CSCE 420 – Fall 2025

Homework 4 (HW4)

Name: Ayushi Sharma

1. Translate the following sentences into First-Order Logic. Remember to break things down to simple concepts (with short predicate and function names), and make use of quantifiers. For example, don't say "tasteDelicious(someRedTomatos)", but rather: "$x tomato(x)^red(x)^ taste(x,delicious)". See the lecture slides for more examples and guidance.

- **bowling balls are sporting equipment**

$\forall x(BowlingBall(x) \rightarrow SportEquipment(x))$

- **horses are faster than frogs (there are many ways to say this in FOL; try expressing it this way: "all horses have a higher speed than any frog")**

$\forall h \forall f(Horse(h) \land Frog(f) \rightarrow GT(speed(h), speed(f)))$.

- **all domesticated horses have an owner**

$\forall x(Horse(x) \land Domesticated(x) \rightarrow \exists y Owner(y, x))$

- **the rider of a horse can be different than the owner**

$\forall x(Horse(x) \rightarrow \exists r \exists o(Rider(r,x) \land Owner(o, x) \land r \neq o))$.

- **a finger is any digit on a hand other than the thumb**

$\forall x(Finger(x) \leftrightarrow (Digit(x) \land \exists h(Hand(h) \land On(x, h)) \land \neg Thumb(x)))$

- **an isosceles triangle is defined as a polygon with 3 edges connected at 3 vertices, where 2 (but not 3) edges have the same length**

$\forall t(IsoscelesTriangle(t) \leftrightarrow Polygon(t) \land Has3Edges(t) \land Has3Vertices(t) \land \exists e1,e2,e3(EdgesOf(t,e1,e2,e3) \land (len(e1) = len(e2) \lor len(e1) = len(e3) \lor len(e2) = len(e3)) \land \neg(len(e1) = len(e2) \land len(e2) = len(e3))))$.

2. Convert the following first-order logic sentence into CNF:

$$\boldsymbol{\forall x\ person(x) \wedge [\exists z\ petOf(x, z) \wedge [\forall y\ petOf(x, y) \rightarrow dog(y)]] \rightarrow doglover(x)}$$

Keep in mind that the quantifiers have *lower* precedence than all the other operators in FOL sentences.

Eliminate:

$\forall x(person(x) \land \exists z petOf(x,z) \land \forall y(\neg petOf(x,y) \lor dog(y)) \rightarrow doglover(x))$

Outer Implication $A \rightarrow B \equiv \neg A \lor B$:

Let $A := person(x) \land \exists z\ petOf(x,z) \land \forall y(\neg petOf(x,y) \lor dog(y))$

$\forall x(\neg A \lor doglover(x))$

Push the negation through A:

$\forall x(\neg person(x) \lor \neg \exists z petOf(x,z) \lor \neg \forall y(\neg petOf(x,y) \lor dog(y)) \lor doglover(x))$

Flip Quantifiers Under Negation:

$\forall x(\neg person(x) \lor \forall z \neg petOf(x,z) \lor \exists y \neg(\neg petOf(x,y) \lor dog(y)) \lor doglover(x))$

$\neg(\neg P \lor Q) \equiv P \land \neg Q$:

$\forall x(\neg person(x) \lor \forall z \neg petOf(x,z) \lor \exists y(petOf(x,y) \land \neg dog(y)) \lor doglover(x))$

Pull quantifiers to the front :

$\forall x \forall z \exists y\ (\neg person(x) \lor \neg petOf(x,z) \lor (petOf(x,y) \land \neg dog(y)) \lor doglover(x))$

Distribute $\lor$ over $\land$: $D \lor (A \land B) \equiv (D \lor A) \land (D \lor B)$:

Let $D := \neg person(x) \lor \neg petOf(x,z) \lor doglover(x)$

$\forall x \forall z \exists y\ ((D \lor petOf(x,y)) \land (D \lor \neg dog(y)))$

Clauses (CNF):

1. **$\neg person(x) \lor \neg petOf(x,z) \lor doglover(x) \lor petOf(x, f(x, z))$**
2. **$\neg person(x) \lor \neg petOf(x,z) \lor doglover(x) \lor \neg dog(f(x, z))$**

3. Determine whether or not the following pairs of predicates are **unifiable**. If they are, give the most-general unifier and show the result of applying the substitution to each predicate. If they are not unifiable, indicate why. Capital letters represent variables; constants and function names are lowercase. For example, 'loves(A,hay)' and 'loves(horse,hay)' are unifiable, the unifier is u={A/horse}, and the unified expression is 'loves(horse,hay)' for both.

