First-Order Logic

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, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ...
  - Properties: blue, oval, even, large, ...
  - Functions: father-of, best-friend, second-half, one-more-than ...

User provides

- **Constant symbols,** which represent individuals in the world
  - Mary
  - 3
  - Green
- **Function symbols,** which map individuals to individuals
  - father-of(Mary) = John
  - color-of(Sky) = Blue
- **Predicate symbols,** which map individuals to truth values
  - greater(5,3)
  - green(Grass)
  - color(Grass, Green)

FOL Provides

- **Variable symbols**
  - E.g., x, y, foo
- **Connectives**
  - Same as in PL: not ($\neg$), and ($\wedge$), or ($\vee$), implies ($\rightarrow$), if and only if (biconditional $\leftrightarrow$)
- **Quantifiers**
  - Universal $\forall$**x** or **(Ax)**
  - Existential $\exists$**x** or **(Ex)**

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 $f(x_1, \ldots, x_n)$ are terms, where each x_i is a term.
  - 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.

A BNF for FOL

```
S := <Sentence> ;
<Sentence> := <AtomicSentence> |
          <Sentence> <Connective> <Sentence> |
          <Quantifier> <Variable>,... <Sentence> |
          "NOT" <Sentence> |
          "(" <Sentence> ")";
<AtomicSentence> := <Predicate> "(" <Term>, ... ")" |
                    <Term> "=" <Term>;
<Term> := <Function> "(" <Term>, ... ")" |
          <Constant> |
          <Variable>;
<Connective> := "AND" | "OR" | "IMPLIES" | "EQUIVALENT";
<Quantifier> := "EXISTS" | "FORALL" ;
<Constant> := "A" | "X1" | "John" | ... ;
<Variable> := "a" | "x" | "s" | ... ;
<Predicate> := "Before" | "HasColor" | "Raining" | ... ;
<Function> := "Mother" | "LeftLegOf" | ... ;
```

Quantifiers

- **Universal quantification**
  – $(\forall x)P(x)$ means that P holds for **all** values of x in the domain associated with that variable
  – E.g., $(\forall x)$ dolphin(x) $\rightarrow$ mammal(x)
- **Existential quantification**
  – $(\exists\, x)P(x)$ means that P holds for **some** value of x in the domain associated with that variable
  – E.g., $(\exists\, x)$ mammal(x) $\wedge$ lays-eggs(x)
  – Permits one to make a statement about some object without naming it

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 person who is *not* a student?

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 *does* change meaning:
  - Everyone likes someone: $(\forall x)(\exists y)$ likes(x,y)
  - Someone is liked by everyone: $(\exists y)(\forall x)$ likes(x,y)

Connections between All and Exists

We can relate sentences involving $\forall$ and $\exists$ using De Morgan's laws:

$$(\forall x)\ \neg P(x) \leftrightarrow \neg(\exists x)\ P(x)$$

$$\neg(\forall x)\ P \leftrightarrow (\exists x)\ \neg P(x)$$

$$(\forall x)\ P(x) \leftrightarrow \neg\ (\exists x)\ \neg P(x)$$

$$(\exists x)\ P(x) \leftrightarrow \neg(\forall x)\ \neg P(x)$$

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
  - $\exists x\ P(x) \therefore P(F)$ $\leftarrow$ **skolem constant F**
- Existential generalization
  - $P(A) \therefore \exists x\ P(x)$

Universal instantiation (a.k.a. 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)
- The variable symbol can be replaced by any ground term, i.e., any constant symbol or function symbol applied to ground terms only

Existential instantiation (a.k.a. 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 by introducing the constant
- Convenient to use this to reason about the unknown object, rather than constantly manipulating the existential quantifier

Existential generalization (a.k.a. existential introduction)

- If P(c) is true, then $(\exists x)$ P(x) is inferred.
- Example

  eats(Ziggy, IceCream) $\Rightarrow$ $(\exists x)$ 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

Translating English to FOL

Every gardener likes the sun.

