FIRST ORDER LOGIC

Fundamentals of Artificial Intelligence

Session 19

Pramod Sharma
pramod.sharma@prasami.com

2

Agenda

- Logic – Continues…
- Inference and Resolution for Problem Solving
- Propositional Logic
- First Order Logic

pra-sâmi

3

Beyond Propositional Logic

- In propositional logic, we can only represent the facts
- Facts which are either true or false
- Not sufficient to represent the complex sentences or natural language statements
  - Example: Every rose has a thorn
- Has very limited expressive power
- So we use First Order Logic
  - First-order logic is also known as Predicate logic or First-order predicate logic

pra-sâmi

4

First Order Logic

- More expressive logic than Propositional Logic
- Sufficiently expressive to represent the natural language statements in a concise way
- Constants are objects: john, apples
- Predicates are properties and relations:
  - likes(john, apples)
- Functions transform objects:
  - likes(john, fruit_of(apple_tree))
- Variables represent any object: likes(X, apples)

pra-sâmi

5

First-order logic

- First-order logic (FOL) models the world in terms of
  - Objects, which are things with individual identities
  - Properties of objects that distinguish them from other objects
  - Relations that hold among sets of objects
  - Functions, which are a subset of relations where there is only one “value” for any given “input”

- Examples:
  - Objects: Students, lectures, companies, cars ...
  - Relations: Brother-of, father-of, outside, part-of, has-color, occurs-after, owns, visits, precedes, ...
  - Properties: blue, oval, even, large, ...
  - Functions: best-friend, second-half, bigger-than, one-more-than ...

6

First Order Logic

- An extension of Propositional logic
  - All connectives of Propositional logic are still applicable

- FOL has extra symbols ∀ (all) and ∃ (there exists)

- These are called quantifiers
  - Universal Quantifier – True for all objects : ∀ (all); ∀ X. likes (X, apples)
  - Existential Quantifier – Exists at least one object : ∃ (there exists); ∃ X. likes (X, apples)

- Allows structure to be represented
  - Proposition Logic : p = “My car is pearl white”
  - Predicate (FOL) Logic: p = color (Car, pearl white)

Predicate or relation

Arguments or objects

  - Mohan likes apples ➔ likes (Mohan, Apples)

7

Sentences are Built from Terms and Atoms

- A term (denoting a real-world individual) is a constant symbol, a variable symbol, or an n-place function of n terms
  - x and $x_1, x_2 \ldots, x_n$ are terms,
  - A term with no variables is a ground term
- An atomic sentence (which has value true or false) is an n-place predicate of n terms
- A complex sentence is formed from atomic sentences connected by the logical connectives:
  - $\neg$P, P$\vee$Q, P$\wedge$Q, P$\rightarrow$Q, P$\leftrightarrow$Q where P and Q are sentences
- A quantified sentence adds quantifiers $\forall$ and $\exists$
- A well-formed formula (wff) is a sentence containing no "free" variables
  - That is, all variables are "bound" by universal or existential quantifiers
  - ($\forall$x) P (x , y) has x bound as a universally quantified variable, but y is free

4/16/2024

pra-sâmi

8

Sentences

- Atomic Sentences
  - The most basic sentences of first-order logic
  - These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms
  - Represented as Predicate ($term_1, term_2, \ldots \ldots, term_n$).
  - Example:
    - Ravi and Vijay are brothers: ➔ Brothers (Ravi, Vijay)
    - Chinky is a cat: ➔ cat (Chinky)
- Complex Sentences
  - Complex sentences are made by combining atomic sentences using connectives.

4/16/2024

pra-sâmi

9

Quantifiers

- Universal quantifiers are often used with "implies" to form "rules":
  - $(\forall x)$ student(x) $\rightarrow$ smart(x) means "All students are smart"
- Universal quantification is rarely used to make blanket statements about every individual in the world:
  - $(\forall x)$student(x)$\wedge$smart(x) means "Everyone in the world is a student and is smart"
- Existential quantifiers are usually used with "and" to specify a list of properties about an individual:
  - $(\exists x)$ student(x) $\wedge$ smart(x) means "There is a student who is smart"
- A common mistake is to represent this English sentence as the FOL sentence:
  - $(\exists x)$ student(x) $\rightarrow$ smart(x)
  - But what happens when there is a smart person who is not a student?

4/16/2024

pra-sâmi

10

Quantifier Scope

- Switching the order of universal quantifiers does not change the meaning:
  - $(\forall x)$ $(\forall y)$ P(x, y) $\leftrightarrow$ $(\forall y)$ $(\forall x)$ P(x, y)
- Similarly, you can switch the order of existential quantifiers:
  - $(\exists x)$ $(\exists y$ P(x, y) $\leftrightarrow$ $(\exists y)$ $(\exists x)$ P(x, y)
- Switching the order of universals and existentials **change** meaning:
  - Everyone likes someone: $(\forall x)(\exists y)$ likes(x, y)
  - Someone is liked by everyone: $(\exists y)(\forall x)$ likes(x, y)
- We can relate sentences involving $\forall$ and $\exists$ using De Morgan's laws

