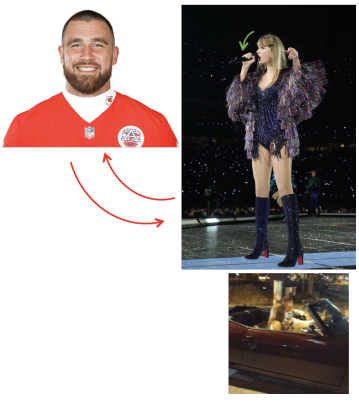
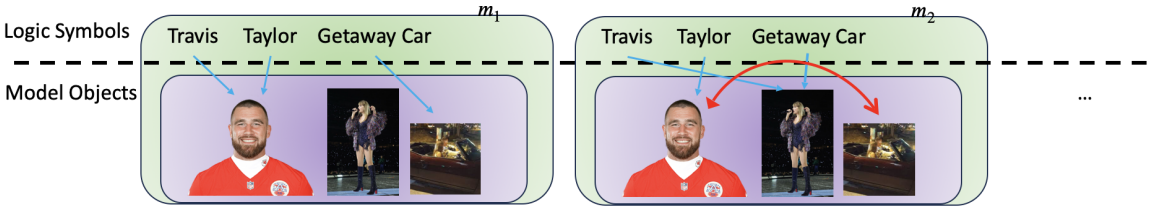
CAS CS 440
Lec 16

Logic V

1. First Order Logic
    a. Prepositional logic has some nice properties
        i. Declarative
            1. Knowledge and inference are separate things
            2. Knowledge (i.e. sentences) declare things to be true/false (no ifs)
            3. Inference is domain independent!
                a. How to create prepositional logics
        ii. Compositionality
            1. Sentence meaning is a function of its parts (so that we can divide and conquer)
    b. Prepositional logic is not concise (problem)
        i. Lots
        ii. And lots
        iii. And lots of sentences
2. FOL Models
    a. Logical languages have models:
        i. Hypothetical worlds
        ii. Links the vocabulary to elements
        iii. Determine the truth of sentence(s)
    b. Models in FOL:
        i. Have objects in them
        ii. Domain of a model = set of objects (domain elements) it contains
        iii. Objects can be related:
            1. Relation is the set of tuples of objects that are related
    c. Dating = {(Taylor, Travis), (Travis, Taylor)} Holding = {(Microphone, Taylor)}, Driving = {(Traivs, Getaway Car)}
    d. 

e. Some Relations are Functions
    i. An object is related to exactly one other object
f. Driving is a function!
    i. Only drive one thing (at a time)
g. Functions in FOL must be total (functions must be returning something always)
    i. Must be a value for every input tuple
    ii. Taylor must be driving something!
    iii. The thing that Taylor is driving must be driving something!
h. Add an "invisible" object for base case
    i. Stands for "nothing"
    ii. Taylor is driving nothing
    iii. Nothing is driving nothing

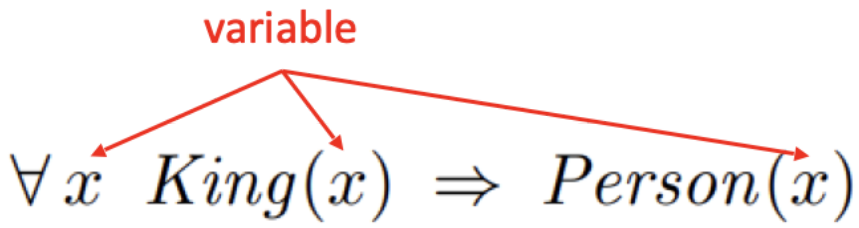
3. FOL Syntax
    a. Types of symbols in the language:
        i. Constants (objects)
        ii. Predicates (relations)
        iii. Functions
    b. Convention in FOL is to start symbols with capital letters (Driving, Taylor, etc.)
    c. Predicate/Function symbols have arity
        i. Number of arguments
4. FOL Syntax & Models
    a. Every model must map symbols to objects (in the model)
        i. Called an "interpretation" of the symbols (can be good or bad)
        ii. Don't need to name all objects in the model
        iii. Can assign multiple symbols to the same object (many to one)
    b. Remember, it's the KBs job to rule out inconsistent models!
        i. Lots of models are garbage
        ii. Models can have relations in them but it is not necessary
    c.