$\forall x\ gardener(x) \rightarrow likes(x,Sun)$

You can fool some of the people all of the time.

$\exists x\ \forall t\ \ person(x) \wedge time(t) \rightarrow can\text{-}fool(x,t)$

You can fool all of the people some of the time.

$\forall x\ \exists t\ (person(x) \rightarrow time(t) \wedge can\text{-}fool(x,t))$

$\forall x\ (person(x) \rightarrow \exists t\ (time(t) \wedge can\text{-}fool(x,t)))$

Equivalent

All purple mushrooms are poisonous.

$\forall x\ (mushroom(x) \wedge purple(x)) \rightarrow poisonous(x)$

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

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 and ending with Y.

$\forall x\ \forall y\ above(x,y) \leftrightarrow (on(x,y) \vee\ \exists z\ (on(x,z) \wedge above(z,y)))$

Examples

- Everyone likes McDonalds
  - $\forall x,\ likes(x, McDonalds)$
- Someone likes McDonalds
  - $\exists x,\ likes(x, McDonalds)$
- All children like McDonalds
  - $\forall x,\ child(x) \Rightarrow likes(x, McDonalds)$
- Everyone likes McDonalds unless they are allergic to it
  - $\forall x, likes(x, McDonalds) \vee allergic(x, McDonalds)$
  - $\forall x, \neg allergic\ (x, McDonalds) \Rightarrow likes(x, McDonalds)$

Properties of Quantifiers

- $\forall x\ \forall y$ is the same as $\forall y\ \forall x$
- $\exists x\ \exists y$ is the same as $\exists y\ \exists x$
- $\exists x\ \forall y$ is not the same as $\forall y\ \exists x$
  - $\exists x\ \forall y$ Loves(x, y)
    - “There is a person who loves everyone in the world”
  - $\forall y\ \exists x$ Loves(x, y)
    - “Everyone in the world is loved by at least one person”

Nesting Quantifiers

- Everyone likes some kind of food

  $\forall y \, \exists x, \; food(x) \wedge likes(y, x)$
- There is a kind of food that everyone likes

  $\exists x \, \forall y, \; food(x) \wedge likes(y, x)$
- Someone likes all kinds of food

  $\exists y \, \forall x, \; food(x) \wedge likes(y, x)$
- Every food has someone who likes it

  $\forall x \, \exists y, \; food(x) \wedge likes(y, x)$

Examples

- Quantifier Duality
  - Not everyone like McDonalds

    $\neg(\forall x,\ likes(x, McDonalds))$

    $\exists x, \neg likes(x, McDonalds)$

  - No one likes McDonalds

    $\neg(\exists x,\ likes(x, McDonalds))$

    $\forall x, \neg likes(x, McDonalds)$

Fun with Sentences

- Brothers are siblings

  $\forall x,y \;\; Brother(x,y) \Rightarrow Sibling(x, y)$

- Sibling is "symmetric"

  $\forall x,y \;\; Sibling(x,y) \Leftrightarrow Sibling(y, x)$

Fun with Sentences

- One's mother is one's female parent

  $\forall x,y \; Mother(x,y) \Leftrightarrow (Female(x) \wedge Parent(x,y))$

- A first cousin is a child of a parent's sibling

  $\forall x,y \; FirstCousin(x,y) \Leftrightarrow \exists p,ps \; Parent(p,x) \wedge Sibling(ps,p) \wedge (Parent(ps,y)$

Other Comments About Quantification

- To say “everyone likes McDonalds”, the following is too broad!
  - $\forall x$, likes(x, McDonalds)
  - Rush’s example: likes (McDonalds, McDonalds)

- We mean: Every one (who is a human) likes McDonalds
  - $\forall x$, person(x) $\Rightarrow$ likes(x, McDonalds)

- Essentially, the left side of the rule declares the class of the variable x

- Constraints like this are often called “domain constraints”