Elective Course
Course Code: CS4103
Autumn 2025-26

Lecture #26

Artificial Intelligence for Data Science

Week-7:

Introduction to Knowledge Representation and Logic [Part-V]

First Order Logic (Reasoning)

Course Instructor:

Dr. Monidipa Das
Assistant Professor
Department of Computational and Data Sciences
Indian Institute of Science Education and Research Kolkata, India 741246

Variable Substitutions

- **Variables** in the sentences can be substituted with **terms**.
- **Substitution lists** are the means used to track the value (or binding) of variables as processing proceeds.

$$\{\text{var}/\textit{term}, \text{var}/\textit{term}, \text{var}/\textit{term}...\}$$

– Application of the substitution to sentences

$$SUBST(\{x/Sam, y/Pam\}, Likes(x,y)) = Likes(Sam, Pam)$$

$$SUBST(\{x/z, y/fatherof(John)\}, Likes(x,y)) = Likes(z, fatherof(John))$$

More Examples

3

$$Cat(Felix)$$
$$\forall x Cat(x) \rightarrow Annoying(x)$$
$$\{x / Felix\}$$
$$Cat(Felix) \rightarrow Annoying(Felix)$$

$$\forall x, y Near(x, y) \rightarrow Near(y, x)$$
$$\{x/TRC, y/LHC\}$$
$$\mathrm{Near}(TRC, LHC) \rightarrow \mathrm{Near}(LHC, TRC)$$

Inference rules for quantifiers

4

Universal Elimination

If (∀x)P(x) is true, then P(c) is true, where c is a constant in the domain of x.

- (∀x)eats(Mary, x) → eats(Mary, IceCream).
- The variable symbol can be replaced by any ground term, i.e., any constant symbol or function symbol applied to ground terms only.

Existential Elimination

From (E_x)P(x) infer P(c).

- (E_x)eats(Mary, x) → eats(Mary, Cheese).
- Note that the variable is replaced by a brand new constant that does not occur in this or any other sentence in the Knowledge Base.

Existential Introduction

If P(c) is true, then (E_x)P(x) is inferred.

- eats(Mary, IceCream) →(E_x)eats(Mary, x).
- All instances of the given constant symbol are replaced by the new variable symbol. Note that the variable symbol cannot already exist anywhere in the expression.

Reasoning

5

- We can do all the same reasoning with FOL that we did with Propositional logic
  - **Forward Chaining**
  - **Backward Chaining**
  - **Resolution etc.**

- **But the presence of variables and quantifiers makes things more complicated**

FOL: Inference Methods

6

- Proof as Generic Search
- **Resolution**
- **Proof by Modus Ponens**
  - **Forward Chaining**
  - **Backward Chaining**

Resolution

7

Main procedure (steps):

1. Convert $KB, \neg \alpha$ to CNF with ground terms and universal variables only
2. Apply repeatedly the resolution rule while keeping track and consistency of substitutions
3. Stop when empty set (contradiction) is derived or no more new resolvents (conclusions) follow

Conversion to CNF

8

1. Eliminate implications, equivalences

$$(p \Rightarrow q) \rightarrow (\neg p \vee q)$$

2. Move negations inside (DeMorgan's Laws, double negation)

$$\neg(p \wedge q) \rightarrow \neg p \vee \neg q$$
$$\neg(p \vee q) \rightarrow \neg p \wedge \neg q$$
$$\neg \forall x\ p \rightarrow \exists x\ \neg p$$
$$\neg \exists x\ p \rightarrow \forall x\ \neg p$$
$$\neg\neg\ p \rightarrow p$$

3. Standardize variables (rename duplicate variables)

$$(\forall x\ P(x)) \vee (\exists x\ Q(x)) \rightarrow (\forall x\ P(x)) \vee (\exists y\ Q(y))$$

Conversion to CNF

9

4. Move all quantifiers left (no invalid capture possible)

$$(\forall x\ P(x)) \vee (\exists y\ Q(y)) \rightarrow \forall x\ \exists y\ P(x) \vee Q(y)$$

5. Skolemization (removal of existential quantifiers through elimination)

- If no universal quantifier occurs before the existential quantifier, replace the variable with a new constant symbol

$$\exists y\ P(A) \vee Q(y) \rightarrow P(A) \vee Q(B)$$

- If a universal quantifier precede the existential quantifier replace the variable with a function of the "universal" variable

$$\forall x\ \exists y\ P(x) \vee Q(y) \rightarrow \forall x\ \ P(x) \vee Q(F(x))$$