4/16/2024

pra-sâmi

11

Difference between Propositional Logic and Predicate Logic

Propositional Logic	First Order Logic (Predicate Logic)
Propositional logic is the logic that deals with a collection of declarative statements which have a truth value, true or false.	Predicate logic is an expression consisting of variables with a specified domain. It consists of objects, relations and functions between the objects.
It is the basic and most widely used logic. Also known as Boolean logic.	It is an extension of propositional logic covering predicates and quantification.
A proposition has a specific truth value, either true or false.	A predicate's truth value depends on the variables' value.
Scope analysis is not done in propositional logic.	Predicate logic helps analyze the scope of the subject over the predicate. There are three quantifiers : Universal Quantifier (∀) depicts for all, Existential Quantifier (∃) depicting there exists some and Uniqueness Quantifier (∃!) depicting exactly one.
Propositions are combined with Logical Operators or Logical Connectives like Negation(¬), Conjunction(∧), Disjunction(∨), Exclusive OR(⊕), Implication(⇒), Bi-Conditional or Double Implication(⇔).	Predicate Logic adds by introducing quantifiers to the existing proposition.
It is a more generalized representation.	It is a more specialized representation.
It cannot deal with sets of entities.	It can deal with set of entities with the help of quantifiers.

4/16/2024

pra-sâmi

12

Quantified Inference Rules

- Universal instantiation
  - $\forall x\ P(x)\ \therefore\ P(A)$
- Universal generalization
  - $P(A) \wedge P(B) \ldots \therefore\ \forall x\ P(x)$
- Existential instantiation
  - $\exists x\ P(x)\ \therefore P(F)$ [skolem constant F]
- Existential generalization
  - $P(A)\ \therefore\ \exists x\ P(x)$

4/16/2024

pra-sâmi

13 Universal Instantiation (aka Universal Elimination)

- If $(\forall x)\ P(x)$ is true, then P(C) is true, where C is any constant in the domain of x
- Example:
  - $(\forall x)$ eats(Ziggy, x) $\Rightarrow$ eats(Ziggy, IceCream)
- Note that the variable is replaced by a brand-new constant not occurring in this or any other sentence in the KB
- The variable symbol can be replaced by any ground term, i.e., any constant symbol or function symbol applied to ground terms only

4/16/2024 pra-sâmi

14 Existential Instantiation (aka Existential Elimination)

- From $(\exists x)\ P(x)$ infer P(c)
- Example:
  - $(\exists x)$ eats(Ziggy, x) $\rightarrow$ eats(Ziggy, Stuff)
  - Note that the variable is replaced by a brand-new constant not occurring in this or any other sentence in the KB
- Also known as skolemization; constant is a skolem constant
- In other words, we don't want to accidentally draw other inferences about it
- Convenient to use this to reason about the unknown object, rather than constantly manipulating the existential quantifier

4/16/2024 pra-sâmi

15

Existential Generalization (aka Existential Introduction)

- If P(c) is true, then (∃x) P(x) is inferred
- Example:
  - eats(Ziggy, IceCream) ⇒ (∃x) eats(Ziggy, 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

4/16/2024

pra-sâmi

16

Semantics of FOL

- Domain M: the set of all objects in the world (of interest)
- Interpretation includes
  - Assign each constant to an object in M
  - Define each function of n arguments as a mapping M(n) ⇒M
  - Define each predicate of n arguments as a mapping M(n) ⇒{T, F}
  - Therefore, every ground predicate with any instantiation will have a truth value
  - In general there is an infinite number of interpretations because |M| is infinite

4/16/2024

pra-sâmi

Semantics of FOL

- Model: an interpretation of a set of sentences such that every sentence is True
- A sentence is
  - Satisfiable if it is true under some interpretation
  - Valid if it is true under all possible interpretations
  - Inconsistent if there does not exist any interpretation under which the sentence is true
- Logical consequence: S ⊨ X if all models of S are also models of X

Example: FOL

- All humans are mortals
  - $\forall$ x human(x) $\Rightarrow$ mortal(x)
  - For all x , if x is human then x is mortal
- Some apples are sweet
  - $\forall$ x $\exists$ y sweet(x, y)
  - Among all x , there exist a y which is sweet
- Everybody loves somebody
  - $\forall$ x $\exists$ y loves(x, y)
  - Among all x , there exist a y loved by x

19

Example: FOL

- "Every rose has a thorn"
  - ∀ X . (Rose(X) → ∃ Y . (has (X, Y) ∧ Thorn (Y)))
  - For all X, if (X is a rose), then there exists Y, (X has Y) and (Y is a thorn)

- One's mother is one's female parent
  - ∀ (m, c) . (Mother (c) = m ⇔ Female(m) ∧ Parent(m, c))
  - For all m and c, if m is mother of c, Then m is Female and m is Parent of c and Vice versa

- "On Mondays and Wednesdays I go to John's house for dinner"
  - ∀ X . ((isMon (X) ∨ isWed (X)) → eatMeal (me, houseOf (john), X))
  - For all days, if it's a Monday "or" Wednesday, I eat meal at John's house

Note the change from "and" to "or"
Translating is problematic

4/16/2024

pra-sâmi

20

Example: FOL

- Parent and child are inverse relations:
  - ∀ (p, c).(Parent(p, c) ⇔ Child(c, p))
  - For all p & c, if p is parent of c, then c is child of p

