CAS CS 440
Lec 17

Logic VI

1. Review
   i. Want to say:
      1. “Andrew has two siblings: Nathaniel and Elizabeth”
      2. Does this work?

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

      3. No! Sentence is true for models where I have only one sibling
         a. (Nathaniel & Elizabeth can be mapped to the same object)
      4. Doesn’t rule out models where I am assigned more than two siblings
      5. Correct sentence:

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

   ii. Easy to make mistakes (these are database semantics)
      1. Insist that every constant symbol refer to a distinct object (unique-names assumption) → two separate variables refer to different objects
      2. Atomic sentences not known to be true are false (closed-world assumption)
      3. model cannot have more objects than constant symbols (domain closure)
      4. With these assumptions,

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

         works now
   b. FOL Semantics
      i. There is no single correct semantics for FOL
      ii. Standard FOL semantics:
         1. Infinite many models
         2. Don’t need to know all symbols beforehand (entities can be given names later when we need them)
      iii. Database Semantics: (assumptions are strict)
         1. Finite number of models
         2. Need definite knowledge of what the world contains (cannot give new names in the future)

c. Using FOL in Agents
   i. Add sentences to KB with TELL routine (just like prep logic) → we can either ask or add sentences
      1. Sentences called assertions (adding sentences)
      2. $$TELL(KB,\ Holding(Microphone,\ Taylor))$$ $$TELL(KB,\ Dating(Travis,\ Taylor))$$ $$TELL(KB,\ \forall x\ Driving(x,\ Getaway\ Car) \Rightarrow Person(x))$$
   ii. Query the KB with ASK routine (asking questions)
      1. Questions asked are called queries/goals
      2. $$ASK(KB,\ Holding(Microphone,\ Taylor))$$ Returns true $$ASK(KB,\ \exists x\ Driving(x,\ Getaway\ Car))$$ Returns true
   iii. Sometimes want to know variable values where query is true
      1. ASKVARS routine
      2. $$ASK(KB,\ Driving(x,\ Getaway\ Car))$$ Returns [{x/Travis}] $$ASK(KB,\ \exists y\ Dating(x,\ y))$$ Returns [{x/Travis}, {x/Taylor}]
         a. If the variable x is unbound, I am asking who is the variable x → Travis in this case
         b. The second question → give me the value of x that makes the statement true

2. The Kinship Domain
   a. Lets work through this example:
      i. Add axioms about family tree

$$\forall m,\ c \quad Mother(c) \Leftrightarrow Female(m) \cap Parent(m,c)$$
$$\forall w,\ h \quad Husband(h,w) \Leftrightarrow Male(h) \cap Spouse(h,w)$$
$$\forall x \quad Male(x) \Leftrightarrow \neg Female(x)$$
$$\forall p, c\ Parent(p,c) \Leftrightarrow Child(c,p)$$
$$\forall g,\ c\ Grandparent(g,c) \Leftrightarrow \exists p\ Parent(g,p) \cap Parent(p,c)$$
$$\forall x,\ y\ Sibling(x,y) \Leftrightarrow x \neq y \cap \exists p\ Parent(p,\ x) \cap Parent(p,\ y)$$

   b. Theorem (not an axiom):

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

   c. Not all axioms are definitions:

$\forall x\ Person(x) \Leftrightarrow \ldots ?$ → do not need a definition of a person

   d. Good news!
      i. We don't need a complete definition of Person in order to use it!

ii. We can write partial specifications of properties that every person has

$\forall x\ Person(x) \Rightarrow \ldots$

iii. We can write partial specifications of properties that make something a person (can split things up → divide and conquer)

$\forall x \ldots \Rightarrow Person(x)$

3. Numbers, Sets, and Lists
   a. We can build large KBs from a tiny amount of axioms
   b. Let's talk about natural numbers (non-negative ints)
      i. Natnum predicate (i.e. relation)
      ii. One constant symbol 0 → Naming 0 to be a natural number
      iii. One function symbol S ("successor" function)

      → If n is a natural number, then so is S(n), etc

      $Natnum(0)$
      $\forall n\ \ Natnum(n) \Rightarrow NatNum(S(n))$

      iv. Need some other axioms to constrain S (0 cannot be a successor)

      $\forall n\ \ 0 \neq S(n)$
      $\forall m, n\ \ m \neq n \Rightarrow S(m) \neq S(n)$ → successors of two distinct numbers are distinct

4. Natural Numbers
   a.
   $$Natnum(0)$$
   $$\forall n\ \ Natnum(n) \Rightarrow NatNum(S(n))$$
   $$\forall n\ \ 0 \neq S(n)$$
   $$\forall m, n\ \ m \neq n \Rightarrow S(m) \neq S(n)$$
   b. With these axioms: Addition!
      i. We can define operators! Adding 0 does nothing!

      $$\forall m\ \ Natnum(m) \Rightarrow +(0, m) = m$$
      $$\forall m, n\ \ Natnum(m) \cap Natnum(n) \Rightarrow +(S(m), n) = S(+(m, n))$$

      $(m + 1) + n = (m + n) + 1$

   c. Once we have addition:
      i. Can do subtraction (addition in negative direction)
      ii. Can do multiplication (repeated addition)
      iii. Can do division (repeated subtraction)
   d. All of number theory & cryptography are built from these!