Session 7

I. *Announcements [5 minutes]*

- Assignment 2 has been extended to Sunday. Get it done early and enjoy the weekend.
- The midterm is coming up and *will not* be extended. It is on 10/18. I'm going to be spending parts of section reviewing for the exam.
- **Feedback**: *I'd like to get feedback from everybody in section on how I can improve sections: 3 things you like, 3 things you dislike.*
    - I've sent out a link to a *Remailer* you can send email to me from anonymously. Also the link is posted on my webpage.
    - I know you're busy with Sudoku's, train tracks, and so forth, but take 10 minutes and send me feedback – it'll make the course better for you and me.
- **Handouts**: I'm now handing out a weekly topic sheet. These sheets should be good for *reviewing* for the exam, but will not replace reading the course material. I'll make them available in class and on my website – are people able to get to my website?

II. *First-Order Logic… briefly [20 minutes]*

- Propositional and First-Order Logic can be the most difficult topics for students to prepare for. Most students do not take any logic course during their entire curriculum and the notation can be daunting.
- Questions
    - How do you feel about first-order logic?
    - More importantly, what questions do you have?
- The Essentials…

Syntax of First Order Logic

- **Objects**: the <u>domain</u> of the model is the set of objects in it ➔ *constant* symbols.
- **Relations**: a set of <u>tuples</u> of objects that are related ➔ *predicate* symbols.
- **Function**: an object can be related to exactly 1 object. Functions in FOL must be <u>total functions</u> that must be defined over the whole domain ➔ *function* symbols.
    - The *arity* of a relation or function is the number of arguments it takes.
- **quantifiers** – allow us to express properties of groups of objects.
    - **Universal** $\forall x \quad P(x)$ **-** for all objects x, P(x) holds. P(x) usually formed with connective ==>.
    - **Existential** $\exists x \quad P(x)$ - there exists an object x, such that P(x) holds. Typically used with the connective AND.
    - ***It can not be overemphasized.*** Universal goes with implies. Existential goes with AND.

Essential Rules to Remember

- $\neg\left[\forall x \quad P(x)\right] \;\equiv\; \exists x \quad \neg P(x)$
- $\neg\left[\exists x \quad P(x)\right] \;\equiv\; \forall x \quad \neg P(x)$
- DeMorgan's Laws:
    - $\neg[A \wedge B] \;\equiv\; \neg A \vee \neg B$
    - $\neg[A \vee B] \;\equiv\; \neg A \wedge \neg B$
- *Definite Clauses* – disjunction of literals of which exactly one is positive.

$$\neg n_1 \vee \ldots \vee \neg n_m \vee p \quad \equiv \quad \underbrace{n_1 \wedge \ldots \wedge n_m}_{body} \Rightarrow \underbrace{p}_{head}$$

Propositionalization – the process of converting a first-order sentences into propositional logic.

- a propositionalized KB is *inferentially equivalent* to the original KB but not *logically equivalent*.
- entailment is preserved thus resolution is *complete*.
- functions cause the number of possible ground terms to be infinite.
- This process begins by converting the KB into CNF.

Conjunctive Normal Form – a conjunction of clauses where each clause is a disjunction of literals.

- *Every sentence in FOL can be converted into an inferentially equivalent CNF sentence.*
- Conversion to CNF:
    1. Eliminate implications using: $p \Rightarrow q \quad \equiv \quad \neg p \vee q$
    2. Move $\neg$ inwards in quantified statements.
    3. **Standardize Variables**: eliminate repeated names.
    4. **Skolemization**: removal of existential quantifiers with Skolem functions (for every enclosing universal quantifier variable) or Skolem constants.
    5. Drop universal quantifiers.
    6. Distribute $\wedge$ over $\vee$.

Unification – the process of finding substitutions that make different logical expressions look identical. Given two sentences, *p* and *q*, UNIFY returns the most general unifier θ, if a unifier exists.

$$\text{UNIFY}(p,q)=\theta \quad | \quad \text{SUBST}(\theta,p)=\text{SUBST}(\theta,q)$$

- standardizing apart – renaming variables to avoid name clashes.
- most general unifier – for every unifiable pair of expressions there is a unique unifier that is more general than any other.
    - A unifier is *more general* than another unifier if the first places fewer restrictions on the values of the variables.
- occur check – when matching a variable against a complex term, one must check whether the variable itself occurs in the term, in which case the unification fails.

Generalized Resolution – an extension of propositional resolution to FOL.

- **Proof by contradiction –** *The goal is negated and added to the KB. If the empty clause {} is derived, the goal has been proved.*
    - Proofs derived in this way are *non-constructive*: they only indicate whether the query is true or false, not what variables make it so.
        1. Restrict query variables to a single binding and backtrack.
        2. Add an **answer literal** as a disjunction with the negated goal. The resulting non-constructive proof will have a disjunction of possible answers instead of an empty clause → multiple answers.
- **refutation completeness –** resolution can be used to confirm or refute any sentence, but it cannot enumerate all true sentences.
- Algorithm outline
    - Begin with <u>propositionalization</u> and conversion to CNF. Each conjunction of clauses is broken into individual clauses.
    - The binary resolution rule can be applied to pairs of clauses (assumingly standardized apart) if they contain complementary literals.
    - The Resolution Inference Rule
        - *First-order literals are complementary if one unifies with the negation of the other*.
        - **binary resolution rule**:

$$\frac{l_1 \vee \ldots \vee l_k \quad m_1 \vee \ldots \vee m_n}{\mathrm{SUBST}\left(\theta, l_1 \vee \ldots \vee l_{i-1} \vee l_{l+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n\right)}$$

$$\text{where } UNIFY\left(l_i, \neg m_j\right) = \theta$$

        - **factoring –** reduces two literals to one if they are unifiable.
    - Together, the *binary resolution rule* and *factoring* are complete.
- **Completeness of Resolution –** *If S is an unsatisfiable set of clauses, then the application of a finite number of resolution steps to S will yield a contradiction.*

III. Exam Question [10 minutes]

- Go over previous exam question