- A grandparent is a parent of one's parent:
  - ∀ (g, c). (Grandparent(g, c) ⇔ ∃ p Parent(g, p) ∧ Parent(p, c))
  - For all g & c, if g is grand parent of c, then there exists p such that p is parent of c and g is parent of p

- A sibling is another child of one's parents:
  - ∀ (x, y).(Sibling(x, y) ⇔ x = y ∧ ∃ p Parent(p, x) ∧ Parent(p, y))
  - For all x & y, if x and y are sibling, then there exist a p who is parent of x and parent of y

4/16/2024

pra-sâmi

21

Example: FOL

- Consider the following knowledge:
  - Bob is Fred's father $\Rightarrow$ father(Bob, Fred)
  - Sue is Fred's mother $\Rightarrow$ mother(Sue, Fred)
  - Barbara is Fred's sister $\Rightarrow$ sister(Barbara, Fred)
  - Jerry is Bob's father $\Rightarrow$ father(Jerry, Bob)
- And the following rules:
  - A person's father's father is the person's grandfather
  - A person's father or mother is that person's parent
  - A person's sister or brother is that person's sibling
  - If a person has a parent and a sibling, then the sibling has the same parent
- These might be captured in first-order predicate calculus as:
  - $\forall$ x, y, z : if father(x, y) and father(y, z) then grandfather(x, z)
  - $\forall$ x, y : if father(x, y) or mother(x, y) then parent(x, y)
  - $\forall$ x, y : if sister(x, y) or brother(x, y) then sibling(x, y) and sibling(y, x)
  - $\forall$ x, y, z : if parent(x, y) and sibling(y, z) then parent(x, z)
- We would rewrite these as
  - grandfather(x, z) $\Rightarrow$ father(x, y) $\land$ father(y, z)
  - parent(x, y) $\Rightarrow$ father(x, y)
  - parent(x, y) $\Rightarrow$ mother(x, y)… etc.

4/16/2024 pra-sâmi

22

Example: FOL

- Every gardener likes the sun.
  - $\forall$x gardener(x) $\rightarrow$ likes(x,Sun)
- You can fool some of the people all the time.
  - $\exists$x $\forall$t person(x) $\land$time(t) $\rightarrow$ can-fool(x,t)
- You can fool all of the people some of the time.
  - $\forall$x $\exists$t (person(x) $\rightarrow$ time(t) $\land$can-fool(x,t))
  - $\forall$x (person(x) $\rightarrow$ $\exists$t (time(t) $\land$can-fool(x,t))

Equivalent

- All purple mushrooms are poisonous.
  - $\forall$x (mushroom(x) $\land$ purple(x)) $\rightarrow$ poisonous(x)

4/16/2024 pra-sâmi

23

Example: FOL

- No purple mushroom is poisonous.
  - $\neg\exists x\ purple(x) \wedge mushroom(x) \wedge poisonous(x)$
  - $\forall x\ (mushroom(x) \wedge purple(x)) \rightarrow \neg poisonous(x)$

Equivalent

- There are exactly two purple mushrooms.
  - $\exists x\ \exists y\ mushroom(x) \wedge purple(x) \wedge mushroom(y) \wedge purple(y) \wedge \neg(x=y) \wedge \forall z\ (mushroom(z) \wedge purple(z)) \rightarrow ((x=z) \vee (y=z))$
- Clinton is not tall.
  - $\neg tall(Clinton)$
- X is above Y iff X is on directly on top of Y or there is a pile of one or more other objects directly on top of one another starting with X on top and ending with Y.
  - $\forall x\ \forall y\ above(x,y) \leftrightarrow (on(x,y) \vee \exists z\ (on(x,z) \wedge above(z,y)))$

4/16/2024 pra-sâmi

24

Example: A simple genealogy KB by FOL

- Build a small genealogy knowledge base using FOL that
  - contains facts of immediate family relations (spouses, parents, etc.)
  - contains definitions of more complex relations (ancestors, relatives)
  - is able to answer queries about relationships between people
- Predicates:
  - parent(x, y), child(x, y), father(x, y), daughter(x, y), etc.
  - spouse(x, y), husband(x, y), wife(x,y)
  - ancestor(x, y), descendant(x, y)
  - male(x), female(y)
  - relative(x, y)
- Facts:
  - husband(Joe, Mary), son(Fred, Joe)
  - spouse(John, Nancy), male(John), son(Mark, Nancy)
  - father(Jack, Nancy), daughter(Linda, Jack)
  - daughter(Liz, Linda)
  - etc.

