CAS CS 440
Lec 18

Logic VII

1. Inference in FOL
    a. Inference rules for Quantifiers
        i. Take the following sentence

        $$\forall x \;\; King(x) \cap Greedy(x) \Rightarrow Evil(x)$$

        ii. We could now infer:

        $$King(John) \cap Greedy(John) \Rightarrow Evil(John)$$
        $$King\big(Father(John)\big) \cap Greedy\big(Father(John)\big) \Rightarrow Evil\big(Father(John)\big)$$

        iii. Universal Instantiation (UI):
            1. Infer any sentence obtained for substituting a ground term for a variable
        iv. Existential Instantiation (EI)
            1. Infer any sentence obtained for substituting a single new constant symbol
                a. New constant symbol unused anywhere else in the KB
                b. $\exists x \;\; Crown(x) \cap OnHead(x,\; John)$
    b. $$\frac{\forall v \;\; \alpha}{\textsc{Subst}(\{v/g\}, \alpha)} \qquad\qquad \frac{\exists v \;\; \alpha}{\textsc{Subst}(\{v/k\}, \alpha)}$$

        Universal Instantiation Existential Instantiation

    c. Skolem constant

        $$Crown(C_1) \cap OnHead(C_1,\; John)$$

        As long as C_1 does not appear elsewhere in KB!
    d. Once we can convert qualified sentences to non-qualified sentences
        i. Prepositional inference applies!

e. Ex:

$$\forall x \;\; King(x) \cap Greedy(x) \Rightarrow Evil(x)$$
$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard,\ John)$$

   i. Apply UI with substitutions {x/John} and {x/Richard}

   $$King(John) \cap Greedy(John) \Rightarrow Evil(John)$$
   $$King(Richard) \cap Greedy(Richard) \Rightarrow Evil(Richard)$$

   ii. KB is now (essentially) prepositional logic
      1. Can infer Evil(John)

f. This process is called propositionalization

g. Big problem:
   i. what happens if we have a function symbol (like Father)
   ii. This can nest infinitely!
      1.
         - $Father(John)$
         - $Father(Father(John))$
         - $Father(Father(Father(John)))$
         - ...

h. Idea:
   i. Generate all instances with constant symbols (i.e. John, etc.)
   ii. Generate all instances with depth 1 functional symbols (i.e. Father(John))
   iii. Generate all instances with depth 2 functional symbols (i.e. Father(Father(John)))
   iv. …
   v. Stop when you construct a propositional proof of an entailed sentence

i. Trouble
   i. How do we know if a sentence is not entailed?
      1. Semidecidable: we can't!

j.
$$\forall x \;\; King(x) \wedge Greedy(x) \;\Rightarrow\; Evil(x)$$
$$King(John)$$
$$\forall y \;\; Greedy(y)$$

k. Propositionalization isn't great

l. Can we do better?
   i. This is obvious to a human
   ii. Can we make it obvious to the machine?

m. Find a substitution θ that makes each conjunct in premise identical to sentences in KB
    i. Then we can assert conclusion!
n. Suppose our KB was slightly different:
    i. Find substitution for variables in implication sentence and sentences in KB
    ii. {x/John, y/John} satisfies this!

2. Generalized Modus Ponens
    a. Given atomic sentences pi, p'i, q
        i. If there is a substitution θ such that SUBST(θ , p'i) = SUBST(θ , pi) for all i
        ii. $$\frac{p_1', \quad p_2', \quad \ldots, \quad p_n', \quad (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{\textsc{Subst}(\theta, q)}$$ p_1 is
    b. Generalized Modus Ponens lifts Modus Ponens:
        i. Raises modus ponens from prep logic to FOL
    c. Can lift forward/backward chaining & resolution
    d. p_1 is $King(x)$
       p_2 is $Greedy(x)$
       q is $Evil(x)$
       p_1' is $King(John)$
       p_2' is $Greedy(y)$
       θ is $\{x/John, y/John\}$
       $\textsc{Subst}(\theta, q)$ is $Evil(John)$