Elective Course
Course Code: CS4103
Autumn 2025-26

Lecture #25

Artificial Intelligence for Data Science

Week-7:

Introduction to Knowledge Representation and Logic [Part-IV]

Propositional Logic (Backward Chaining)
First Order Logic (Introduction)

Course Instructor:

Dr. Monidipa Das

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

Backward chaining (BC)

2

- **Idea:** work backwards from the query q to infer q by BC, check if q is known already, or infer by BC all premises of some rule concluding q
- **Avoid loops:** check if new subgoal is already on the goal stack
- **Avoid repeated work:** check if new subgoal
  - 1. has already been inferred, or
  - 2. has already failed

Backward chaining example

KB is

$P \implies Q$

$L \wedge M \implies P$

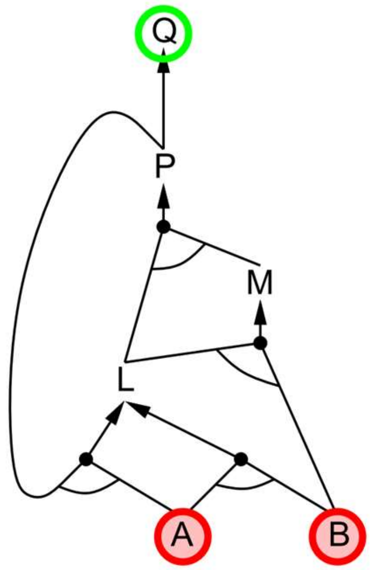
$B \wedge L \implies M$

$A \wedge P \implies L$

$A \wedge B \implies L$

A

B



Backward chaining example

KB is

$P \implies Q$

$L \wedge M \implies P$

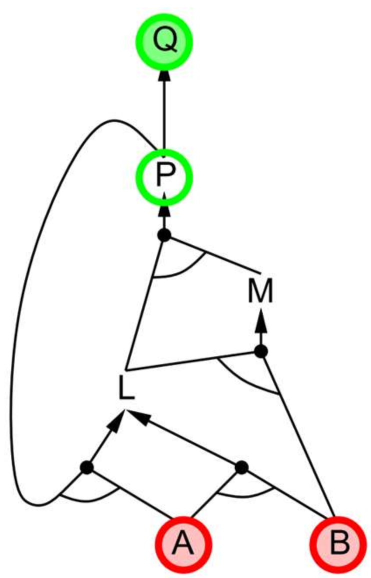
$B \wedge L \implies M$

$A \wedge P \implies L$

$A \wedge B \implies L$

A

B



Backward chaining example

KB is

$P \implies Q$

$L \wedge M \implies P$

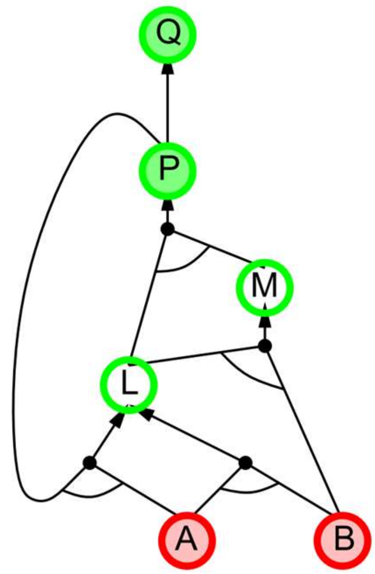
$B \wedge L \implies M$

$A \wedge P \implies L$

$A \wedge B \implies L$

A

B



Backward chaining example

KB is

$P \implies Q$

$L \wedge M \implies P$

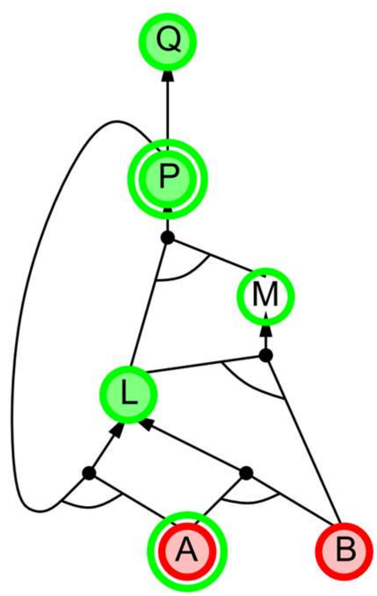
$B \wedge L \implies M$

$A \wedge P \implies L$

$A \wedge B \implies L$

A

B



Backward chaining example

7

KB is

$P \implies Q$

$L \wedge M \implies P$

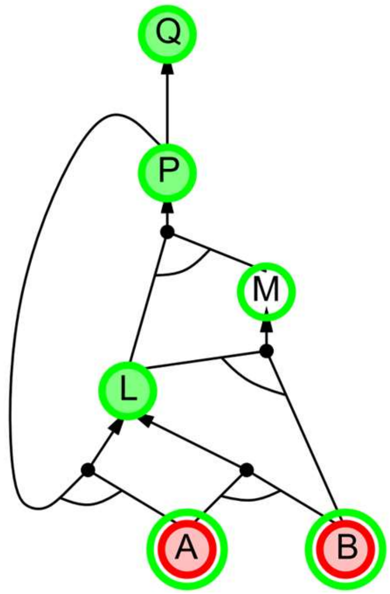
$B \wedge L \implies M$

$A \wedge P \implies L$

$A \wedge B \implies L$

A

B



Dr. Monidipa Das, Department of CDS, IISER Kolkata

Backward chaining example

8

KB is

$P \implies Q$