4/16/2024 pra-sâmi

25

Example: A simple genealogy KB by FOL

- Rules for genealogical relations
  - $(\forall x,y)$ parent(x, y) $\leftrightarrow$ child (y, x)
    - $(\forall x,y)$ father(x, y) $\leftrightarrow$ parent(x, y) $\wedge$ male(x) (similarly for mother(x, y))
    - $(\forall x,y)$ daughter(x, y) $\leftrightarrow$ child(x, y) $\wedge$ female(x) (similarly for son(x, y))
  - $(\forall x,y)$ husband(x, y) $\leftrightarrow$ spouse(x, y) $\wedge$ male(x) (similarly for wife(x, y))
    - $(\forall x,y)$ spouse(x, y) $\leftrightarrow$ spouse(y, x) (spouse relation is symmetric)
  - $(\forall x,y)$ parent(x, y) $\rightarrow$ ancestor(x, y)
    - $(\forall x,y)(\exists z)$ parent(x, z) $\wedge$ ancestor(z, y) $\rightarrow$ ancestor(x, y)
  - $(\forall x,y)$ descendant(x, y) $\leftrightarrow$ ancestor(y, x)
  - $(\forall x,y)(\exists z)$ ancestor(z, x) $\wedge$ ancestor(z, y) $\rightarrow$ relative(x, y) (related by common ancestry)
  - $(\forall x,y)$ spouse(x, y) $\rightarrow$ relative(x, y) (related by marriage)
  - $(\forall x,y)(\exists z)$ relative(z, x) $\wedge$ relative(z, y) $\rightarrow$ relative(x, y) (transitive)
  - $(\forall x,y)$ relative(x, y) $\leftrightarrow$ relative(y, x) (symmetric)

- Queries
  - ancestor(Jack, Fred) ### the answer is yes ###
  - relative(Liz, Joe) ### the answer is yes ###
  - relative(Nancy, Matthew) ### no answer in general, no if under closed world assumption ###
  - $(\exists z)$ ancestor(z, Fred) $\wedge$ ancestor(z, Liz)

26

Resolution and Unification

- Given a collection of knowledge
  - We will want to prove certain statements are true or answer questions
- For instance, we might ask
  - Who is Bob's grandfather?
  - Is Sue Barbara's parent?
- How can this be done? Through backward chaining through rules

30

Complete Logic Example

- Assume that we know the following about pets:
  - poodle(COCOA)
  - setter(BIG)
  - terrier(SCOTTY)

- dog(X) ⇒poodle(X) (poodles are dogs)
- dog(X) ⇒ setter(X) (setters are dogs)
- dog(X) ⇒ terrier(X) (terriers are dogs)
- small(X) ⇒ poodle(X) (poodles are small)
- small(X) ⇒ terrier(X) (terriers are small)
- big(X) ⇒ setter(X) (setters are big)
- pet(X) ⇒ dog(X) (dogs are pets)
- indoorPet(X) ⇒ pet(X) and small(X) (small pets are indoor pets)
- outdoorPet(X) ⇒ pet(X) and big(X) (big pets are outdoor pets)

- If we want to find out what would make a good indoor pet, we ask '*indoorPet*'

- This requires finding pet(X) and small(X)
  - find an X to make both predicates true

- pet(X) is implied by dog(X),
- dog(X) is implied by terrier(X),
- SCOTTY is a terrier so SCOTTY is a dog so SCOTTY is a pet

- Can we find if SCOTTY is small? small(SCOTTY) is implied by terrier(SCOTTY) which we already know is true,
- since terrier(SCOTTY) is true, small(SCOTTY) and pet(SCOTTY) are true, so *indoorPet*(SCOTTY) is True

- Continuing with this process will also prove that *indoorPet*(COCOA) is true.

4/16/2024 pra-sâmi

31

Logical Agents for the Wumpus World

- Three (non-exclusive) agent architectures:
  - Reflex agents
    - Have rules that classify situations, specifying how to react to each possible situation
  - Model-based agents
    - Construct an internal model of their world
  - Goal-based agents
    - Form goals and try to achieve them

4/16/2024 pra-sâmi

32

A Simple Reflex Agent

- Rules to map percepts into observations:
  - $\forall$b,g,u,c,t Percept([Stench, b, g, u, c], t) $\Rightarrow$ Stench(t)
  - $\forall$s,g,u,c,t Percept([s, Breeze, g, u, c], t) $\Rightarrow$ Breeze(t)
  - $\forall$s,b,u,c,t Percept([s, b, Glitter, u, c], t) $\Rightarrow$ AtGold(t)
- Rules to select an action given observations:
  - $\forall$t AtGold(t) $\Rightarrow$ Action(Grab, t);
- Some difficulties:
  - Consider Climb. There is no percept that indicates the agent should climb out – position and holding gold are not part of the percept sequence
  - Loops – the percept will be repeated when you return to a square, which should cause the same response (unless we maintain some internal model of the world)

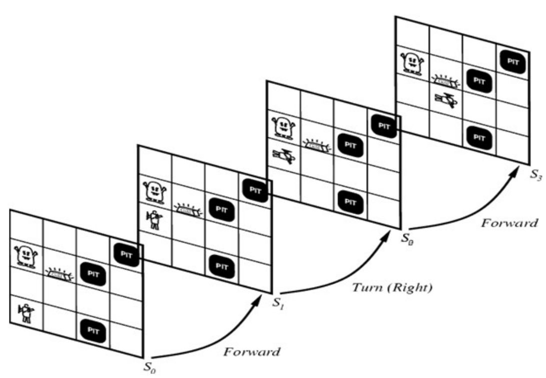
4/16/2024

pra-sâmi

33

Representing Change

- Representing change in the world in logic can be tricky.
- One way is just to change the KB
  - Add and delete sentences from the KB to reflect changes
  - How do we remember the past, or reason about changes?
- Situation calculus is another way
- A situation is a snapshot of the world at some instant in time
- When the agent performs an action A in situation S1, the result is a new situation S2.



