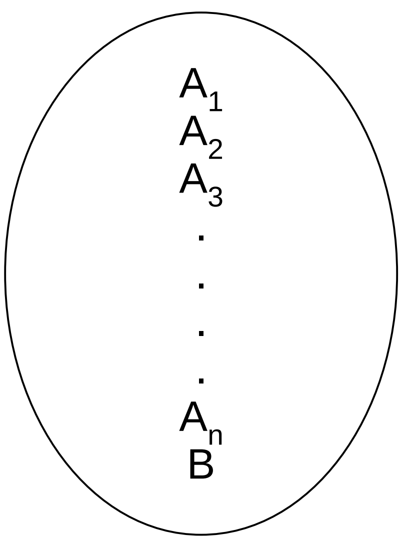
$A_1, A_2, A_3 \ldots\ldots\ldots\ldots A_n \vdash B$

then

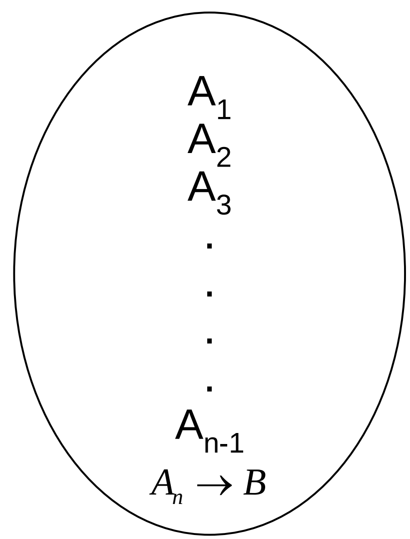
$A_1, A_2, A_3, \ldots\ldots\ldots\ldots\ldots A_{n-1} \vdash \quad A_n \rightarrow B$

$\vdash$ is read as 'derives'

Given



Picture 1



Picture 2

Use of Deduction Theorem

Prove

$$A \rightarrow \neg(\neg(A))$$

i.e., $$A \rightarrow ((A \rightarrow F) \rightarrow F)$$

$A, A \rightarrow F \vdash F$ (M.P)

$A \vdash (A \rightarrow F) \rightarrow F$ (D.T)

$\vdash A \rightarrow ((A \rightarrow F) \rightarrow F)$ (D.T)

Very difficult to prove from first principles, *i.e.*, using axioms and inference rules only

Prove $P \to (P \vee Q)$

i.e. $P \to ((P \to F) \to Q)$

$$P, P \to F, Q \to F \vdash F$$

$$P, P \to F \vdash (Q \to F) \to F \qquad \text{(D.T)}$$

$$\vdash Q \qquad \text{(M.P with A3)}$$

$$P \vdash (P \to F) \to Q$$

$$\vdash P \to ((P \to F) \to Q)$$

More proofs

1. $(P \wedge Q) \rightarrow (P \vee Q)$
2. $(P \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P)$
3. $(P \rightarrow Q) \rightarrow ((\neg Q \rightarrow P) \rightarrow Q)$