$F(x)$ - **a Skolem function**

Conversion to CNF

10

6. Drop universal quantifiers (all variables are universally quantified)

$$\forall x\ \ P(x) \vee Q(F(x)) \rightarrow P(x) \vee Q(F(x))$$

7. Convert to CNF using the distributive laws

$$p \vee (q \wedge r) \rightarrow (p \vee q) \wedge (p \vee r)$$

The result is a CNF with variables, constants, functions

Resolution Rule for FOL

11

- Given sentence $P1 \lor \ldots \lor Pn$ and sentence $Q1 \lor \ldots \lor Qm$ where each Pi and Qi is a literal, if Pj and $\sim Qk$ ***unify*** with substitution list σ, then derive the resolvent sentence as:

  $\boldsymbol{subst(\sigma, P1 \lor \cdots \lor Pj-1 \lor Pj+1 \lor \cdots \lor Pn \lor Q1 \lor \cdots \; Qk-1 \lor Qk+1 \lor \cdots \lor Qm)}$

- **Example**
  - From clause $P(x, f(a)) \lor P(x, f(y)) \lor Q(y)$ and
    clause $\sim P\big(z, f(a)\big) \lor \sim Q(z)$
    derive resolvent clause $P(z, f(y)) \lor Q(y) \lor \sim Q(z)$ using $\sigma = \{x/z\}$

Unification

12

- Unification is a "pattern matching" procedure that **takes two atomic sentences, as input**, and **returns "failure"** if they do not match or **a substitution list**, σ, if they do match.

$$UNIFY(p, q) = \sigma \text{ s.t. } SUBST(\sigma, p) = SUBST(\sigma, q)$$

- σ is called the most general unifier

Unification

13

- **Examples:**

$$UNIFY(Knows(John, x), Knows(John, Jane)) = \{x / Jane\}$$

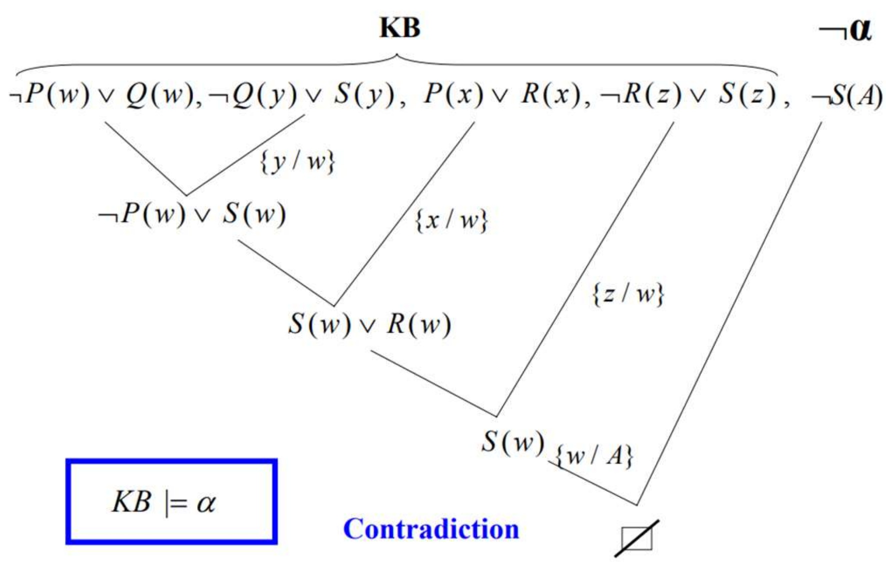
$$UNIFY(Knows(John, x), Knows(y, Ann)) = \{x / Ann, y / John\}$$

$$UNIFY(Knows(John, x), Knows(y, MotherOf(y))) = \{x / MotherOf(John), y / John\}$$

$$UNIFY(Knows(John, x), Knows(x, Elizabeth)) = fail$$

Resolution Example

14



Forward Chaining

15

- When a new **fact** p is added to the KB
  - For each **rule** such that p **unifies** with **part of the premise**
    - If all the other premises are **known**
    - Then **add consequent to the KB**

This is a **data-driven method**.

