CSCI 561 Foundations of Artificial Intelligence

Lecture 12: Practical first-order logic

(Chapter 8)

SPRING 2015

INSTRUCTOR: PROF. ANDREW GORDON

Exists $\exists$

$$\exists \quad x \quad . \underbrace{\mathit{owns}(x, \mathit{Dog}) \wedge \mathit{bites}(\mathit{Dog}, x)}_{\text{variable scope}}$$

quantifier variable

There exists an *x* such that *x* owns Dog and Dog bites *x*.

For all $\forall$

$$\forall \quad x, y \quad . \underbrace{\mathit{sonOf}(x, y) \Rightarrow \mathit{parentOf}(y, x)}_{\text{variable scope}}$$

quantifier variables

For all *x* and *y*, if *x* is the son of *y*, then *y* is the parent of *x*.

Nested scope

$$\forall x\,.\,\underbrace{car(x) \Rightarrow \exists y\,.\,\underbrace{engineOf(y, x)}_{\text{Scope of y}}}_{\text{Scope of x}}$$

For all x, x is a car implies that there exists a y that is the engine of x.

For all x, if x is a car then there exists a y that is the engine of x.

For all cars x, there exists a y that is the engine of x.

All cars x have an engine y.

Prenex normal form

$$\forall x \,.\, car(x) \Rightarrow \exists y \,.\, engineOf(y, x)$$

$$\Leftrightarrow$$

$$\underbrace{\forall x \;\exists y}_{\text{Prefix}} \,.\, \underbrace{\neg car(x) \vee engineOf(y, x)}_{\text{Matrix}}$$

For all x there exists a y such that x is not a car, or y is the engine of x.

- All sentences in first-order logic can be converted into prenex normal form via a series of transformations.
- We'll need to do this later, when we try first-order theorem proving.

Quantifier ordering

"Everybody loves somebody."

$\forall x \; \exists y \; . \; Loves(x, y)$

"Everybody is loved by somebody."

$\forall y \; \exists x \; . \; Loves(x, y)$

"There is someone who is loved by everybody."

$\exists y \; \forall x \; . \; Loves(x, y)$

"There is someone who loves everyone."

$\exists x \; \forall y \; . \; Loves(x, y)$

Quantifier negation

$\forall$ and $\exists$ are connected to each other through **negation**.

$$\forall x \,.\, \mathit{Likes}(x, \mathit{IceCream}) \Leftrightarrow \neg \exists x \,.\, \neg \mathit{Likes}(x, \mathit{IceCream})$$

More generally:

$$\forall x \,.\, P \Leftrightarrow \neg \exists x \,.\, \neg P$$

Hey, this looks familiar…

$$\mathrm{P} \wedge \mathrm{Q} \Leftrightarrow \neg(\neg \mathrm{P} \vee \neg \mathrm{Q})$$

De Morgan

$\forall$ is really acting like **conjunction**, and $\exists$ is acting like a **disjunction**.

Quantifier equivalences

$$\forall x \,.\, \neg P \iff \neg \exists x \,.\, P$$

$\forall x \,.\, \neg Likes(x, Parsnips) \iff \neg \exists x \,.\, Likes(x, Parsnips)$

$$\neg \forall x \,.\, P \iff \exists x \,.\, \neg P$$

$\neg \forall x \,.\, Likes(x, PrincessLeia) \iff \exists x \,.\, \neg Likes(x, PrincessLeia)$

$$\forall x \,.\, P \iff \neg \exists x \,.\, \neg P$$

$\forall x \,.\, Likes(x, IceCream) \iff \neg \exists x \,.\, \neg Likes(x, IceCream)$

$$\neg \forall x \,.\, \neg P \iff \exists x \,.\, P$$

$\neg \forall x \,.\, \neg Likes(x, DarthVader) \iff \exists x \,.\, Likes(x, DarthVader)$

Exercise 8.10

Consider an ontology with the following symbols:

Occupation(p,o) : Person p has occupation o

Customer(p1, p2) : Person p1 is a customer of person p2

Boss(p1, p2) : Person p1 is a boss of person p2

Doctor, *Surgeon*, *Lawyer*, *Actor* : Constants denoting occupations

Emily, *Joe*: Constants denoting people

Use these symbols to express knowledge in first-order logic

a.

Emily is either a surgeon or a lawyer.

$Occupation(Emily, Surgeon) \lor Occupation(Emily, Lawyer)$

b.

Joe is an actor, but he also holds another job.

$Occupation(Joe, Actor) \wedge \exists x\,.\,Occupation(Joe, x) \wedge \neg(x = Actor)$

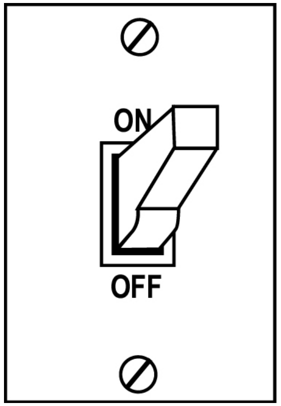
* The equality literal states that two terms refer to the same object. Here, the negation indicates that $x \neq Actor$.

Equality $=$

An **equality** literal states that two objects in the domain are the same.

As an atomic sentence, it can be either True or False.

$$(x = Sarah) \lor (x = Julie)$$



Like other literals, it can be negated.

$$\exists\, x\ .\ tallest(x) \lor \neg(x = Everest)$$

$$\exists\, x\ .\ tallest(x) \lor (x \neq Everest)$$ *(acceptable)*

Sometimes useful to equate functions with named constants.

$$Father(John) = Henry$$

c.

All surgeons are doctors.

$\forall\ x\ .\ Occupation(x, Surgeon) \Rightarrow Occupation(x, Doctor)$

d.

Joe does not have a lawyer (i.e., is not a customer of any lawyer).

$\neg \exists\ x\ .\ Occupation(x, Lawyer) \wedge Customer(Joe, x)$

$\forall\ x\ .\ \neg(Occupation(x, Lawyer) \wedge Customer(Joe, x))$

$\forall\ x\ .\ \neg Occupation(x, Lawyer) \vee \neg Customer(Joe, x)$

e.

Emily has a boss who is a lawyer.

$\exists\ x\ .\ Boss(x, Emily) \land Occupation(x, Lawyer)$

f.

There exists a lawyer all of whose customers are doctors.

$$\exists\ x\ .\ Occupation(x, Lawyer) \land$$
$$\forall\ y\ .\ Customer(y, x) \Rightarrow Occupation(y, Doctor)$$

This one is difficult for many students, who don't understand why we need an implication instead of just another conjunction. The key thing to remember is that universal quantification applies to ALL objects in domain, and for all of these objects, we want the sentence in the scope of y to be True. Implications are always true EXCEPT in the case where the antecedent is True and the consequent is False, which is what we want here. Another way to think about it is this: for all objects in the domain, this "rule" is True: if they are a customer of this Lawyer x, then their occupation is Doctor. Easy!

g.

Every surgeon has a lawyer.

$$\forall\ x\ .\ Occupation(x, Surgeon) \Rightarrow$$
$$\exists\ y\ .\ Occupation(y, Lawyer) \wedge Customer(x, y)$$

More examples

A dog bites a man who is its owner.

$\exists\ x, y\ .\ dog(x) \land man(y) \land own(y, x) \land bite(x, y)$

A policeman arrests a protester.

$\exists\ x, y\ .\ policeman(x) \land protester(y) \land arrest(x, y)$

A boy builds a boat.

$\exists\ x, y\ .\ boy(x) \land boat(y) \land build(x, y)$