$L \wedge M \implies P$

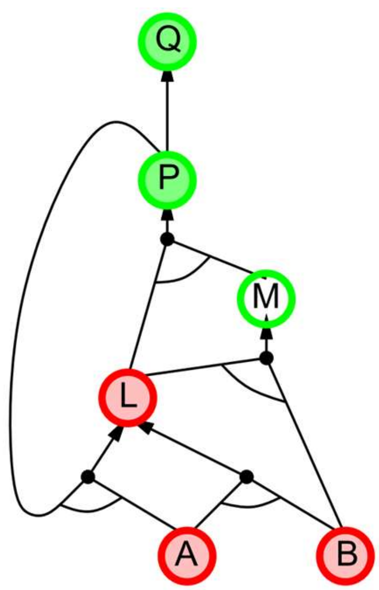
$B \wedge L \implies M$

$A \wedge P \implies L$

$A \wedge B \implies L$

A

B



Dr. Monidipa Das, Department of CDS, IISER Kolkata

Backward chaining example

9

KB is

$P \implies Q$

$L \wedge M \implies P$

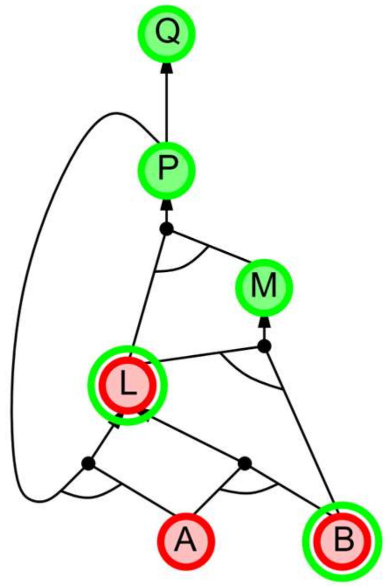
$B \wedge L \implies M$

$A \wedge P \implies L$

$A \wedge B \implies L$

A

B



Backward chaining example

10

KB is

$P \implies Q$

$L \wedge M \implies P$

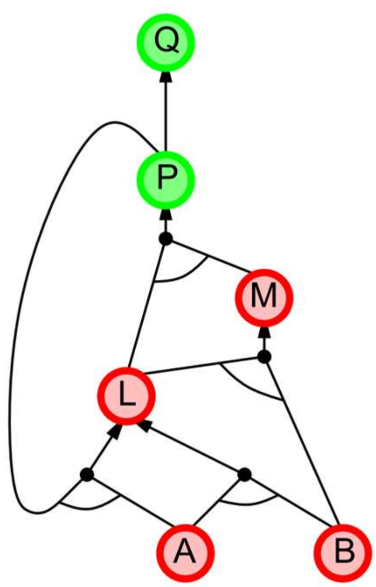
$B \wedge L \implies M$

$A \wedge P \implies L$

$A \wedge B \implies L$

A

B



Backward chaining example

KB is

$P \implies Q$

$L \wedge M \implies P$

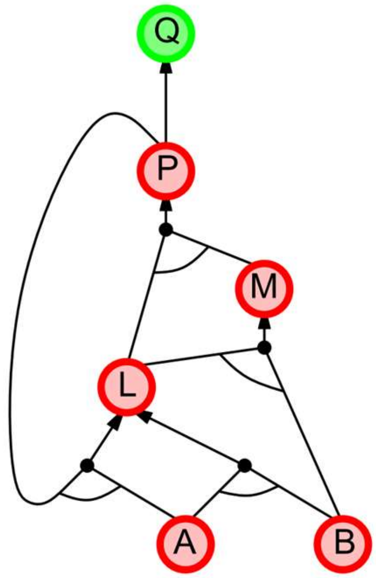
$B \wedge L \implies M$

$A \wedge P \implies L$

$A \wedge B \implies L$

A

B



Backward chaining example

KB is

$P \implies Q$

$L \wedge M \implies P$

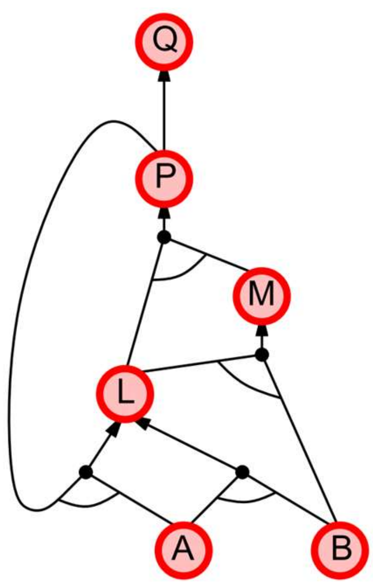
$B \wedge L \implies M$

$A \wedge P \implies L$

$A \wedge B \implies L$

A

B



First-order logic

13

- First-order logic (like natural language) assumes the world contains
  - Objects: people, houses, numbers, colors, baseball games, wars, …
  - Relations: red, round, prime, brother of, bigger than, part of, comes between, …
  - Functions: father of, best friend, one more than, plus, …

Syntax of FOL

14

• Constants	John, Richard
• Predicates	Brother, Near, Likes,...
• Functions	Sqrt, LeftLegOf,...
• Variables	x, y, a, b,...
• Connectives	$\neg, \Rightarrow, \wedge, \vee, \Leftrightarrow$
• Equality	=
• Quantifiers	$\forall, \exists$

Atomic sentences

15

Atomic sentence = *predicate* ($term_1$,...,$term_n$)
or $term_1 = term_2$

Term = *function* ($term_1$,...,$term_n$)
or *constant* or *variable*

- E.g.,
  - *Brother(John, Richard)*
  - *GreaterThan (Length(LeftLegOf(Richard)), Length(LeftLegOf(John)))*

