First-Order Logic: Review

Review question

- Interpret the following sentences.
- *(Smoke ▲Fire) ▲(▲Smoke ▲▲Fire)*
- *[(Smoke ◀Heat) ▲Fire] ◀[(Smoke ▲Fire) ▶(Heat ▲Fire)]*
- (P ▲Q) ▲[(R v P) ▲(R v Q)]

- If the rain continues, then the river rises. If rain continues and the river rises, then the bridge will wash out. If continuation of rain will wash the bridge out, then a single road is not sufficient for the town. Either a single road is sufficient for the town or the traffic engineers have made a mistake. Prove the traffic engineers have made a mistake.

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 others
  - **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, more-than ...

User provides

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

FOL Provides

- **Variable symbols**
  - E.g., x, y
- **Connectives**
  - Same as in propositional logic: not ($\neg$), and ($\wedge$), or ($\vee$), implies ($\Rightarrow$), iff ($\Leftrightarrow$)
- **Quantifiers**
  - Universal $\forall$**x** or **(Ax)**
  - Existential $\exists$**x** or **(Ex)**

Sentences: built from terms and atoms

- A **term** (denoting a real-world individual) is a constant symbol, variable symbol, or n-place function of n terms.
  - x and $f(x_1, ..., x_n)$ are terms, where each x_i is a term
  - A term with no variables is a **ground term** (i.e., john, father_of(father_of(john))
- An **atomic sentence** (which has value true or false) is an n-place predicate of n terms (e.g., green(Kermit))
- A **complex sentence** is formed from atomic sentences connected by the logical connectives:
  - $\neg P$, $P \wedge Q$, $P \vee Q$, $P \Rightarrow Q$, $P \Leftrightarrow Q$ where P and Q are sentences

Sentences: built from terms and atoms

- 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

Quantifiers

- **Universal quantification**
  - $(\forall x)P(x)$ means P holds for **all** values of x in domain associated with variable
  - E.g., $(\forall x)$ dolphin(x) $\Rightarrow$ mammal(x)
- **Existential quantification**
  - $(\exists x)P(x)$ means P holds for **some** value of x in domain associated with 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 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"
- Common mistake: represent this EN sentence in FOL as:
  - $(\exists x)$ student(x) $\Rightarrow$ smart(x)
  - Its not acceptable.

Quantifier Scope

- FOL sentences have structure, like programs
- In particular, the variables in a sentence have a scope
- For example, suppose we want to say
  - “everyone who is alive loves someone”
  - $(\forall x)\ alive(x) \Rightarrow (\exists y)\ loves(x,y)$
- Here’s how we scoce the variables

$$(\forall x)\ alive(x) \Rightarrow (\exists y)\ loves(x,y)$$

Scope of x

Scope of y

Quantifier Scope

- **Switching order of universal quantifiers *does not* change the meaning**
  - $(\forall x)(\forall y)P(x,y) \leftrightarrow (\forall y)(\forall x)\, P(x,y)$
  - "Dogs hate cats"
- **You can switch order of existential quantifiers**
  - $(\exists x)(\exists y)P(x,y) \leftrightarrow (\exists y)(\exists x)\, P(x,y)$
  - "A cat killed a dog"
- **Switching order of universals and existentials *does* change meaning:**
  - Everyone likes someone: $(\forall x)(\exists y)$ likes(x,y)
  - Someone likes 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**:
  1. $(\forall x)\ \neg P(x) \leftrightarrow \neg(\exists x)\ P(x)$
  2. $\neg(\forall x)\ P \leftrightarrow (\exists x)\ \neg P(x)$
  3. $(\forall x)\ P(x) \leftrightarrow \neg\ (\exists x)\ \neg P(x)$
  4. $(\exists x)\ P(x) \leftrightarrow \neg(\forall x)\ \neg P(x)$
- Examples
  1. All dogs don't like cats ↔ No dogs like cats
  2. Not all dogs dance ↔ There is a dog that doesn't dance
  3. All dogs sleep ↔ There is no dog that doesn't sleep
  4. There is a dog that talks ↔ Not all dogs can't talk

Quantified inference rules

- Universal instantiation
  - $\forall x\ P(x) \Rightarrow P(A)$
- Universal generalization
  - $P(A) \wedge P(B) \ldots \Rightarrow \forall x\ P(x)$
- Existential instantiation
  - $\exists x\ P(x) \Rightarrow P(F)$

    ← **skolem constant F**

    F must be a "new" constant not appearing in the KB
- Existential generalization
  - $P(A) \Rightarrow \exists x\ P(x)$

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, e.g.:

  $(\forall x)$ eats(John, x) $\Rightarrow$

  eats(John, Cheese18)

- Note that function applied to ground terms is also a constant

  $(\forall x)$ eats(John, x) $\Rightarrow$

  eats(John, contents(Box42))

Existential instantiation (a.k.a. existential elimination)

- From ($\exists$x) P(x) infer P(c), e.g.:
  - ($\exists$x) eats(Mickey, x) $\vdash$ eats(Mickey, Stuff345)
- The variable is replaced by a **brand-new constant** not occurring in this or any sentence in the KB
- Also known as skolemization; constant is a **skolem constant**
- We don't want to accidentally draw other inferences about it by introducing the constant
- Can use this to reason about unknown objects, rather than constantly manipulating existential quantifiers