4/16/2024

pra-sâmi

34

Situation Calculus

- A situation is a snapshot of the world at an interval of time during which nothing changes
- Every true or false statement is made with respect to a particular situation.
  - Add situation variables to every predicate.
  - at(Agent,1,1) becomes at(Agent,1,1,s0): at(Agent,1,1) is true in situation (i.e., state) s0.
  - Alternatively, add a special 2nd-order predicate, holds(f,s), that means "f is true in situation s." E.g., holds(at(Agent,1,1),s0)
- Add a new function, result(a, s), that maps a situation s into a new situation as a result of performing action a.
  - For example, result(forward, s) is a function that returns the successor state (situation) to s
- Example: The action agent-walks-to-location-y could be represented by
  - $(\forall x)(\forall y)(\forall s)$ (at(Agent,x,s) $\wedge \neg$onbox(s)) $\Rightarrow$ at(Agent,y,result(walk(y),s))

pra-sâmi

35

Deducing Hidden Properties

- From the perceptual information we obtain in situations, we can infer properties of locations
  - $\forall$ l, s at(Agent, l, s) $\wedge$ Breeze(s) $\Rightarrow$ Breezy(l)
  - $\forall$l, s at(Agent, l, s) $\wedge$ Stench(s) $\Rightarrow$ Smelly(l)
- Neither Breezy nor Smelly need situation arguments because pits and Wumpus do not move around

pra-sâmi

36

Deducing Hidden Properties

- We need to write some rules that relate various aspects of a single world state (as opposed to across states)

- There are two main kinds of such rules:
  - Causal rules reflect the assumed direction of causality in the world:
    - $(\forall l1, l2, s)$ At(Wumpus,l1,s) $\wedge$ Adjacent(l1, l2) $\Rightarrow$ Smelly(l2)
    - $(\forall$ l1, l2, s) At(Pit,l1,s) $\wedge$ Adjacent(l1, l2) $\Rightarrow$ Breezy(l2)
  - Systems that reason with causal rules are called model-based reasoning systems
  - Diagnostic rules infer the presence of hidden properties directly from the percept-derived information:
    - $(\forall$ l, s) At(Agent,l , s) $\wedge$ Breeze(s) $\Rightarrow$ Breezy(l)
    - $(\forall$ l, s) At(Agent, l, s) $\wedge$ Stench(s) $\Rightarrow$ Smelly(l)

4/16/2024

pra-sâmi

37

Preferences Among Actions

- A problem with the Wumpus world knowledge base that we have built so far is that it is difficult to decide which action is best among a number of possibilities.

- For example, to decide between a forward and a grab, axioms describing when it is OK to move to a square would have to mention glitter.

- This is not modular!

- We can solve this problem by separating facts about actions from facts about goals. This way our agent can be reprogrammed just by asking it to achieve different goals.

4/16/2024

pra-sâmi

38

Preferences Among Actions

- The first step is to describe the desirability of actions
- In doing this we will use a simple scale
- Actions can be Great, Good, Medium, Risky, or Deadly.
- Obviously, the agent should always do the best action it can find:
  - $(\forall a, s)$ Great(a,s) $\Rightarrow$ Action(a,s)
  - $(\forall a, s)$ Good(a,s) $\wedge \neg(\exists b)$ Great(b,s) $\Rightarrow$ Action(a,s)
  - $(\forall a, s)$ Medium(a,s) $\wedge$ $(\neg(\exists b)$ Great(b,s) $\vee$ Good(b,s)) $\Rightarrow$ Action(a,s)
  - ...

pra-sâmi

39

Preferences among actions

- We use this action quality scale in the following way.
- Until it finds the gold, the basic strategy for our agent is:
  - Great actions include picking up the gold when found and climbing out of the cave with the gold.
  - Good actions include moving to a square that's OK and hasn't been visited yet.
  - Medium actions include moving to a square that is OK and has already been visited.
  - Risky actions include moving to a square that is not known to be deadly or OK.
  - Deadly actions are moving into a square that is known to have a pit or a Wumpus.

pra-sâmi

40

Goal-based Agents

- Once the gold is found, it is necessary to change strategies. So now we need a new set of action values.
- We could encode this as a rule:
  - $(\forall s)$ Holding(Gold,s) $\Rightarrow$ GoalLocation([1,1]),s)
- We must now decide how the agent will work out a sequence of actions to accomplish the goal.
- Three possible approaches are:
  - Inference: good versus wasteful solutions
  - Search: make a problem with operators and set of states
  - Planning: to be discussed later

41

Resolution by Graph

42

FOL Resolution - Steps

- Step 1: Convert into first order logic
- Step 2: Convert first order logic (FOL) to Conjunctive Normal Form
- Step 3: Negate the statement to be proven
- Step 4: Draw resolution graph