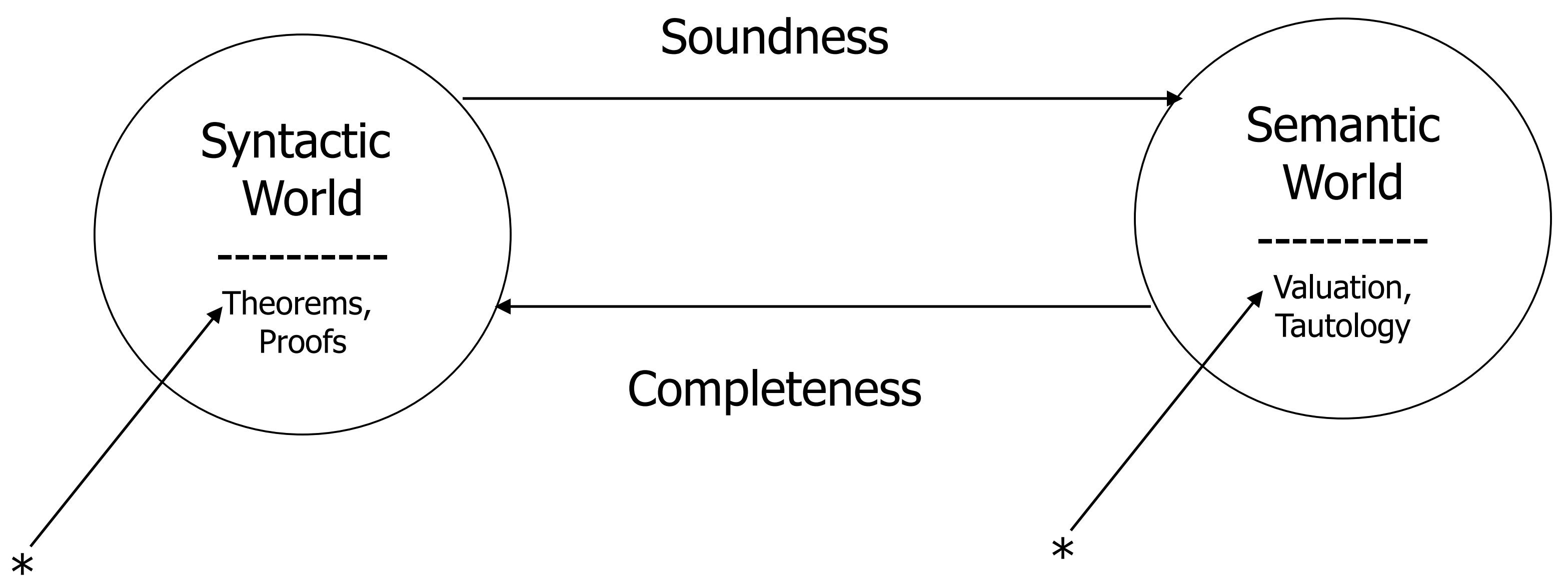
Important to note

- Deduction Theorem is a meta-theorem (statement **about** the system)
- $P \rightarrow P$ is a theorem (statement **belonging to** the system**)**
- The distinction is crucial in AI
- Self reference, diagonalization
- Foundation of Halting Theorem, Godel Theorem etc.

Example of '*of-about*' confusion

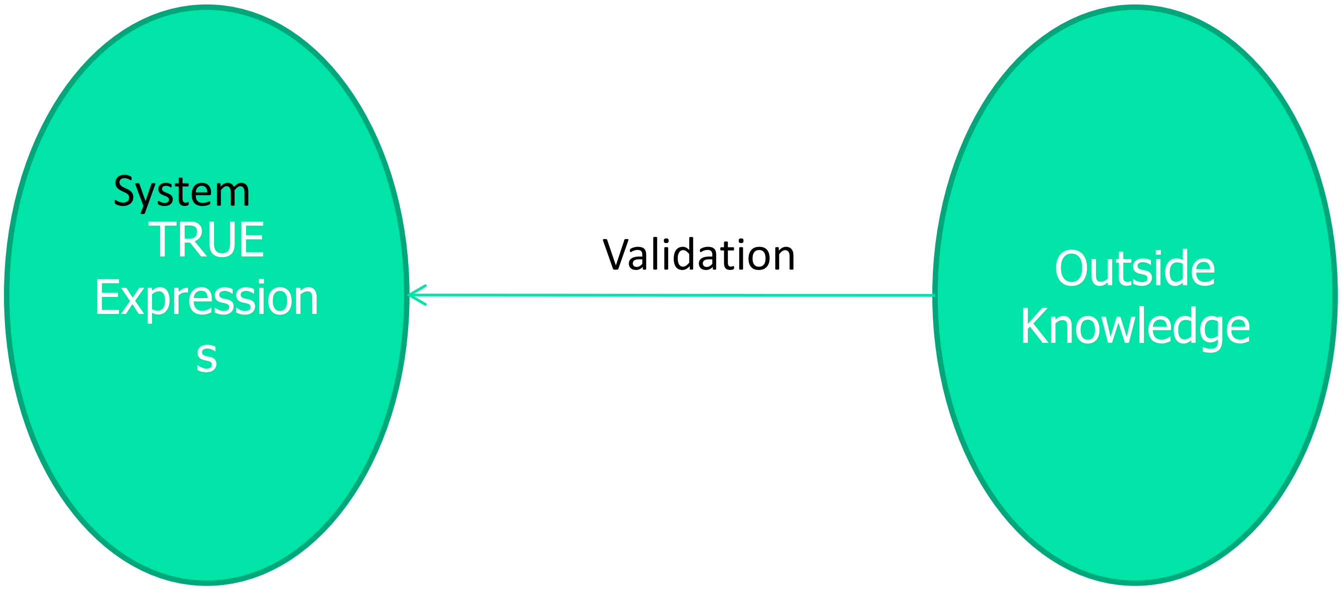
- "*This statement is false*"
- Truth of falsity cannot be decided
- Another example: "A city has a barber that shaves ALL AND ONLY those who do NOT shave themselves; Question- does the barber shave himself?"
  - Cannot be answered

