Today’s goals

- Practice understanding sentences
  - FOL -> English

- Practice translating sentences
  - English -> FOL

- Start FOL inference
Symbol Names

Compare:
- \( \forall x. \text{MyFriend}(x) \Rightarrow \text{SendBirthdayCard}(x) \)
- \( \forall y. \text{MyFriend}(y) \Rightarrow \text{SendBirthdayCard}(y) \)
- \( \forall x. \text{P0001}(x) \Rightarrow \text{P0002}(x) \)
- \( \forall x. \text{MyEnemy}(x) \Rightarrow \text{SendBirthdayCard}(x) \)

FOL to English

- \( \forall m,c. \text{IsMotherOf}(c) = m \iff \text{IsFemale}(m) \land \text{IsParentOf}(m,c) \)
- \( \forall w,h. \text{IsHusbandOf}(h,w) \iff \text{IsMale}(h) \land \text{IsSpouseOf}(h,w) \)
- \( \forall x. \text{IsMale}(x) \iff \neg \text{IsFemale}(x) \)
- \( \forall p,c. \text{ParentOf}(p,c) \iff \text{ChildOf}(c,p) \)
- \( \forall g,c. \text{IsGrandparentOf}(g,c) \iff \exists p. \text{IsParentOf}(g,p) \land \text{IsParentOf}(p,c) \)
- \( \forall x,y. \text{IsSiblingOf}(x,y) \iff x \neq y \land \exists p. \text{IsParentOf}(p,x) \land \text{IsParentOf}(p,y) \)

Your turn:
- GrandChild, GreatGrandparent, BrotherInLaw, FirstCousin, NthCousin,…
English to FOL

What’s the “right” translation of the sentence “Not all students take both history and biology.”?

NotAllStudentsTakeBothHistoryAndBiology()
NotAllStudentsTakeBoth(History, Biology)
NotAllStudentsTake(History \& Biology)
NotAllStudentsTake(History) \& NotAllStudentsTake(Biology)
¬AllStudentsTakeBoth(History,Biology)
¬∀x. Student(x) ⇒ TakesBoth(History,Biology)
¬∀x. Student(x) ⇒ Takes(x,History) \& Takes(x,Biology)
(¬∀x. Student(x) ⇒ Takes(x,History)) \& (¬∀y. Student(y) ⇒ Takes(y,Biology))

More History and Biology

- Not all students take both History and Biology.
- Only one student failed History.
- Only one student failed both History and Biology.
- The best score in History was better than the best score in Biology.
FOL Inference: Reduction to PL

- Standard PL inference rules sound for FOL as well
  - E.g., modus ponens

\[
\alpha \Rightarrow \beta, \alpha \\
\beta
\]

IsFriend(Arnold)

IsFriend(Arnold) \Rightarrow \text{ShouldSendBirthdayCard}(Arnold)

ShouldSendBirthdayCard(Arnold)

Universal Instantiation

- Replace a universally quantified variable with a ground term

\[
\forall v. \alpha \\
\text{Subst}\left(\{v/g\}, \alpha\right)
\]

UI

\forall x. \text{IsFriend}(x) \Rightarrow \text{ShouldSendBirthdayCard}(x)

MP

\begin{align*}
\text{IsFriend}(Arnold) & \Rightarrow \text{ShouldSendBirthdayCard}(Arnold) \\
\text{IsFriend}(Arnold) & \\
\text{ShouldSendBirthdayCard}(Arnold)
\end{align*}

How should we handle existential quantification?
Existential Instantiation

- Replace an existentially quantified variable with a Skolem constant

\[ \exists v. \alpha \]
\[ \text{Subst}(\{v/Sk\}, \alpha) \]

FOL Inference

- We can now reduce FOL to PL inference:
  - Existentially instantiate everywhere.
  - Universally instantiate with respect to every object
  - Treat resulting terms as propositions
    - E.g., “IsFriend(FatherOf(Arnold))” is just a long name for a proposition.

- Uh oh!
  - Universal instantiation explodes if we have functions!
    - IsFriend(FatherOf(FatherOf(FatherOf(FatherOf(...

\[ \exists x. \text{IsFriend}(x) \]
\[ \text{IsFriend}(\text{F0001}) \]
\[ \forall x. \text{IsFriend}(x) \implies \text{ShouldSendBirthdayCard}(x) \]
\[ \text{ShouldSendBirthdayCard}(\text{F0001}) \]
Herbrand’s Theorem

- If a KB entails A, then there is a proof involving a finite subset of the propositionalized knowledge base.

- I.e., any proof requires only a finite number of f(f(...))’s

- What strategy does that suggest?

Herbrand’s Theorem

- What if the sentence is not entailed?
  - We won’t find a solution at depth 1.
  - Or depth 2.
  - Or depth 3.
  - Or depth 4.
  - ...

- When can we stop searching?

- Entailment of FOL is semidecidable: we can prove entailments, but can’t disprove every non-entailed sentence.
Another problem with FOL $\rightarrow$ PL

- We can have an infinite number of propositions!
  - Proving statements about arithmetic

- Peano Axioms
  - $\text{NatNum}(0)$
  - $\forall n. \text{NatNum}(n) \implies \text{NatNum}(S(n))$

Today’s big idea

- We can reduce FOL to PL
  - Use all of our familiar inference techniques!

- Reducing FOL to PL is often impractical
  - Universal instantiation creates many sentences and propositions.
    - Remember that inference is exponential in number of propositions. (Why?)

  - Functions create infinitely large models
    - (Herbrand’s theorem rescues us a bit)

  - Peano axioms create infinitely many propositions
Next time

- Inference within FOL (without reducing to PL)
  - Forward/backward chaining
  - Resolution