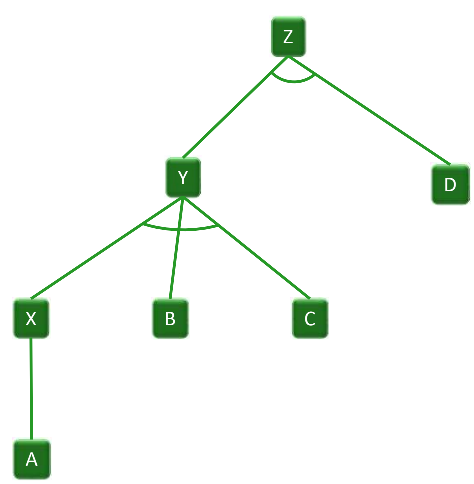
4/16/2024

pra-sâmi

43

Inference Chain

- Rule 1:
  - If A is True we can inference X
- Rule 2:
  - IF X is True
  - And B is True
  - And C is True
  - THEN Y is True
- Rule 3:
  - IF Y is True
  - And D is True
  - THEN Z is True



4/16/2024

pra-sâmi

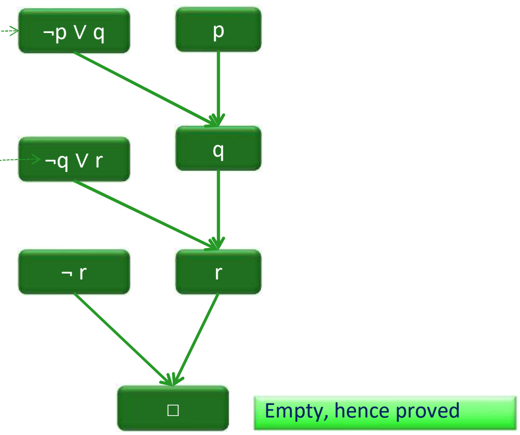
44

Example 1 : Resolution

$p \rightarrow q$
$q \rightarrow r$
$\neg r$

$\therefore \neg p$

- To prove that it is ¬p, we will negate it and draw graph with ¬¬p ➔p
- First convert them in CNF
  - p → q becomes ¬p V q
  - q → r becomes ¬q V r
  - ¬ r



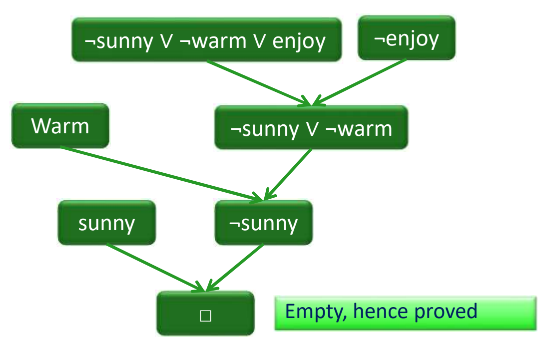
4/16/2024

45

Example 2: Resolution

1. It is a sunny and warm day, therefore you will enjoy
   - Sunny ∧ warm → enjoy
   - becomes ¬ (sunny ∧ warm) V enjoy
   - becomes ¬ sunny V ¬ warm V enjoy
2. It is raining, therefore you will get wet
   - Raining → wet
   - becomes ¬ (raining) V wet
3. It is a warm day
   - Warm
4. It is raining
   - Rainy
5. It is sunny day
   - Sunny

- To prove that we are going to enjoy lets negate it becomes ¬enjoy



- BTW: We never used 2 and 4.

4/16/2024

48

Horn Clauses

- Definition:

  A Horn clause is a clause that has at most one positive literal.

- Examples:
  - P; P ∨ ¬Q; ¬ P ∨ ¬Q; ¬ P ∨ ¬Q ∨ R;

- Types of Horn Clauses:
  - Fact – single atom – e.g., P;
  - Rule – implication, whose antecendent is a conjunction of positive literals and whose consequent consists of a single positive literal – e.g., P∧Q → R; Head is R; Tail is (P∧Q)
  - Set of negative literals - in implication form, the antecedent is a conjunction of positive literals and the consequent is empty.
    - e.g., P∧Q → ; equivalent to ¬ P ∨ ¬Q.

Inference with propositional Horn clauses can be done in linear time ☺!

4/16/2024 pra-sâmi

49

Forward Chaining

- Start with atomic sentences in the KB
- Apply inference rules (Modus Ponens) in the forward direction
- Keep adding new atomic sentences
- Until no further inferences can be made
- Facts are kept in working memory and condition-action rule represents action to be taken when some facts are true
  - It may add and/or delete facts from working memory

4/16/2024 pra-sâmi

50

Forward Chaining

- Given a new fact, generate all consequences
- Assumes all rules are of the form
  - C1 and C2 and C3 and…. Lead to 'Result'
- Each rule & binding generates a new fact
- This new fact will "trigger" other rules
- Keep going until the desired fact is generated
- Semi-decidable as is FOL in general

4/16/2024

pra-sâmi

51

Properties

- A Bottom-Up Approach
  - Moves from facts to goal
- It is a process of making a conclusion based on known facts or data
  - Start from initial state keep going till you reach goal state
- It is also called Data-Driven approach
  - Data determines which rules are selected and used
- Example
  - A X exercises regularly
  - A → B Regular exercise imply being fit
  - B X is fit.

