COMP 4475 (Soln) *Assignment Two*

Due: March 15^{th}, Before Class (10AM)

1. (**20 marks**) (Ex.2 in Ch2 of the book KRR) This question involves formalizing the properties of mathematical groups in FOL. Recall that a set is considered to be a group relative to a binary function f and an object e if and only if

   - f is associative;
   - e is an identity element for f, that is for any x, $f(e, x) = f(x, e) = x$; and
   - every element has an inverse, that is, for any x, there is an i such that $f(x, i) = f(i, x) = e$.

   (a) Formalize these as sentences of FOL with two nonlogical symbols, a function symbol f, and a constant symbol e, and show using interpretations that the sentences logically entail the following property of groups: For every x and y, there is a z such that $f(x, z) = y$.

   soln: Note that given x and y, this z should be the function of inverse of x (call it x_i) and y.
   We have three sentences in the set of S:
   - S1: $\forall x \forall y \forall z \Big[f(x, f(y, z)) = f(f(x, y), z)\Big]$,
   - S2: $\forall x \Big[\big(f(e, x) = f(x, e)\big) \wedge f(e, x) = x\Big]$,
   - S3: $\forall x \exists i \Big[f(x, i) = f(i, x) \wedge f(i, x) = e\Big]$.

   Hence, should $\mathcal{I} \models S$ (i.e., $\mathcal{I}$ is a model for S), then
   $\mathcal{I} \models \forall y \Big[f(e, y) = y\Big]$ (due to S2),
   $\mathcal{I} \models \forall x \forall y \exists x_i \Big[f(f(x, x_i), y) = y\Big]$ (due to S3); That is, replacing e with $f(x, x_i)$ where x and x_i are inverse to each other),
   $\mathcal{I} \models \forall x \forall y \exists x_i \Big[f(x, f(x_i, y)) = y\Big]$ (due to S1).
   Let $f(x_i, y) = z$, we have that for every x and y, there is a z such that $f(x, z) = y$, as stated.

   (b) Repeat the entailment proof using Resolution. To do so, you will need to treat equality as a predicate and add to the sentences of part (a) some or all of the axioms of equality (Section 4.2.4 in KRR): reflexibility, symmetry, transitivity. In addition, add the axiom of the substitution of equals for equals, that is for every x, y and z, if $x = y$, then $f(x, z) = f(y, z)$ and $f(z, x) = f(z, y)$.
   soln:
   We have the following axioms in the knowledge base (in clausal forms).
   - S1: $[f(x, f(y, z)) = f(f(x, y), z)]$;
   - S2: $[f(e, x) = f(x, e)]$
   - S3: $[f(x, k(x)) = f(k(x), x)]$, (skolemization to get rid of the existential quantifier, $k(x)$ is actually the inverse of a given x).
   - S4: $[f(x, k(x)) = e]$,
   - S5: $[f(e, x) = x]$,
   - S6: $[x = x]$ (reflexitivity of equality in clausal form),
   - S7: $[x \neq y, y = x]$ (symmetry of equality)
   - S8: $[x \neq y, y \neq z1, x = z1]$ (transitivity)
   - S9: $[x_1 \neq y_1, x_2 \neq y_2, f(x_1, x_2) = f(y_1, y_2)]$ (substitution for functions)
   - S10: $[x \neq y, f(x, z) = f(y, z)]$; $[x \neq y, f(z, x) = f(z, y)]$ (substitution of equals for equals).

We need to prove that

$$KB \models \forall x \forall y \exists z (f(x, z) = y),$$

That is, $KB \cup \neg\Big(\forall x \forall y \exists z (f(x, z) = y)\Big) \models [])$ That is, $KB \cup \neg\Big(\forall x \forall y \exists z (f(x, z) = y)\Big) \vdash [])$
$\neg\Big(\forall x \forall y \exists z (f(x, z) = y)\Big)$ is logically equivalent to

$$\exists x \exists y \forall z \Big(f(x, z) \neq y\Big),$$

and its skolemized version is $\forall z \Big(f(a, z) \neq b\Big)$. In clausal form

- S11: $[f(a, z) \neq b]$

With the following substitutions

- $x = f(a, f(k(a), b))$
- $y = f(f(a, k(a)), b)$
- $z1 = b$
- $z = f(k(a), b)$

The resulting clause for S8 is

- S12: $[f(a, f(k(a), b)) \neq f(f(a, k(a)), b), f(f(a, k(a)), b) \neq b, f(a, f(k(a), b)) = b]$

The resulting clause for S11 is

- S13: $[f(a, f(k(a), b)) \neq b]$

.

S12 resolves with S13 to get

- S14: $[f(a, f(k(a), b)) \neq f(f(a, k(a)), b), f(f(a, k(a)), b) \neq b]$

.

