Artificial Intelligence

Lec 20: First Order Logic (contd.)

Pratik Mazumder

Content taken from various sources

Equivalence in FOL

- In FOL, two formulas F_1 and F_2 are equivalent if $F_1 \Leftrightarrow F_2$ is valid
- In Propositional Logic, we could have proved equivalence using truth tables, but not possible in FOL.
  - because FOL deals with quantifiers that range over an infinite domain of objects, whereas
  - propositional logic deals with a finite set of propositional variables.
- However, we can still use known equivalences to re-write one formula as another.
- Equivalent: $\neg \forall x \ \exists y \ P(x,y) \equiv \exists x \ \forall y \ \neg P(x,y)$

Equivalence in FOL

<u>The **name** of the equivalence used **MUST** be mentioned.</u>

Otherwise No Considerations

$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

Equivalence in FOL

The **name** of the equivalence used **MUST** be mentioned.

Otherwise No Considerations

- Law of double negation: $\neg\neg\phi \equiv \phi$
- Identity Laws: $\phi \wedge T \equiv \phi \qquad \phi \vee F \equiv \phi$
- Domination Laws: $\phi \vee T \equiv T \qquad \phi \wedge F \equiv F$
- Idempotent Laws: $\phi \vee \phi \equiv \phi \qquad \phi \wedge \phi \equiv \phi$
- Negation Laws: $\phi \wedge \neg\phi \equiv F \qquad \phi \vee \neg\phi \equiv T$
- Absorption Laws: $\phi_1 \wedge (\phi_1 \vee \phi_2) \equiv \phi_1 \qquad \phi_1 \vee (\phi_1 \wedge \phi_2) = \phi_1$

Rules of Inference

- In First-Order logic (FOL), **sentences need to be instantiated before applying inference rules.**

- Most basic inference rule: Modus Ponens

$$\frac{\begin{array}{l}P(a)\\ P(a)\Rightarrow Q(a)\end{array}}{Q(a)}$$

Rules of Inference

- Modus Tollens

  $Q(a)$
  $\neg P(a) \Rightarrow \neg Q(a)$
  $\overline{}$
  $P(a)$

Rules of Inference

- Hypothetical Syllogism: Implication is Transitive

$$\frac{\begin{array}{l} P(a) \Rightarrow Q(a) \\ Q(a) \Rightarrow R(a) \end{array}}{P(a) \Rightarrow R(a)}$$

Rules of Inference

- Or Introduction

  $P(a)$ [Or Introduction] [If P is true, then $P \vee Q$ will also be True]

  ———

  $P(a) \vee Q(a)$

- Or Elimination [If P or Q is true, and $P \Rightarrow R$ and $Q \Rightarrow R$ are True, then R will also be True]

  $P(a) \vee Q(a)$
  $P(a) \Rightarrow R(a)$
  $Q(a) \Rightarrow R(a)$

  ———

  $R(a)$

Rules of Inference

- And Introduction

  $P(a)$ [And Introduction] [If we know that P is True and Q is True, then P and Q will also be True]
  $Q(a)$

  $$\frac{P(a) \quad Q(a)}{P(a) \land Q(a)}$$

- And Elimination

  $P \land Q$ [And Elimination] [If P and Q both are true, then P will also be True]

  $$\frac{P \land Q}{P}$$

Rules of Inference

- Resolution

$$\frac{\begin{array}{l} A(a) \lor B(a) \lor \neg C(a) \\ \neg A(a) \lor \neg E(a) \end{array}}{B(a) \lor \neg C(a) \lor \neg E(a)}$$

Rules of Inference

- Universal Elimination/Instantiation - Elimination of the universal quantifier
  - If we know that something is True for all members of a set then it will also be True for a specific element from that set.

$\forall$x Likes(x,flower)

  - Substituting x by Sagar gives

Likes (Sagar,flower)

  - The substitution should be done by a constant term

Rules of Inference

- Universal Elimination/Instantiation -
    - Consider Predicates man(x) and mortal(x) and the hypotheses:

1. All men are mortal
2. Socrates is a man

    - Using rules of inference, prove mortal(Socrates)

1. $\forall x$ man(x)$\Rightarrow$mortal(x) [Hypothesis]
2. man(Socrates) [Hypothesis]
3. man(Socrates)$\Rightarrow$mortal(Socrates) [Universal Instantiation for 1]
4. mortal(Socrates) [Modus Ponens to 2,3]

Rules of Inference

- Existential Elimination/Instantiation (Skolemization) - allows you to infer that there is a specific element, say c, that satisfies P(c), where c is a new constant (a "witness" or "placeholder") introduced to represent the existence of such an element.

  $\exists x\, P(x)$

  - Suppose we know that there is some element c from the domain for which P(c) is True, then we can do existential instantiation and write

    $P(c)$

  $\exists x$ Likes(x,flower) –> Likes (c,flower)

  - Placeholder c should not be in the Knowledge Base.
  - You can replace c with the element in the domain that satisfies the above predicate.
  - It's important to note that the constant introduced by existential instantiation is arbitrary and cannot be assumed to represent a specific, named object unless additional information is provided.
    - e.g., if P(x) is True for Ram, the P(Ram) is valid.

Rules of Inference

- Existential Introduction/Generalization
  - Suppose we know that P(c) is True for some constant c
  - Then there exists atleast one element c for which P is True
  - So we can conclude that $\exists x\, P(x)$

Likes (Sagar,flower) can be written as

$\exists x$ Likes(x,flower)

This will be True since we know that atleast one element of the domain, i.e., Sagar likes flowers.

Reasoning in FOL

- Consider the following problem:

  If a perfect square is divisible by a prime p, then it is also divisible by square of p.