Soundness, Completeness & Consistency



- <u>Soundness</u>
  - Provability $\longrightarrow$ Truth
- <u>Completeness</u>
  - Truth $\longrightarrow$ Provability

- <u>Soundness:</u> Correctness of the System
    - Proved entities are indeed true/valid
- <u>Completeness:</u> Power of the System
    - True things are indeed provable


System
TRUE
Expression
s
Validation
Outside
Knowledge

Consistency

The System should not be able to

prove both P and ~P, *i.e.*, should not be

able to derive

$\boxed{F}$

Examine the relation between

Soundness

&

Consistency

Soundness $\equiv$ Consistency

If a System is inconsistent, *i.e.,* can derive

$\mathcal{F}$, it can prove any expression to be a

theorem. Because

$\mathcal{F} \rightarrow$ P is a theorem

Inconsistency→Unsoundness

To show that

$\mathcal{F} \rightarrow P$ is a theorem

Observe that

$\mathcal{F}, P \rightarrow \mathcal{F} \vdash \mathcal{F}$ By D.T.

$\mathcal{F} \vdash (P \rightarrow \mathcal{F}) \rightarrow \mathcal{F}; \mathcal{A}3$

$\vdash P$

i.e. $\vdash \mathcal{F} \rightarrow P$

Thus, inconsistency implies unsoundness

Unsoundness→Inconsistency

- Suppose we make the Hilbert System of propositional calculus unsound by introducing *(A /| B)* as an axiom
- Now AND can be written as
  - $(A \rightarrow (B \rightarrow \mathcal{F})) \rightarrow \mathcal{F}$
- If we assign $\mathcal{F}$ to A, we have
  - $(\mathcal{F} \rightarrow (B \rightarrow \mathcal{F})) \rightarrow \mathcal{F}$
  - But $(\mathcal{F} \rightarrow (B \rightarrow \mathcal{F}))$ is an axiom (A1)
  - Hence $\mathcal{F}$ is derived

Inconsistency is a <u>Serious</u> issue.

Informal Statement of Godel Theorem:

If a sufficiently powerful system is complete it is inconsistent.

Sufficiently powerful: Can capture at least Peano Arithmetic

Introduce Semantics in Propositional logic

Valuation Function V

<u>Definition of V</u>

Syntactic ‘false

$$V(\mathscr{F}) = F$$

Semantic ‘false’

Where F is called ‘false’ and is one of the two symbols (T, F)

$V(\mathscr{F}) = F$

$V(A \rightarrow B)$ is defined through what is called the truth table

$V(A)$	$V(B)$	$V(A \rightarrow B)$
T	F	F
T	T	T
F	F	T
F	T	T

Tautology

An expression ‘E’ is a tautology if

$$V(E) = T$$

for all valuations of constituent propositions

Each ‘valuation’ is called a ‘model’.