Forward chaining example: FOL

16

Assume the KB with the following rules:

KB: R1: $Steamboat\,(x) \wedge Sailboat\,(y) \Rightarrow Faster\,(x, y)$

R2: $Sailboat\,(y) \wedge RowBoat\,(z) \Rightarrow Faster\,(y, z)$

R3: $Faster\,(x, y) \wedge Faster\,(y, z) \Rightarrow Faster\,(x, z)$

F1: $Steamboat\,(Titanic\,)$

F2: $Sailboat\,(Mistral\,)$

F3: $RowBoat(PondArrow)$

Theorem: $Faster\,(Titanic\,, PondArrow\,)$?

Forward chaining example

17

KB: R1: $Steamboat\ (x) \wedge Sailboat\ (y) \Rightarrow Faster\ (x, y)$
R2: $Sailboat\ (y) \wedge RowBoat\ (z) \Rightarrow Faster\ (y, z)$
R3: $Faster\ (x, y) \wedge Faster\ (y, z) \Rightarrow Faster\ (x, z)$

F1: $Steamboat\ (Titanic\ )$
F2: $Sailboat\ (Mistral\ )$
F3: $RowBoat(PondArrow)$

Rule R1 is satisfied:

F4: $Faster(Titanic, Mistral)$ ⬅

Rule R2 is satisfied:

F5: $Faster(Mistral, PondArrow)$ ⬅

Rule R3 is satisfied:

F6: $Faster(Titanic, PondArrow)$ ⬅

Backward Chaining

18

- When a **query** q is asked
  - If a **matching** q' is found **return substitution list**
  - Else, For each **rule** q' whose **consequent matches** q, attempt to **prove each antecedent** by backward chaining

This is a **goal-directed method**. And it's the basis for **Prolog.**

Backward Chaining: Example

19

- **Backward chaining (goal reduction)**

  Idea: To prove the fact that appears in the conclusion of a rule prove the $\boxed{\text{antecedents (if part)}}$ of the rule & $\boxed{\text{repeat recursively.}}$

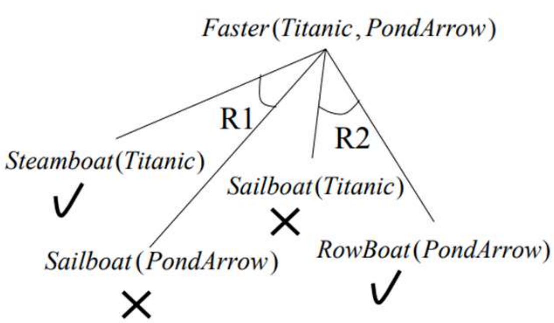
KB:
R1: $Steamboat\ (x) \wedge Sailboat\ (y) \Rightarrow Faster\ (x, y)$
R2: $Sailboat\ (y) \wedge RowBoat\ (z) \Rightarrow Faster\ (y, z)$
R3: $Faster\ (x, y) \wedge Faster\ (y, z) \Rightarrow Faster\ (x, z)$

F1: $Steamboat\ (Titanic\ )$
F2: $Sailboat\ (Mistral\ )$
F3: $RowBoat(PondArrow)$

Theorem: $Faster\ (Titanic\ , PondArrow\ )$

Backward Chaining: Example

20



KB:
R1: $Steamboat\ (x) \wedge Sailboat\ (y) \Rightarrow Faster\ (x, y)$
R2: $Sailboat\ (y) \wedge RowBoat\ (z) \Rightarrow Faster\ (y, z)$
R3: $Faster\ (x, y) \wedge Faster\ (y, z) \Rightarrow Faster\ (x, z)$

F1: $Steamboat\ (Titanic\ )$
F2: $Sailboat\ (Mistral\ )$
F3: $RowBoat(PondArrow)$