  Every perfect square is divisible by some prime number.

  36 is a perfect square.

- Does there exist a prime q such that square of q divides 36

Representation in FOL

- If a perfect square is divisible by a prime p, then it is also divisible by square of p.

  $\forall x, \forall y\ (perfect\_sq(x) \land prime(y) \land divides(x,y) \Rightarrow divides(x,square(y)))$

- Every perfect square is divisible by some prime number.

  $\forall x, \exists y\ (perfect\_sq(x) \land prime(y) \land divides(x,y))$

- 36 is a perfect square.

  perfect_sq(36)

- Does there exist a prime q such that square of q divides 36

  $\exists y\ prime(y) \land divides(36,square(y))$

Substitution

- Substitution **replaces variables in Predicates with constants.**

  SUBST ({x/49}, Perfect_sq(x)) = Perfect_sq(49)
  SUBST ({x/49,y/7}, Divides(x,y)) = Divides(49,7)

- The rule of Universal Instantiation (UI for short) says that we can infer any sentence obtained by substituting a ground term (a term without variables) for the variable.

- Let SUBST(θ, α) denote the result of applying the substitution θ to the sentence α. Then the rule is written as $\dfrac{\forall v,\ \alpha}{\mathrm{SUBST}(v/g,\ \alpha)}$, for any variable v and ground term g.

- In the rule for Existential Instantiation (EI), the variable is replaced by a single new constant symbol. The formal statement is as follows: for any sentence α, variable v, and constant symbol k that does not appear elsewhere in the knowledge base $\dfrac{\exists v,\ \alpha}{\mathrm{SUBST}(v/k,\ \alpha)}$

- Whereas Universal Instantiation can be applied many times to produce many different consequences, Existential Instantiation can be applied once.

Reduction to propositional inference

- Once we have rules for inferring non quantified sentences from quantified sentences, it becomes possible to **reduce first-order inference to propositional inference**.
- Just as an existentially quantified sentence can be replaced by one instantiation, a universally quantified sentence can be replaced by the set of all possible instantiations.
- e.g., suppose the KB contains just the sentences.

  $\forall$ x King(x) $\wedge$ Greedy (x) $\Rightarrow$ Evil(x)
  King(John)
  Greedy (John)
  Brother (Richard , John)

- We apply UI to the first sentence using all possible ground-term substitutions from the vocabulary of the knowledge base—in this case, {x/John} and {x/Richard }.
- We obtain

  King(John) $\wedge$ Greedy (John) $\Rightarrow$ Evil(John)
  King(Richard) $\wedge$ Greedy (Richard) $\Rightarrow$ Evil(Richard)

- and we discard the universally quantified sentence.

- Now, the KB is essentially propositional if we view the ground atomic sentences—King (John), Greedy (John), and so on—as proposition symbols.
- Therefore, we can apply any of the propositional inference methods to obtain conclusions.

Reduction to propositional inference

- When the knowledge base includes a function symbol, the set of possible ground-term substitutions is infinite.

- For example, if the knowledge base mentions the Father function, then infinitely many nested terms such as Father (Father (Father (John))) can be constructed.

- Propositionalization will also produce lots of sentences that are not relevant to the current task, such as, Greedy(Richard) from

  $\forall$ x King(x) $\land$ Greedy (x) $\Rightarrow$ Evil(x)
  King(John)
  $\forall$ y Greedy (y)
  Brother (Richard , John)

- With p k-ary predicates and n constants, there are around p x n^k instantiations.

- The propositional inference methods will have difficulty with an infinitely large set of sentences making the propositionalization approach rather inefficient.

Reduction to propositional inference

- <u>KB</u>
  $\forall$ x King(x) $\wedge$ Greedy (x) $\Rightarrow$ Evil(x)
  King(John)
  Greedy (John)
  Brother (Richard , John)

  Query: Evil(John)

- Find some x such that x is a king and x is greedy, and then infer that this x is evil.
    - We can easily show this for SUBST{x/John}

- Suppose that instead of knowing Greedy (John), we know that everyone is greedy, i.e. $\forall$y Greedy(y).
    - We need to find a substitution both for the variables in the implication sentence and for the variables in the sentences that are in the knowledge base.

- Applying the substitution {x/John, y/John} to the implication premises King(x) and Greedy (x) and the knowledge-base sentences King(John) and Greedy (y) will make them identical.
- Then, we can apply Modus Ponens.

Generalized Modus Ponens

- For atomic sentences pi, pi′, and q, where there is a substitution θ, such that SUBST(θ, p_i′) = SUBST(θ, p_i), for all i,

$$\frac{p1', p2', \ldots, pn', (p1 \land p2 \land \ldots \land pn \Rightarrow q)}{\text{SUBST}(\theta, q)}$$

- There are n + 1 premises to this rule: the n atomic sentences p_i′ and the one implication.

- The conclusion is the result of applying the substitution θ to the consequent q. (Generalized Modus Ponens)

KB
∀ x King(x) ∧ Greedy (x) ⇒ Evil(x)
King(John)
Greedy (John)
Brother (Richard , John)
----

p1′ is King(John)	p1 is King(x)
p2′ is Greedy (y)	p2 is Greedy (x)
θ is {x/John, y/John}	q is Evil(x)
SUBST(θ, q) is Evil(John)	

Generalized Modus Ponens

- Generalized Modus Ponens is a lifted version of Modus Ponens.
  - it raises Modus Ponens from ground (variable-free) propositional logic to first-order logic.

- The key advantage of lifted inference rules over propositionalization is that they make only those substitutions that are required to allow particular inferences to proceed.

- All variables are assumed to be universally quantified.