4/16/2024

pra-sâmi

52

Forward Chaining: Diagnosis systems

- Example: diagnostic system
  - IF the engine is getting gas and the engine turns over
    THEN the problem is spark plugs
  - IF the engine does not turn over and the lights do not come on
    THEN the problem is battery or cables
  - IF the engine does not turn over and the lights come on
    THEN the problem is starter motor
  - IF there is gas in the fuel tank and there is gas in the carburettor
    THEN the engine is getting gas

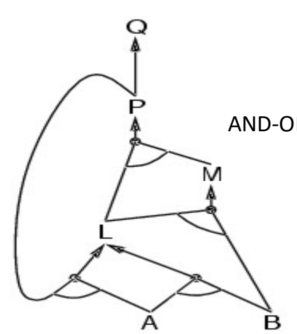
pra-sâmi

53

Forward chaining :Data driven reasoning

- Idea: fire any rule whose premises are satisfied in the KB,
  - add its conclusion to the KB, until query is found

$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$



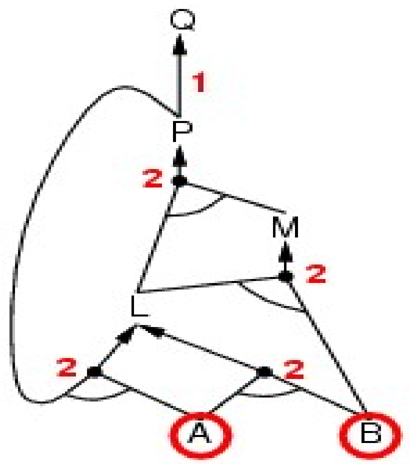
AND-OR graph

pra-sâmi

54

Forward chaining example

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B



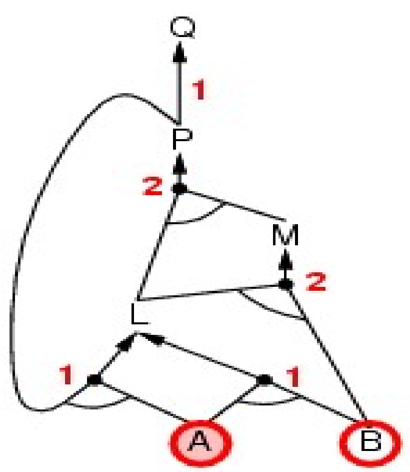
4/16/2024

pra-sâmi

55

Forward chaining example

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B



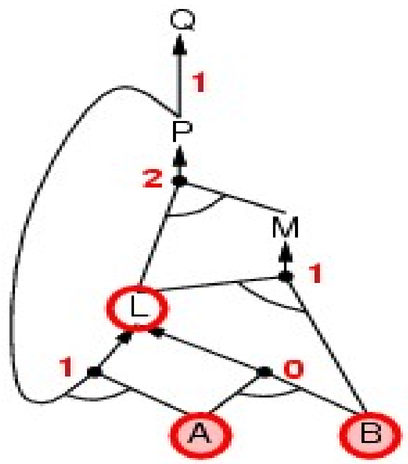
4/16/2024

pra-sâmi

56

Forward chaining example

$$\begin{aligned} P &\Rightarrow Q \\ L \wedge M &\Rightarrow P \\ B \wedge L &\Rightarrow M \\ A \wedge P &\Rightarrow L \\ A \wedge B &\Rightarrow L \\ A & \\ B & \end{aligned}$$



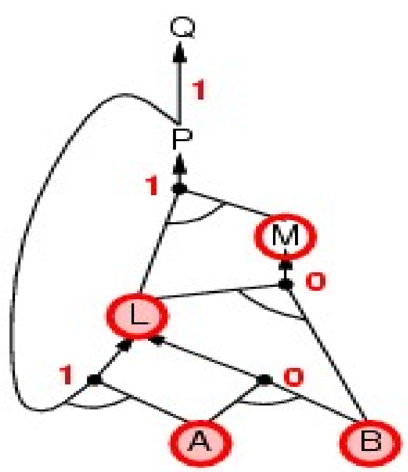
4/16/2024

pra-sâmi

57

Forward chaining example

$$\begin{aligned} P &\Rightarrow Q \\ L \wedge M &\Rightarrow P \\ B \wedge L &\Rightarrow M \\ A \wedge P &\Rightarrow L \\ A \wedge B &\Rightarrow L \\ A & \\ B & \end{aligned}$$



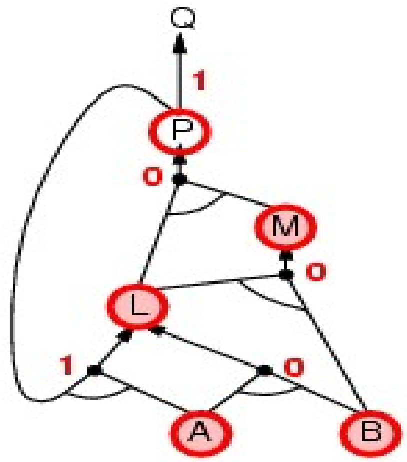
4/16/2024

pra-sâmi

58

Forward chaining example

$$
\begin{aligned}
&P \Rightarrow Q \\
&L \wedge M \Rightarrow P \\
&B \wedge L \Rightarrow M \\
&A \wedge P \Rightarrow L \\
&A \wedge B \Rightarrow L \\
&A \\
&B
\end{aligned}
$$