With the following substitutions on S8

- $x = f(f(a, k(a)), b)$
- $y = f(e, b)$
- $z1 = b$

S8 becomes

- S15: $[f(f(a, k(a)), b) \neq f(e, b), f(e, b) \neq b, f(f(a, k(a)), b) = b]$

.

S14 resolves with S15 to get

- S16: $[f(a, f(k(a), b)) \neq f(f(a, k(a)), b), f(f(a, k(a)), b) \neq f(e, b), f(e, b) \neq b]$

.

S1 after substitution of $x/a, y/k(a), z/b$ becomes

- S17: $[f(a, f(k(a), b) = f(f(a, k(a)), b)]$

.

S16 resolves with S17 to get

- S18: $[f(f(a, k(a)), b) \neq f(e, b), f(e, b) \neq b]$

.

S5 after x/b is S19: $[f(e, b) = b]$, which resolves with S18 to get

- S19: $[f(f(a, k(a)), b) \neq f(e, b)]$

.

S10 after $x/f(a, k(a)), y/e, z/b$ to become

- S20: $[f(a, k(a)) \neq e, f(f(a, k(a)), b) = f(e, b)]$

.

S19 resolves with S20 to get

- S21: $[f(a, k(a)) \neq e]$

.

S21 resolves with S4: $[f(a, k(a)) = e]$ to get the empty clause $[\,]$.

2. (**30 marks**) Suppose we use the following KB (where x, y, z are variables and *r1, r2, r3, goal* are constants) to determine whether a particular robot can score.

   (a) $Open(x) \wedge HasBall(x) \rightarrow CanScore(x)$

   (b) $Open(x) \wedge CanAssist(y, x) \wedge HasBall(y) \rightarrow CanScore(x)$

   (c) $PathClear(x, y) \rightarrow CanAsist(x, y)$

   (d) $PathClear(x, z) \wedge CanAssist(z, y) \rightarrow CanAssist(x, y)$

   (e) $PathClear(x, goal) \rightarrow Open(x)$

   (f) $PathClear(y, x) \rightarrow PathClear(x, y)$

   (g) $HasBall(r3)$

   (h) $PathClear(r1, goal)$

   (i) $PathClear(r2, r1)$

   (j) $PathClear(r3, r2)$

   (k) $PathClear(r3, goal)$

   Intuitively, $CanScore(x)$ means x can score on goal. $CanAssist(x, y)$ means there exists some series of passes that can get the ball from x to y. $Open(x)$ means x can shoot on goal directly. And $PathClear(x, y)$ means the path between x and y is clear.

   - Provide a SLD-derivation for the query $CanScore(x)$ in which the answer provided is $r1$.
     soln: Note that for Horn SLD, the clauses generated below from (1) to (9) should all be negative clauses. Negate the query, with x/r1, to have (1) $[\neg CanScore(r1)]$
     - Clause (b) resolves with (1) to get
       $$Clause\ (2) : [\neg Open(r1), \neg CanAssist(r3, r1), \neg HasBall(r3)]$$
     - Clause (g) resolves with (2) to get
       $$Clause\ (3) : [\neg Open(r1), \neg CanAssist(r3, r1)]$$
     - Clause (e) resolves with (3) to get
       $$Clause\ (4) : [\neg PathClear(r1, goal), \neg CanAssist(r3, r1)]$$
     - Clause (h) resolves with (4) to get
       $$Clause\ (5) : [\neg CanAssist(r3, r1]$$
     - Clause (d) resolves with (5) to get
       $$Clause\ (6) : [\neg PathClear(r3, r2), \neg CanAssist(r2, r1)]$$
     - Clause (j) resolves with (6) to get
       $$Clause\ (7) : [\neg CanAssist(r2, r1)]$$
     - Clause (c) resolves with (7) to get
       $$Clause\ (8) : [\neg PathClear(r2, r1)]$$
     - Clause (i) resolves with (8) to get
       $$Clause\ (9) : [\,]$$

- Provide a SLD-derivation for the query $CanScore(x)$ in which the answer provided is $r3$.
  soln: Note again for Horn SLD, the clauses generated below from (1) to (5) should all be negative clauses. Negate the query, with x/r3, to have (1) $[\neg CanScore(r3)]$
  - Clause (a) resolves with (1) to get
  $$Clause\ (2) : [\neg Open(r3), \neg HasBall(r3)]$$
  - Clause (g) resolves with (2) to get
  $$Clause\ (3) : [\neg Open(r3)]$$
  - Clause (e) resolves with (3) to get
  $$Clause\ (4) : [\neg PathClear(r3, goal)]$$
  - Clause (k) resolves with (4) to get
  $$Clause\ (5) : [\,]$$
- How many "distinct" derivations (i.e., involving different pass sequences) are there for the fact $CanScore(r3)$?
  soln: There are infinitely many distinct derivations. As r1 and r2 can pass the ball back forth to each other an unbounded number of times, as can r2 and r3, and we have arbitrary mixtures of such sequences.