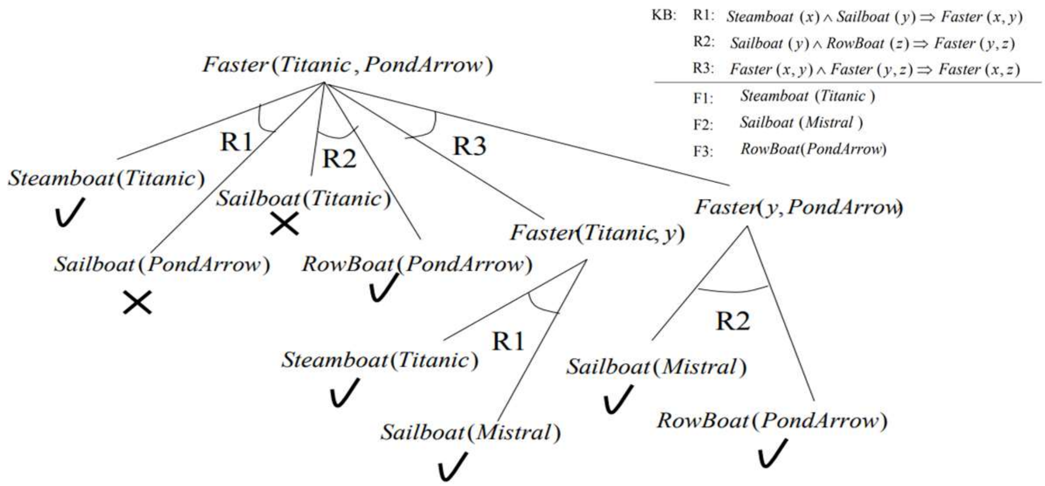
$$Sailboat\ (y) \wedge RowBoat\ (z) \Rightarrow Faster\ (y, z)$$

$$Faster\ (Titanic\ , PondArrow\ )$$

$$\{y / Titanic\ , z / PondArrow\}$$

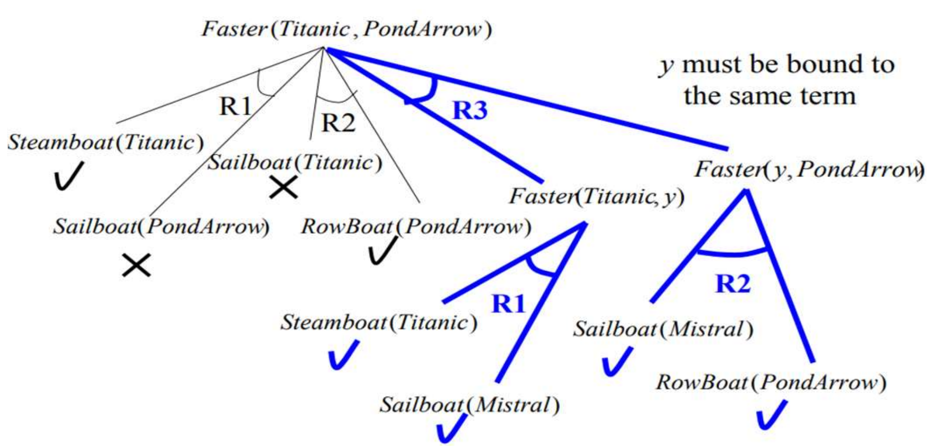
Backward Chaining: Example

21



Backward Chaining: Example

22



Forward vs. Backward Chaining

23

- FC is ***data-driven,*** …for automatic, unconscious processing
  - **May do lots of work that is irrelevant to the goal**

- BC is ***goal-driven***, appropriate for problem-solving,
  - **Complexity of BC can be much smaller than linear in size of KB**

Example: Rules of Inference for **Propositional Logic**

24

Show that the hypotheses:

- It is not sunny this afternoon and it is colder than yesterday.
- We will go swimming only if it is sunny.
- If we do not go swimming, then we will take a canoe trip.
- If we take a canoe trip, then we will be home by sunset.

lead to the conclusion:

- We will be home by the sunset.

Example [contd.]

25

Main steps:

1. Translate the statements into proposional logic.
2. Write a formal proof, a sequence of steps that state hypotheses or apply inference rules to previous steps.

Translation:

s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Example [contd.]

26

Show that the hypotheses:

- It is not sunny this afternoon and it is colder than yesterday.
- We will go swimming only if it is sunny.
- If we do not go swimming, then we will take a canoe trip.
- If we take a canoe trip, then we will be home by sunset.

lead to the conclusion:

- We will be home by the sunset.

s: "it is sunny this afternoon"
c: "it is colder than yesterday"
w: "we will go swimming"
t: "we will take a canoe trip.
h: "we will be home by the sunset."

Example [contd.]