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

$\exists x \; \forall t \;\; person(x) \land time(t) \Rightarrow can\text{-}fool(x, t)$

You can fool all of the people some of the time

$\forall x \; \exists t \; (person(x) \Rightarrow time(t) \land can\text{-}fool(x, t))$

$\forall x \; (person(x) \Rightarrow \exists t \; (time(t) \land can\text{-}fool(x, t)))$

Note 2 possible readings of NL sentence

All purple mushrooms are poisonous

$\forall x \; (mushroom(x) \land purple(x)) \Rightarrow poisonous(x)$

Translating English to FOL

No purple mushroom is poisonous (two ways)

$\neg\exists x\ purple(x) \wedge mushroom(x) \wedge poisonous(x)$

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

There are exactly two purple mushrooms

$\exists x\ \exists y\ mushroom(x) \wedge purple(x) \wedge mushroom(y) \wedge purple(y) \wedge \neg(x=y) \wedge \forall z\ (mushroom(z) \wedge purple(z)) \rightarrow ((x=z) \vee (y=z))$

Obana is not short

$\neg short(Obama)$

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.

- **Rules for genealogical relations**
  - $(\forall x,y)$ parent(x, y) $\leftrightarrow$ child (y, x)
    $(\forall x,y)$ father(x, y) $\leftrightarrow$ parent(x, y) $\land$ male(x) *;similarly for mother(x, y)*
    $(\forall x,y)$ daughter(x, y) $\leftrightarrow$ child(x, y) $\land$ female(x) *;similarly for son(x, y)*
  - $(\forall x,y)$ husband(x, y) $\leftrightarrow$ spouse(x, y) $\land$ 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) $\land$ 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) $\land$ 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) $\land$ 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) $\land$ ancestor(z, Liz)

Semantics of FOL

- **Domain M**: the set of all objects in the world (of interest)
- **Interpretation I**: 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
- **Define logical connectives**: **~, ^, v, =>, <=>** as in PL
- **Define semantics of ($\forall$x) and ($\exists$x)**
  - ($\forall$x) P(x) is true iff P(x) is true under all interpretations
  - ($\exists$x) P(x) is true iff P(x) is true under some interpretation

- **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

Axioms, definitions and theorems

- **Axioms** are facts and rules that attempt to capture all of the (important) facts and concepts about a domain; axioms can be used to prove **theorems**
  - Mathematicians don’t want any unnecessary (dependent) axioms, i.e. ones that can be derived from other axioms
  - Dependent axioms can make reasoning faster, however
  - Choosing a good set of axioms for a domain is a design problem
- A **definition** of a predicate is of the form “p(X) ↔ …” and can be decomposed into two parts
  - **Necessary** description: “p(x) … ”
  - **Sufficient** description “p(x) … ”
  - Some concepts don’t have complete definitions (e.g., person(x))

Higher-order logic

- FOL only allows to quantify over variables, and variables can only range over objects.
- HOL allows us to quantify over relations
- Example: (quantify over functions)

  "two functions are equal iff they produce the same value for all arguments"

  $\forall f\ \forall g\ (f = g) \Leftrightarrow (\forall x\ f(x) = g(x))$
- Example: (quantify over predicates)

  $\forall r\ \text{transitive}(\ r\ ) \Rightarrow (\forall xyz)\ r(x,y) \wedge r(y,z) \Rightarrow r(x,z))$
- More expressive, but undecidable, in general

Expressing uniqueness

- Sometimes we want to say that there is a single, unique object that satisfies a certain condition
- "There exists a unique x such that king(x) is true"
  - $\exists x\ king(x) \land \forall y\ (king(y) \rightarrow x=y)$
  - $\exists x\ king(x) \land \neg\exists y\ (king(y) \land x\ y)$
  - $\exists!\ x\ king(x)$
- "Every country has exactly one ruler"
  - $\forall c\ country(c) \rightarrow \exists!\ r\ ruler(c,r)$
- Iota operator: "$\iota\ x\ P(x)$" means "the unique x such that p(x) is true"
  - "The unique ruler of Freedonia is dead"
  - $dead(\iota\ x\ ruler(freedonia,x))$

Notational differences

- **Different symbols** for *and*, *or*, *not*, *implies*, ...
  -
  - p v (q ^ r)
  - p + (q * r)
  - etc
- **Prolog**

  cat(X) :- furry(X), meows (X), has(X, claws)
- **Lispy notations**

```
(forall ?x (implies (and (furry ?x)
                         (meows ?x)
                         (has ?x claws))
                    (cat ?x)))
```

Exercise

- Prove using *resolution refutation* that Fido will die, given the axioms:
- Fido is a dog.
- All dogs are animals.
- All animals will die.
- dog(Fido).
- $\forall X\ dog(X) \rightarrow animal(X)$. $\neg dog(X1) \lor animal(X1)$.
- $\forall X\ animal(X) \rightarrow dies(X)$. $\neg animal(X2) \lor dies(X2)$.

Summary

- First order logic (FOL) introduces predicates, functions and quantifiers
- Much more expressive, but reasoning is more complex
  - Reasoning is semi-decidable
- FOL is a common AI knowledge representation language
- Other KR languages (e.g., OWL) are often defined by mapping them to FOL
- FOL variables range over objects
- HOL variables can ranger over functions, predicates or sentences