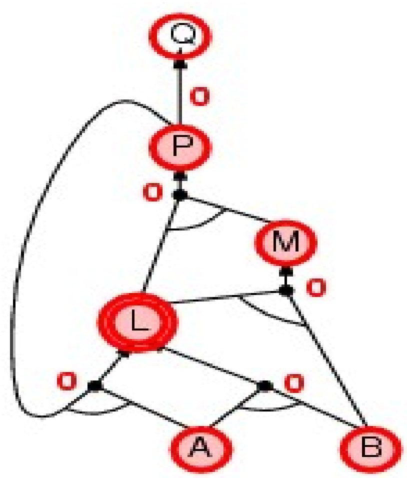
4/16/2024

pra-sâmi

59

Forward chaining example

$$
\begin{aligned}
&P \Rightarrow Q \\
&L \wedge M \Rightarrow P \\
&B \wedge L \Rightarrow M \\
&A \wedge P \Rightarrow L \\
&A \wedge B \Rightarrow L \\
&A \\
&B
\end{aligned}
$$



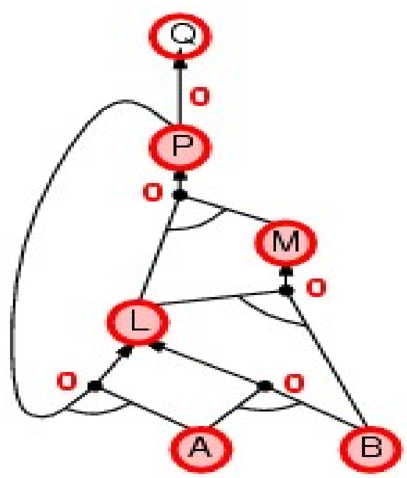
4/16/2024

pra-sâmi

60

Forward chaining example

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B



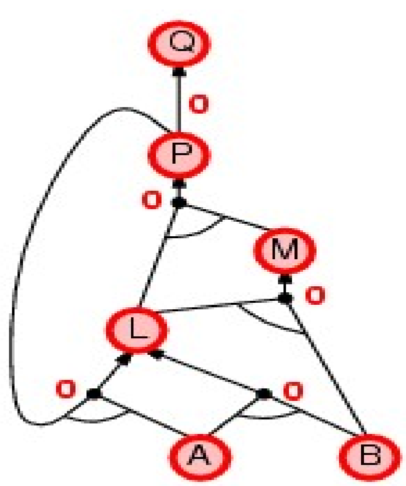
4/16/2024

pra-sâmi

61

Forward chaining example

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B



4/16/2024

pra-sâmi

62

Backward chaining

- Idea: work backwards from the query q:
  - to prove q by BC,
    - check if q is known already, or
    - prove 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
  - has already been proved true, or
  - has already failed

pra-sâmi

63

Backward Chaining

- It is a goal driven method
- Derive a goal given knowledge base and set on inference
- System starts from a goal and works backwards
- Find a rule whose head is the goal (and bindings)
- Apply bindings to the body, and prove these (subgoals) in turn
- If you prove all the subgoals, increasing the binding set as you go, you will prove the item
- It is Depth First Search

pra-sâmi

64

Properties

- A Top-Down Approach
  - Moves from goals to sub-goals to facts
- It is a process of making a conclusion based on goal
  - Start from goal state and investigate if facts support this goal
- It is also called Goal Driven or Decision Driven approach
  - Goals, sub-goals decides which rules are selected and used
- Example
  - B X is fit.
  - A → B Regular exercise imply being fit
  - A X exercises regularly

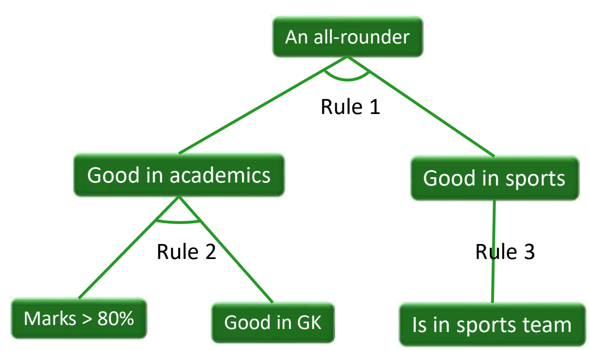
4/16/2024

pra-sâmi

65

Example

- Rule 1:
  - IF A student is good in academics
  - AND also in the school sports team
  - THEN is an all-rounder
- Rule 2:
  - IF Any student gets >80% marks
  - AND also good in GK
  - THEN is good in academics
- Rule 3:
  - IF student is in the sports team
  - THEN is good in sports
- Facts are same
  - Student gets marks > 80 %
  - Also is in sports team
  - And good in GK
- Given : the student is an all rounder



4/16/2024

pra-sâmi

66

Forward vs. backward chaining

- FC is data-driven, automatic, unconscious processing,
  - e.g., object recognition, routine decisions
- May do lots of work that is irrelevant to the goal
- BC is goal-driven, appropriate for problem-solving,
  - e.g., Where are my keys? How do I get into a company?
- Complexity of BC can be much less than linear in size of KB in practice.

pra-sâmi