27

Show that the hypotheses:

- It is not sunny this afternoon and it is colder than yesterday. $\neg s \land c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

lead to the conclusion:

- We will be home by the sunset. h

Step	Reason
1. $\neg s \land c$	hypothesis
2. $\neg s$	AND Elimination applied on 1
3. $w \rightarrow s$	hypothesis
4. $\neg w$	3. $\neg w \lor s$ 2. $\neg s$ Unit Resolution
5. $\neg w \rightarrow t$	hypothesis
6. t	modus ponens of 4 and 5
7. $t \rightarrow h$	hypothesis
8. h	modus ponens of 6 and 7

Same Example: **Resolution-based proof**

28

Show that the hypotheses:

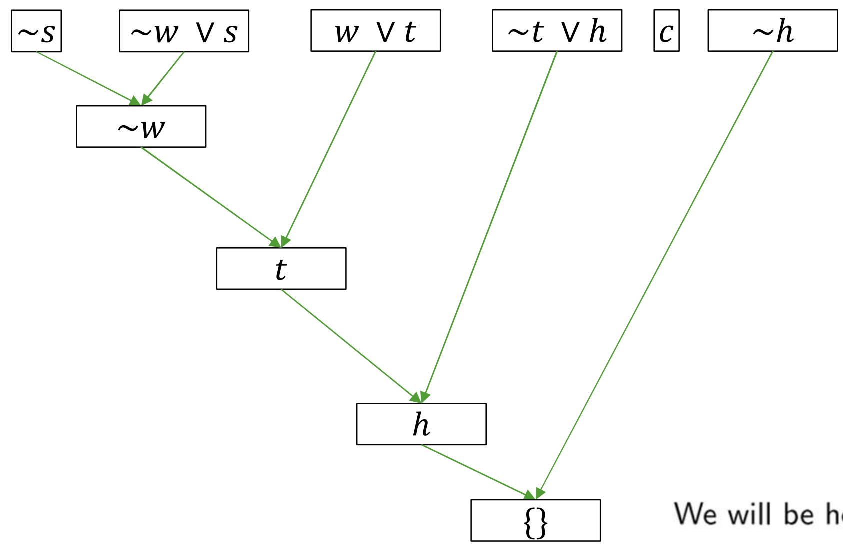
- It is not sunny this afternoon and it is colder than yesterday. $\neg s \land c$
- We will go swimming only if it is sunny. $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip. $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset. $t \rightarrow h$

lead to the conclusion:

- We will be home by the sunset. h

Same Example: **Resolution-based proof**

29



We will be home by the sunset. h

Propositional Logic: Homework

30

- **Which of the following are propositions and which are not?**
  a) It is raining today
  b) 5 + 2 = 8
  c) How are you?
  d) 2 is a prime number

Propositional Logic: Homework

31

- Let
  - p = It is raining
  - q = Mary is sick
  - t = Bob stayed up late last night
  - r = Paris is the capital of France
  - s = John is a loud-mouth

Now, translate the following:

1) John is a loud-mouth but Mary isn't sick
2) If it is raining, then Mary is sick
3) Mary is sick and it is raining implies that Bob stayed up late last night
4) It is raining if and only if Mary is sick
5) It is raining is not equivalent to John is a loud-mouth

Propositional Logic: Homework

32

- Given the following knowledge base (KB)

$$\textbf{KB: } P \wedge Q \quad P \Rightarrow R \quad (Q \wedge R) \Rightarrow S$$

- Prove the following theorem
  - **a. using inference rules (inference by proof approach)**
  - **b. using resolution technique**

$$\textbf{Theorem: } S$$

FOL: Homework

33

- **[Homework]** Convert the following sentence to clause form
  $(\forall x)(P(x) \;=>\; ((\forall y)(P(y) \;=>\; P(f(x,y))) \land \sim(\forall y)(Q(x,y) \;=>\; P(y))))$

- **[Homework]** Consider the following axioms:
  1. All hounds howl at night.
  2. Anyone who has any cat will not have any mice.
  3. Light sleepers do not have anything which howls at night.
  4. John has either a cat or a hound.
  5. (Conclusion) If John is a light sleeper, then John does not have any mice.

  Now, prove the conclusion using Resolution

34

Questions?

Slide Content taken from Prof. Cesare Tinelli, Prof. Stuart Russell, and Prof. Jim Martin