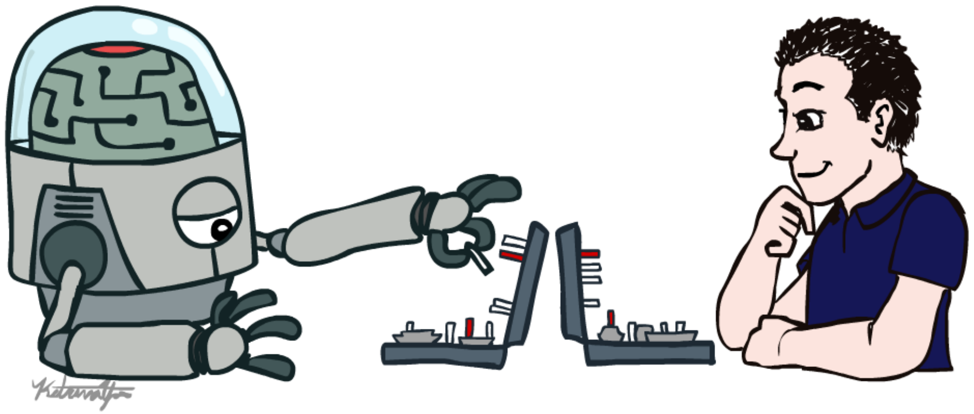
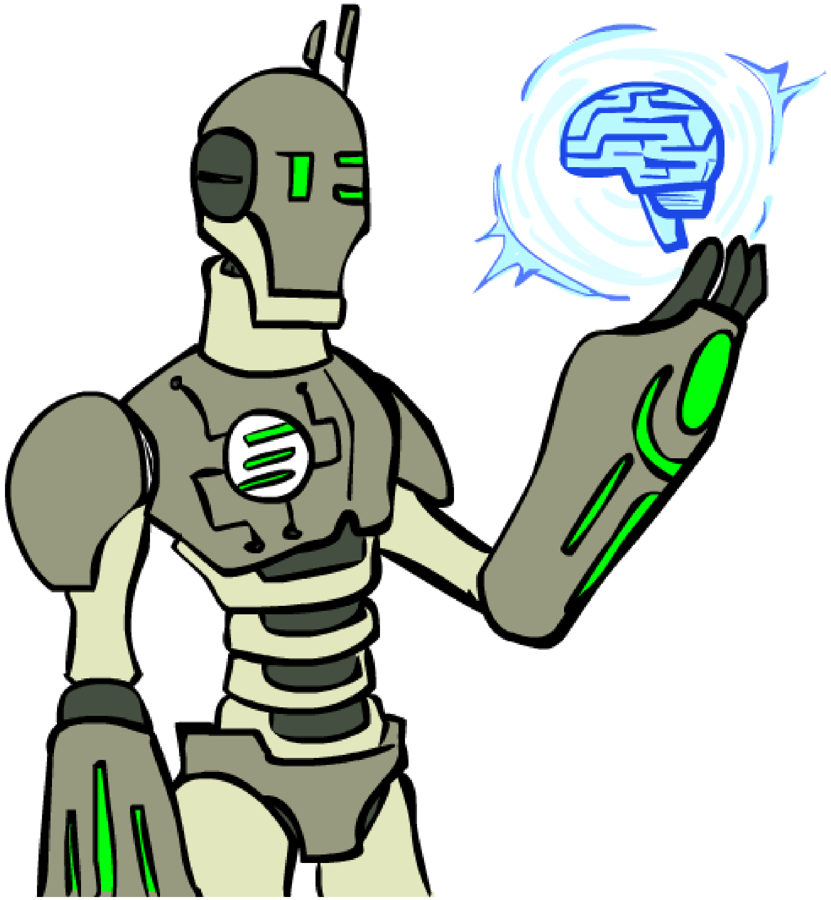
Lecture 07



Ashis Kumar Chanda

chanda@rowan.edu

Today



- Logical implication
  - Forward chaining
  - Backward chaining
- Resolution
- Unification

Clause

- A literal is a propositional symbol P or its negation ~p.
- In logic, a **clause** is an expression formed from a finite collection of literals.
- A clause is true either whenever at least one of the literals that form it is true (a disjunctive clause).
- Or when all of the literals that form it are true (a conjunctive clause).
- $P \vee Q \vee R$ (a disjunctive clause)
- $P \wedge Q \wedge R$ (a conjunctive clause)

Horn Clause

- A Horn Clause is a clause with at most one positive literals.

1. $A \leftarrow B_{1,}\ B_{2,}B_{3,}\ldots..B_m$ (True if all B's are true)

- A program clause is a first type horn clause.
- A logic program is a finite set of program clause.
- Programming language Prolog is built on top of Horn clauses.

Logical Implication

- A proposition P is said to be logically implied by the set of propositions S provided that P is true whenever S is true.

P is Logical consequence of S

Logical implication is proved by

1. Inference
2. Refutation

Inference

- Inference can be implied by two ways.
  - Forward chaining
  - Backward chaining

Forward chaining

- **Forward chaining**: Logical consequence leads us to a goal.
- An inference engine using **forward chaining** searches the inference rules until it finds one where the antecedent (If clause) is known to be true.