3. (**10 marks**) What is the result of the following applications of substitution?
   - $P(x, y, z)\ \ \{x/c, y/f(a)\}$, where a and c are constants, x, y, z are variables.
     soln: $P(c, f(a), c)$.
   - $Q(x, y)\ \ \{x/z, y/z\}$, where x,y,z are all variables.
     soln: $Q(z, z)$.

4. (**10 marks**) Find a most general unifier (if one exists) of the following pairs.
   - $P(y, a, b, y)$ and $P(c, f, g, f)$ where a, b and c are constants, and f,g,y are variables.
     soln: Can not be unified. To unify, it will have to be the case $y = c$. But this means the first atom to become $P(c, a, b, c)$. Consequently, it requires that $f = c$ and $f = a$, hence $a = c$, but a and c are two different constant.
   - $P(f(x), r(x), c)$ and $P(w, r(q), q)$, where c is a constant, and x, w, q are variables.
     soln: $P(f(c), r(c), c)$.

5. (**30 marks**) (Ex.4 in Ch5 of the book KRR) In this question, we will explore the semantic properties of propositional Horn clauses. For any set of clauses S, define $\mathcal{I}_S$ to be the interpretation that satisfies an atom p if and only if $S \models p$.
   - Show that if S is a set of positive Horn clauses, then $\mathcal{I}_S \models S$.

     soln: Formally, we define $\mathcal{I}_S$ as that for any atom p, and any KB S
     $$\mathcal{I}_S \models p \text{ iff } S \models p \ \ \text{(call it } \star)$$
     Now S is a knowledge base of positive Horn clauses. We assume $\mathcal{I}_S \not\models S$. That is we have at least one $h \in S$ such that $\mathcal{I}_S \not\models h$ (let h be in the form of $[\neg a_1, \neg a_2, \ldots, \neg a_n, b]$). This implies that $\mathcal{I}_S \models \neg h$ (i.e., $\mathcal{I}_S \models a_1$, and $\mathcal{I}_S \models a_2$, and $\ldots$, and $\mathcal{I}_S \models a_n$, and $\mathcal{I}_S \models \neg b$). From (*), we know we also have $S \models a_1$, and $S \models a_2$, and $\ldots$, and $S \models a_n$, and $S \models \neg b$. From the fact that $h \in S$, together with the facts that all a_is can be entailed by S too, b must also be entailed by S, this is in contradiction with the derived fact $S \models \neg b$. Hence the assumption must be invalid, i.e., $\mathcal{I}_S \models S$.

- Give an example of a set of clauses S where $\mathcal{I}_S \not\models S$.

  soln: Let S be $\{[a], [b], [\neg a, \neg b, c, d]\}$. Let $\mathcal{I}_S$ be defined as a, b true, and c, d false. In this case, $\mathcal{I}_S \models p$ iff $S \models p$ holds. However, $\mathcal{I}_S \not\models [\neg a, \neg b, c, d]$, i.e., $\mathcal{I}_S \not\models S$.

- Suppose that S is a set of positive Horn clauses and that c is a negative Horn clause. Show that if $\mathcal{I}_S \not\models c$ then $S \cup \{c\}$ is unsatisfiable.

  soln: Define $c = [\neg p_1, \neg p_2, \ldots, \neg p_n]$, where p_is are atoms. Now that $\mathcal{I}_S \not\models c$, we have $\mathcal{I}_S \models p_i$ for all is from 1 to n. But this means, from (*) $S \models p_i$, for all is. Hence, $S \models \neg c$. In other words, $S \cup \{c\}$ is unsatisfiable.

- Suppose that S is a set of positive Horn clauses and that T is a set of negative ones. Using part (c), show that if $S \cup \{c\}$ is satisfiable for every $c \in T$, then $S \cup T$ is satisfiable also.

  soln: Continuing from the previous question, if $S \cup \{c\}$ is satisfiable, we have $\mathcal{I}_S \models c$. Hence, for all i, we have that if $S \cup \{c_i\}$ is satisfiable, then $\mathcal{I}_S \models c_i$ for all i, i.e., $\mathcal{I}_S \models T$, together with $\mathcal{I}_S \models S$, we know that $\mathcal{I}_S$ is an interpretation to satisfy $S \cup T$.

- In the propositional case, the normal Prolog interpreter can be thought of as taking a set of positive Horn clauses S (the program) and a single negative clause c (the query) and determining whether or not $S \cup \{c\}$ is satisfiable. Use part (d) to conclude that Prolog can be used to test the satisfiability of an arbitrary set of Horn Clauses.

  soln: Divide any set of Horn clauses into two parts, S and T, where S contains only positive Horns, and T only negative Horns. Using the method in the previous section, we can check whether $S \cup T$ is satisfiable. Note that if $S \cup c_i$ is unsatisfiable, then $S \cup T$ is unsatisfiable.