To see that

$(\mathscr{F} \rightarrow P)$ is a tautology

two models

$V(P) = T$

$V(P) = F$

$V(\mathscr{F} \rightarrow P) = T$ for both

$$\begin{tikzcd}
\mathcal{F} \rightarrow \mathrm{P} \text{ is a theorem} \arrow[d, shift right=2, "\text{Soundness}"'] \\
\mathcal{F} \rightarrow \mathrm{P} \text{ is a tautology} \arrow[u, shift right=2, "\text{Completeness}"']
\end{tikzcd}$$

If a system is Sound & Complete, it does not

<u>matter</u> how you “Prove” or “show the validity”

Take the Syntactic Path or the Semantic Path

Syntax vs. Semantics issue

Refers to

FORM VS. CONTENT

Form

Tea

(Content)

Form & Content

painter

musician

logician

Godel, Escher, Bach

By D. Hofstadter

Problem

$$(P \wedge Q) \rightarrow (P \vee Q)$$

Semantic Proof

P	Q	A $P \wedge Q$	B $P \vee Q$	$A \rightarrow B$
T	F	F	T	T
T	T	T	T	T
F	F	F	F	T
F	T	F	T	T

To show syntactically

$$(P \wedge Q) \rightarrow (P \vee Q)$$

i.e.

$$[(P \rightarrow (Q \rightarrow \mathscr{F})) \rightarrow \mathscr{F}] \rightarrow [(P \rightarrow \mathscr{F}) \rightarrow Q]$$

If we can establish

$$(P \rightarrow (Q \rightarrow \mathcal{F})) \rightarrow \mathcal{F},$$

$$(P \rightarrow \mathcal{F}),\ Q \rightarrow \mathcal{F} \vdash \mathcal{F}$$

This is shown as

$$Q \rightarrow \mathcal{F} \quad \textit{hypothesis}$$

$$(Q \rightarrow \mathcal{F}) \rightarrow (P \rightarrow (Q \rightarrow \mathcal{F})) \quad \textit{A1}$$

$Q \rightarrow \mathcal{F}$; hypothesis

$(Q \rightarrow \mathcal{F}) \rightarrow (P \rightarrow (Q \rightarrow \mathcal{F}))$; A1

$P \rightarrow (Q \rightarrow \mathcal{F})$; MP

$\mathcal{F}$; MP

Thus we have a proof of the line we started with

Predicate calculus

Introduce through the “Himalayan Club Example”