Complex sentences

16

- Complex sentences are made from atomic sentences using connectives

$$\neg S,\ S_1 \wedge S_2,\ S_1 \vee S_2,\ S_1 \Rightarrow S_2,\ S_1 \Leftrightarrow S_2,$$

E.g.

Sibling(John, Richard) $\Rightarrow$ *Sibling(Richard, John)*

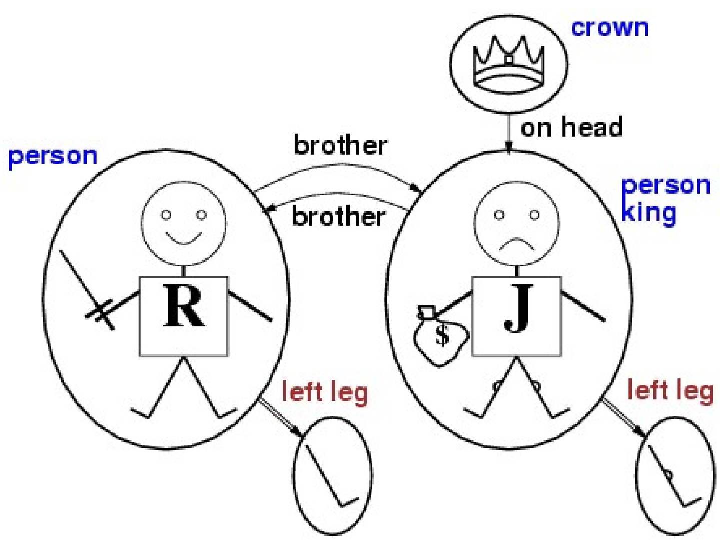
Truth in first-order logic

17

- Sentences are true with respect to a model and an interpretation
- Models contain objects (domain elements) and relations among them
- Interpretation specifies referents for
  constant symbols → objects
  predicate symbols → relations
  function symbols → functional relations
- An atomic sentence *predicate(term*$_1$*,...,term*$_n$*)* is true iff the objects referred to by *term*$_1$*,...,term*$_n$ are in the relation referred to by *predicate*.

Models for FOL: Example

18



Models as Sets

19

- Let's populate a domain:
  - {R, J, RLL, JLL, C}

a set of objects

- Property Predicates
  - Person = {R, J}
  - Crown = {C}
  - King = {J}
- Relational Predicates
  - Brother = { <R,J>, <J,R>}
  - OnHead = {<C,J>}
- Functional Predicates
  - LeftLegOf = {<R, RLL>, <J, JLL>}

Quantifiers

20

- Allow us to express properties of collections of objects instead of enumerating objects by name
- **Universal:** "for all" ∀
- **Existential:** "there exists" ∃

Universal quantification

21

$\forall$<*variables*> <*sentence*>

Everyone at IISERK is smart:

$\forall$**x At(x, IISERK) $\Rightarrow$ Smart(x)**

$\forall$x *P* is true in a model *m* iff *P* is true with *x* being each possible object in the model

Roughly speaking, equivalent to the conjunction of instantiations of *P*

At(John, IISERK) $\Rightarrow$ Smart(John)
$\wedge$ At(Simon, IISERK) $\Rightarrow$ Smart(Simon)
$\wedge$ At(Alan, IISERK) $\Rightarrow$ Smart(Alan)
$\wedge$ ...