5. Good News
    a. Can bring lots over from prepositional logic
        i. Machinary in prepositional logic was great
        ii. Entailment
        iii. Validity
        iv. Etc.

b. Term
    i. Logical expression that refers to an object
    ii. Constant symbols are terms
        Taylor
        Microphone(Taylor) → not a function
    iii. This is not a function call, it's a name!
        1. Can reason about microphones without defining

6. FOL Syntax
    a. Atomic sentences (atom)
        i. State facts
        ii. Predicate symbol (optionally w arguments)
        iii. Can be true/false in a given model
            1. If relation referred by predicate holds among the objects referred by the arguments
                a. Dating(Taylor, Travis)
            2. Is $(Taylor, Travis) \in Dating$?
    b. Complex sentences
        i. Logical operators
            1. Same as prepositional logic!



        ii. Quantifiers
            1. Express properties of collections (rather than enumerating)
            2. $\forall$ (forall) → paired with implies
            3. $\exists$ (exists) → paired with forall

$$\exists x \quad Microphone(x) \cap Holding(x, Taylor)$$

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

    c.

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

7. FOL Equality
    a. Another kind of atomic sentence
        i. Declare two terms refer to the same object
        ii. $$WonGrammy(Taylor,\ 2016) = 1989\ Album$$
        iii. $$Championship(Travis,\ 2023) = Superbowl\ LVII\ Trophy$$
        iv. Can rule our models (interpretations) by checking these!
        v. States facts about functions
    b. Want to say:
        i. Andrew has two siblings:

$$\exists x, y \quad Sibling(x,\ Andrew) \cap Sibling(y, Andrew)$$

$$\exists x, y \quad Sibling(x,\ Andrew) \cap Sibling(y, Andrew) \cap x \neq y$$

8. FOL Sentences can be Tricky
    a. Want to say:
        i. "Andrew has two siblings: Nathaniel and Elizabeth"
        ii. Does this work?

$$Sibling(Nathaniel,\ Andrew) \cap Sibling(Elizabeth, Andrew)$$

        iii. No! Sentence is true for models where I have only one sibling
            1. (Nathaniel & Elizabeth can be mapped to the same object)
        iv. Doesn't rule out models where I am assigned more than two siblings
        v. Correct sentence:

$$Sibling(Nathaniel,\ Andrew) \cap Sibling(Elizabeth, Andrew) \cap Nathaniel \neq Elizabeth \cap \forall x\ \ Sibling(x,\ Andrew) \Rightarrow (x = Nathaniel \cup x = Elizabeth)$$

    b. Easy to make mistakes (these are database semantics)
        i. Insist that every constant symbol refer to a distinct object (unique-names assumption)
        ii. Atomic sentences not known to be true are false (closed-world assumption)
        iii. model cannot have more objects than constant symbols (domain closure)
9. FOL Semantics
    a. There is no single correct semantics for FOL
    b. Standard FOL semantics:
        i. Infinite many models
        ii. Don't need to know all symbols beforehand
    c. Database Semantics:
        i. Finite number of models
        ii. Need definite knowledge of what the world contains

10. Using FOL in Agents
    a. Add sentences to KB with TELL routine (just like prep logic)
        i. Sentences called assertions
        ii.

$$TELL(KB,\ Holding(Microphone,\ Taylor))$$
$$TELL(KB,\ Dating(Travis,\ Taylor))$$
$$TELL(KB,\ \forall x\ Driving(x,\ Getaway\ Car) \Rightarrow Person(x))$$

    b. Query the KB with ASK routine
        i. Questions asked are called queries/goals
        ii.

$$ASK(KB,\ Holding(Microphone,\ Taylor))$$ Returns true
$$ASK(KB,\ \exists x\ Driving(x,\ Getaway\ Car))$$ Returns true

    c. Sometimes want to know variable values where query is true
        i. ASKVARS routine
        ii.

$$ASK(KB,\ Driving(x,\ Getaway\ Car))$$ Returns [{x/Travis}]
$$ASK(KB,\ \exists y\ Dating(x,\ y))$$ Returns [{x/Travis}, {x/Taylor}]