- There is rain.
- If there is rain, the road is wet.

- Can we say, the road is wet? **Yes**

Forward chaining

p
q
r
W ← p, r
v ← w, q, s
s ←w

Is v is a logical consequence of this set of formula?

p
w ← p, r
r
w ← r
w

w
s ←w
s

w
v ← w, q, s
q
v ← q, s
s
v ← s
v (Yes)

Quiz: Forward chaining

- If A and B are true for the following logical relations, prove that D is also true.
- $A \wedge C \rightarrow F$
- $A \wedge E \rightarrow G$
- $B \rightarrow E$
- $G \rightarrow D$

Backward chaining

- Backward chaining: Moves from goal to logical consequence to find truth.
- For example, suppose a new pet, Fritz, is delivered in an opaque box along with two facts about Fritz:
  - Fritz croaks
  - Fritz eats flies
- The goal is to decide whether Fritz is green?
- The following four rules:
  1. If X croaks and X eats flies – Then X is a frog
  2. If X chirps and X sings – Then X is a canary
  3. If X is a frog – Then X is green
  4. If X is a canary – Then X is yellow
- **Fritz is substituted for X in rule #3 to see if its consequent matches the goal**

Quiz: Backward chaining

- If A, B, C and E are true for the following logical relations, prove that Z is also true.
- $F \land B \rightarrow Z$
- $C \land D \rightarrow F$
- $A \rightarrow D$

Look, E is not used for any logical implication

Quiz: Forward chaining

- Can we apply forward chaining for the same problem?

- If A, B, C and E are true for the following logical relations, prove that Z is also true.
- $F \land B \rightarrow Z$
- $C \land D \rightarrow F$
- $A \rightarrow D$

Refutation

- Proving a statement to be wrong or false.
- We use resolution for this purpose.
- **Resolution** is a theorem proving technique that proofs by contradictions.
- It works this way –
  - select two clauses that contain conflicting terms
  - combine those two clauses and
  - cancel out the conflicting terms.
- **Unification** is a key concept in proof by contradiction.

Example: Resolution

- If it is a sunny & warm day, you will enjoy.
- If it is raining, you will get wet.
- It is a warm day.
- It is raining.
- It is sunny.
- Goal: you will enjoy.

Example: Resolution

Resolution steps:

- Convert facts into FOL.
- Convert FOL into Conjunctive Normal Form (CNF).
- Negate the statement to be proved.
- Draw resolution graph.

- CNF: $A \rightarrow B = \neg A \vee B$ [Lecture 6]

Example: Resolution

- If it is a sunny & warm day, you will enjoy.
- If it is raining, you will get wet.
- It is a warm day.
- It is raining.
- It is sunny.
  - $S \land W \rightarrow E$
  - $R \rightarrow T$
  - W
  - R
  - S

Apply CNF, draw resolution graph

Unification

- The process of making expressions look identical.
- We need to do substitution to make identical expression.

- P (x, y)   P(a, b)

- Unification: [a/x, b/y] , we say it as ‘a’ is replaceable by x

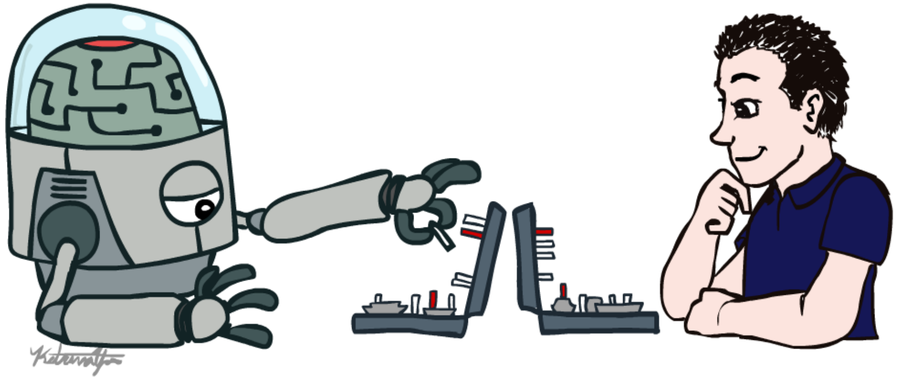
Unification

- Rules:
- Predicate symbol must be same.
- Ex: P(x, y)   P(a, b)   **Q**(r,s)
- Number of arguments in both expressions must be identical.
- Ex: P(x,y)   P(a, b, c)
- Unification will fail if there are two similar variable present in same expression.
- Ex: P (x, y, x)   P(a, b, c)

Unification

- Example:
  - P (x, f(y))   P(a, f(g(z)))
- Unification: [a/x,   g(z)/y]
- Example:
  - Q(a , g(x, c), f(y))   Q(a, g(f(b), c), x)]
- Unification: [a/a,   f(b)/x, c/c, b/y]

Let’s see some logical implication examples in Prolog



Thanks!