Existential quantification

22

$\exists$<*variables*> <*sentence*>

Someone at IISERK is smart:

$\exists$***x*** **At(x, IISERK) $\wedge$ Smart(x)**

$\exists$*x P* is true in a model *m* iff *P* is true with *x* being some possible object in the model

- Roughly speaking, equivalent to the disjunction of instantiations of *P*

At(John, IISERK) $\wedge$ Smart(John)
$\vee$ At(Simon, IISERK) $\wedge$ Smart(Simon)
$\vee$ At(Alan, IISERK) $\wedge$ Smart(Alan)
$\vee$ ...

Properties of quantifiers

23

∀x ∀y is the same as ∀y ∀x
∃x ∃y is the same as ∃y ∃x

∃x ∀y is **not** the same as ∀y ∃x
∃x ∀y Loves(x,y)
– "There is a person who loves everyone in the world"
∀y ∃x Loves(x,y)
– "Everyone in the world is loved by at least one person"

- **Quantifier duality:** each can be expressed using the other
  ∀x Likes(x,IceCream)   ¬∃x ¬Likes(x,IceCream)
  ∃x Likes(x,Broccoli)   ¬∀x ¬Likes(x,Broccoli)

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Sentences with variables

24

First-order logic sentences can include variables.

- **Variable** is:
  - **Bound** – if it is in the scope of some quantifier
    $\forall x \; P(x)$
  - **Free** – if it is not bound.
    $\exists x \; P(y) \wedge Q(x)$ y is free
- **Sentence** (formula) is:
  - **Closed** – if it has no free variables
    $\forall y \exists x \; P(y) \Rightarrow Q(x)$
  - **Open** – if it is not closed
  - **Ground** – if it does not have any variables
    $Likes(John, Jane)$

Dr. Monidipa Das, Department of CDS, IISER Kolkata

FOL Translation: Example

25

- **Syntax**
  - Constants: *gary, heidi, herschel, orfy, willy*
  - Predicates: *Person(x), Pet(x), Dog(x), Cat(x), Chases(x, y), LargerThan(x, y), Loves(x, y)*
- **Semantics**

heidi	Heidi
orfy	Orfy
herschel	Herschel
willy	Willy
gary	Gary
Person(x)	x is a person
Pet(x)	x is a pet
Dog(x)	x is a dog
Cat(x)	x is a cat
Chases(x, y)	x chases y
Larger(x, y)	x is larger than y
Loves(x, y)	x loves y

FOL Translation: Example

26

Constants: *gary, heidi, herschel, orfy, willy*
Predicates: *Person(x), Pet(x), Dog(x), Cat(x), Chases(x, y), LargerThan(x, y), Loves(x, y)*

- Herschel is a cat and Orfy is a cat
  - $Cat(herschel) \wedge Cat(orfy)$
- Not both Orfy and Willy are dogs.
  - $\neg(Dog(orfy) \wedge Dog(willy))$
- Some cats are pets.
  - $\exists x\ (Pet(x) \wedge Cat(x))$
- Some pets are dogs.
  - $\exists x\ (Pet(x) \wedge Dog(x))$
- Every pet is either a dog or a cat.
  - $\forall x\ (Pet(x) \rightarrow (Dog(x) \vee Cat(x)))$
- No dog is a cat.
  - $\forall x\ (Dog(x) \rightarrow \neg Cat(x))$

FOL Translation: Example

27

Constants: *gary, heidi, herschel, orfy, willy*

Predicates: *Person(x), Pet(x), Dog(x), Cat(x), Chases(x, y), LargerThan(x, y), Loves(x, y)*

- Willy is larger than every cat.
  - $\forall x\ (Cat(x) \rightarrow LargerThan(willy, x))$
- Every dog chases some cat (or other).
  - $\forall x\ (Dog(x) \rightarrow \exists y\ (Cat(y) \wedge Chases(x, y)))$
- Gary loves any dog who is larger than every cat.
  - $\forall x\ ((Dog(x) \wedge \forall y\ (Cat(y) \rightarrow LargerThan(x, y))) \rightarrow Loves(gary, x))$
- There is at most one dog larger than Heidi.
  - $\forall x \forall y ((Dog(x) \wedge LargerThan(x, heidi) \wedge Dog(y) \wedge LargerThan(y, heidi)) \rightarrow x=y)$
- Willy is the largest dog.
  - $Dog(willy) \wedge \forall y\ ((Dog(y) \wedge \neg(y=willy)) \rightarrow LargerThan(willy, y))$

28

Questions?

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