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. $lk(A, rain)$

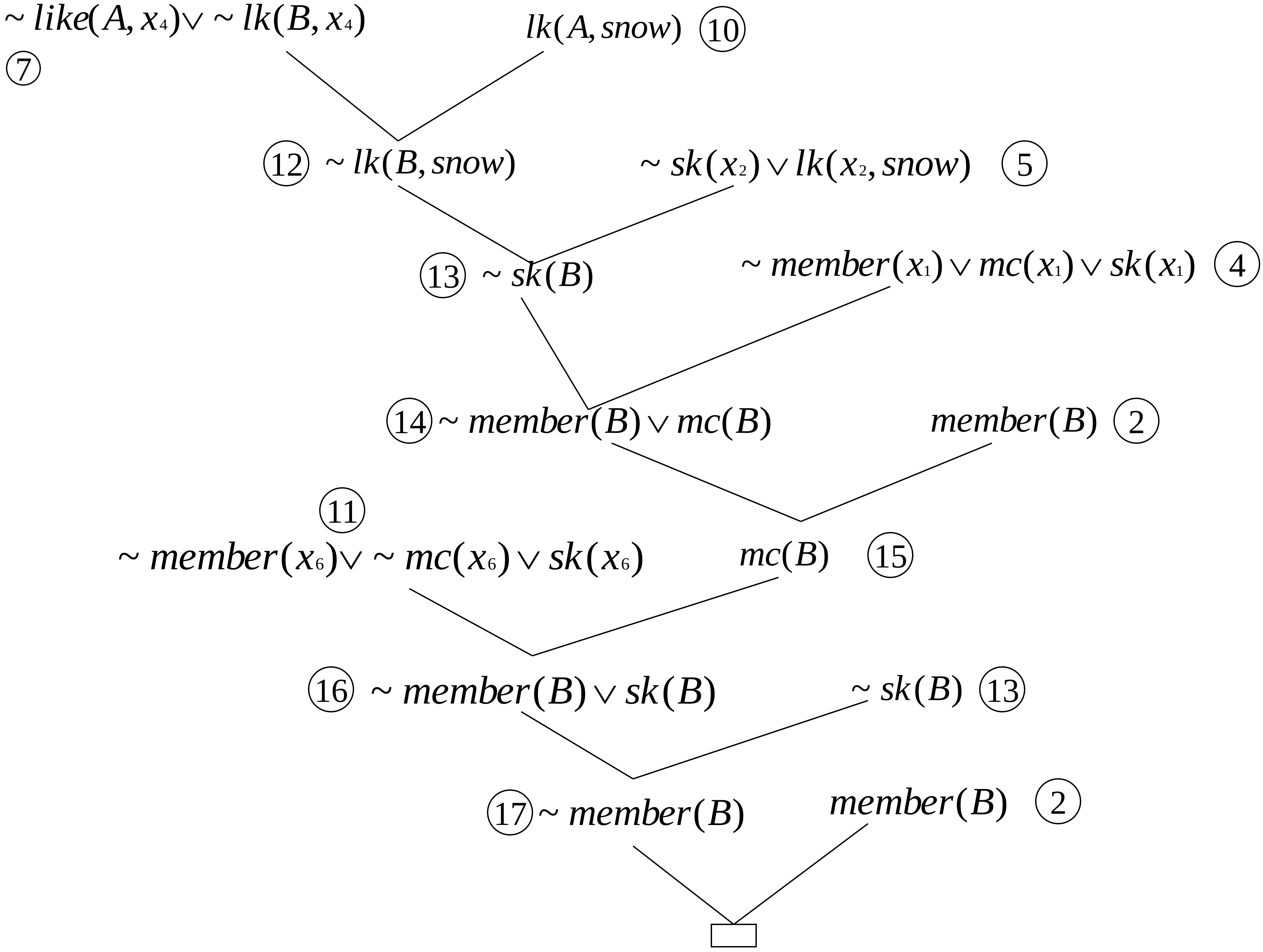
10. $lk(A, snow)$

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


$\sim like(A, x_4) \vee \sim lk(B, x_4)$ 7
$lk(A, snow)$ 10
12 $\sim lk(B, snow)$
$\sim sk(x_2) \vee lk(x_2, snow)$ 5
13 $\sim sk(B)$
$\sim member(x_1) \vee mc(x_1) \vee sk(x_1)$ 4
14 $\sim member(B) \vee mc(B)$
$member(B)$ 2
11 $\sim member(x_6) \vee \sim mc(x_6) \vee sk(x_6)$
$mc(B)$ 15
16 $\sim member(B) \vee sk(B)$
$\sim sk(B)$ 13
17 $\sim member(B)$
$member(B)$ 2

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

Sentence

Subject | Predicate

$$\underset{\text{Subject}}{\text{Grass}}\quad \underbrace{\text{is}\quad\text{green}}_{\text{Predicate}}$$

green(grass)

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

D : Domain

Predicate Calculus: only for declarative sentences

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

Declarative Sentence

Subject

Predicate

- 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 $\propto$ 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)