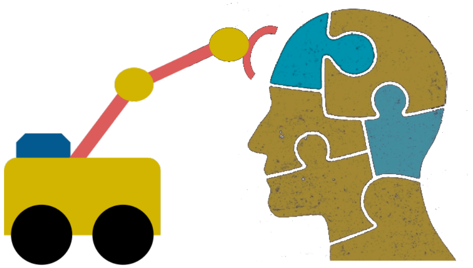


First-Order Logic (FOL) part 2

Overview

- We'll first give some examples of how to translate between FOL and English
- Then look at modelling family relations in FOL
- And finally touch on a few other topics

Translating English to FOL

Every gardener likes the sun

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

All purple mushrooms are poisonous

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

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)

English to FOL: Counting

Use = predicate to identify different individuals

- **There are <u>at least</u> two purple mushrooms**

  $\exists x\ \exists y\ mushroom(x) \wedge purple(x) \wedge mushroom(y) \wedge purple(y) \wedge \neg\mathbf{(x=y)}$

- **There are <u>exactly</u> two purple mushrooms**

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

Saying there are 802 different Pokemon will be hard!

Translating English to FOL

What do these mean?

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

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

Translating English to FOL

What do these mean?

Both English statements are ambiguous

- **You can fool some of the people all of the time**

  There is a nonempty subset of people so easily fooled that you can fool that subset every time*

  For any given time, there is a non-empty subset at that time that you can fool

- **You can fool all of the people some of the time**

  There are one or more times when it's possible to fool everyone*

  Everybody can be fooled at some point in time

* Most common interpretation, I think

Some terms we will need

- **person(x):** True iff x is a person
- **time(t):** True iff t is a point in time
- **canFool(x, t):** True iff x can be fooled at time t

Note: *iff* = *if and only if* = $\leftrightarrow$

Translating English to FOL

You can fool some of the people all of the time

There is a nonempty group of people so easily fooled that you can fool that group every time*

≡ There's (at least) one person you can fool every time

$\exists x \; \forall t \;\; person(x) \land time(t) \rightarrow canFool(x, t)$

For any given time, there is a non-empty group at that time that you can fool

≡ For every time, there's a person at that time that you can fool

$\forall t \; \exists x \;\; person(x) \land time(t) \rightarrow canFool(x, t)$

* Most common interpretation, I think

Translating English to FOL

You can fool all of the people some of the time

There's at least one time when you can fool everyone*

$\exists t\ \forall x\ time(t) \land person(x) \rightarrow canFool(x, t)$

Everybody can be fooled at some point in time

$\forall x\ \exists t\ person(x) \land time(t) \rightarrow canFool(x, t)$

* Most common interpretation, I think

Representation Design

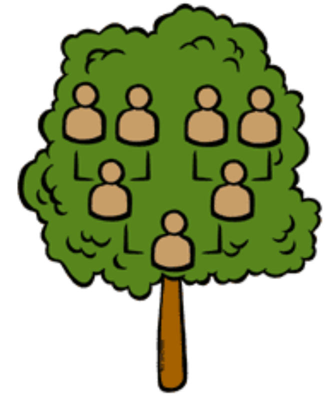
- Many options for representing even a simple fact, e.g., something's color as red, green or blue, e.g.:
  - green(kermit)
  - color(kermit, green)
  - hasProperty(kermit, color, green)
- Choice can influence how easy it is to use
- Last option of representing properties & relations as triples used by modern knowledge graphs
  - Easy to ask: What color is Kermit? What are Kermit's properties?, What green things are there? Tell me everything you know, …

Simple genealogy KB in FOL

Design a knowledge base using FOL that

- Has facts of immediate family relations, e.g., spouses, parents, etc.
- Defines of more complex relations (ancestors, relatives)
- Detect conflicts, e.g., you are your own parent
- Infers relations, e.g., grandparent from parent
- Answers queries about relationships between people

How do we approach this?



- Design an initial ontology of types, e.g.
  - e.g., person, man, woman, male, female
- Extend ontology by defining relations, e.g.
  - spouse, has_child, has_parent
- Add general constraints to relations, e.g.
  - spouse(X,Y) => ~ X = Y
  - spouse(X,Y) => person(X), person(Y)
- Add FOL sentences for inference, e.g.
  - spouse(X,Y) ⇔ spouse(Y,X)
  - man(X) ⇔ person(X) ∧ male(X)

Example: A simple genealogy KB by FOL

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.

Example Axioms

$(\forall x,y)$ parent(x, y) $\leftrightarrow$ child (y, x)

$(\forall x,y)$ father(x, y) $\leftrightarrow$ parent(x, y) $\wedge$ male(x) *;similar for mother(x, y)*

$(\forall x,y)$ daughter(x, y) $\leftrightarrow$ child(x, y) $\wedge$ female(x) *;similar for son(x, y)*

$(\forall x,y)$ husband(x, y) $\leftrightarrow$ spouse(x, y) $\wedge$ male(x) *;similar 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)

$(\forall x,y)$ 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*

Axioms, definitions and theorems

- **Axioms**: facts and rules that capture (important) facts & concepts in a domain; axioms are used to prove **theorems**
  - Mathematicians dislike unnecessary (dependent) axioms, i.e. ones that can be derived from others
  - Dependent axioms can make reasoning faster, however
  - Choosing a good set of axioms is a design problem
- A **definition** of a predicate is of the form "p(X) $\leftrightarrow$ …" and can be decomposed into two parts
  - **Necessary** description: "p(x) $\rightarrow$ …"
  - **Sufficient** description "p(x) $\leftarrow$ …"
  - Some concepts have definitions (e.g., triangle) and some don't (e.g., person)

More on definitions

Example: define father(x, y) by parent(x, y) and male(x)

- **parent(x, y)** is a necessary (but not sufficient) description of father(x, y)

  father(x, y) $\rightarrow$ parent(x, y)

- **parent(x, y) ^ male(x) ^ age(x, 35)** is a sufficient (but not necessary) description of father(x, y):

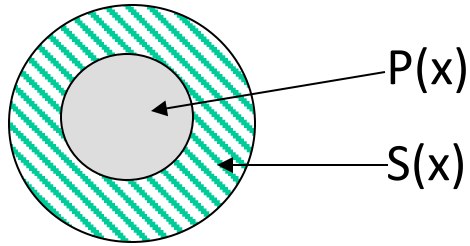
  father(x, y) $\leftarrow$ parent(x, y) ^ male(x) ^ age(x, 35)

- **parent(x, y) ^ male(x)** is a necessary and sufficient description of father(x, y)

  parent(x, y) ^ male(x) $\leftrightarrow$ father(x, y)

More on definitions

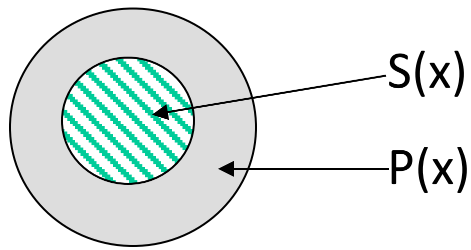
S(x) is a necessary condition of P(x)



 # all Ps are Ss

$(\forall x)\ P(x) \Rightarrow S(x)$

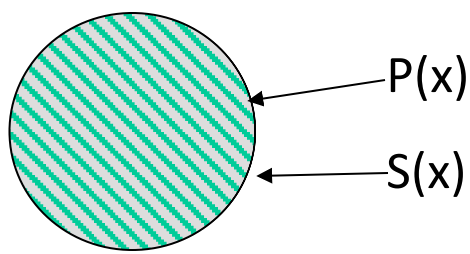
S(x) is a sufficient condition of P(x)



 # all Ps are Ss

$(\forall x)\ P(x) \Leftarrow S(x)$

S(x) is a necessary and sufficient condition of P(x)



 # all Ps are Ss

all Ss are Ps

$(\forall x)\ P(x) \Leftrightarrow S(x)$

Higher-order logic

- FOL only lets us quantify over variables, and **variables can only range over objects**
- HOL allows us to quantify over relations, e.g.

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

  $\forall f \; \forall g \; (f = g) \leftrightarrow (\forall x \; f(x) = g(x))$

- E.g.: (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 reasoning is undecide-able, in general

Examples of FOL in use

- Semantics of W3C's Semantic Web stack (RDF, RDFS, OWL) is defined in FOL
- OWL Full is equivalent to FOL
- Other OWL profiles support a subset of FOL and are more efficient
- The semantics of schema.org is only defined in natural language text
- Wikidata's knowledge graph (and Google's) has a richer schema

FOL Summary

- First order logic (FOL) introduces predicates, functions and quantifiers
- More expressive, but reasoning more complex
  - Reasoning in propositional logic is NP hard, FOL is semi-decidable
- 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 range over functions, predicates or sentences

Fin