1. **owes(owner(X),citibank,cost(X))** **owes(owner(ferrari),Z,cost(Y))**

→ **Unifiable**
→ Most-general unifier: **{ X/ferrari, Z/citibank, Y/ferrari }**
→ Unified Predicate: **owes(owner(ferrari), citibank, cost(ferrari))**

2. **gives(bill, jerry, book21)** **gives(X,brother(X),Z)**

→ **Not unifiable**
→ Requires **jerry = brother(bill),** which can't be done by substitution because a constant can't equal a distinct function term

3. **opened(X,result(open(X),s0)))** **opened(toolbox,Z)**

→ **Unifiable**
→ Most-general unifier: **{ X/toolbox, Z/result(open(toolbox), s0) }**
→ Unified predicate: **opened(toolbox, result(open(toolbox), s0))**

4. Consider the following situation:

Marcus is a Pompeian.
All Pompeians are Romans.
Ceasar is a ruler.
All Romans are either loyal to Caesar or hate Caesar (but not both).
Everyone is loyal to someone.
People only try to assassinate rulers they are not loyal to.
Marcus tries to assassinate Caesar.

a) Translate these sentences to First-Order Logic.

i) Marcus is a Pompeian. → **Pompeian(marcus)**
ii) All Pompeians are Romans. → **∀x[Pompeian(x) → Roman(x)]**
iii) Ceasar is a ruler. → **Ruler(caesar)**
iv) All Romans are either loyal to Caesar or hate Caesar (but not both). →
∀x[Roman(x) → (Loyal(x, caesar) ∨ Hates(x, caesar))]
∀x[Roman(x) → ¬(Loyal(x, caesar) ∧ Hates(x, caesar))]
v) Everyone is loyal to someone. → **∀x∃y Loyal(x, y)**

vi) People only try to assassinate rulers they are not loyal to. →
∀x∀y [Try(x, y) → (Ruler(y) ∧ ¬Loyal(x, y))]
vii) Marcus tries to assassinate Caesar. → **Try(marcus,caesar)**

b) Prove that *Marcus hates Caesar* using Natural Deduction. Label all derived sentences with the ROI and which prior sentences and unifier were used.

Goal: Hates(marcus, caesar)

1. Pompeian(marcus)
2. ∀x(Pompeian(x)→Roman(x))
3. From 2 by UI with {x/marcus}: Pompeian(marcus) → Roman(marcus)
4. From 1,3 by MP: Roman(marcus)
5. Ruler(caesar)
6. Try(marcus, caesar)
7. ∀x∀y [Try(x,y)→(Ruler(y)∧¬Loyal(x,y))]
8. From 7 by **UI** with {x/marcus,y/caesar}: Try(marcus, caesar) → (Ruler(caesar) ∧ ¬Loyal(marcus, caesar))
9. From 6,8 by MP: Ruler(caesar) ∧ ¬Loyal(marcus,caesar)
10. From 9 by ∧-E: ¬Loyal(marcus, caesar)
11. ∀x [Roman(x) → (Loyal(x, caesar) ∨ Hates(x, caesar))]
12. From 11 by **UI** with {x/marcus}: Roman(marcus) → (Loyal(marcus, caesar) ∨ Hates(marcus, caesar))
13. From 4,12 by MP: Loyal(marcus, caesar) ∨ Hates(marcus, caesar)
14. From 10,13 by disjunctive syllogism: Hates(marcus, caesar)

c) Convert all the sentences into CNF

1. Pompeian(marcus)
2. ¬Pompeian(x) ∨ Roman(x)
3. Ruler(caesar)
    4a. ¬Roman(x) ∨ Loyal(x,caesar) ∨ Hates(x,caesar)
    4b. ¬Roman(x) ∨ ¬Loyal(x,caesar) ∨ ¬Hates(x,caesar) (“not both”)
4. ∀x ∃y Loyal(x,y) → Skolemize y=f(x): Loyal(x,f(x))
    6a. From Try(x,y)→Ruler(y): ¬Try(x,y) ∨ Ruler(y)
    6b. From Try(x,y)→¬Loyal(x,y): ¬Try(x,y) ∨ ¬Loyal(x,y)
5. Try(marcus, caesar)

d) Prove that *Marcus hates Caesar* using Resolution Refutation.

Add the negated goal:

Ḡ. ¬Hates(marcus,caesar)

Clauses:

C1. Pompeian(marcus)

C2. ¬Pompeian(x) ∨ Roman(x)

C3. Ruler(caesar)

C4. ¬Roman(x) ∨ Loyal(x,caesar) ∨ Hates(x,caesar)

C5. ¬Roman(x) ∨ ¬Loyal(x,caesar) ∨ ¬Hates(x,caesar) (unused)

C6. Loyal(x,f(x)) (unused)

C7. ¬Try(x,y) ∨ Ruler(y)

C8. ¬Try(x,y) ∨ ¬Loyal(x,y)

C9. Try(marcus,caesar)

C10. ¬Hates(marcus,caesar) (from Ḡ)

Derivation:

R1. From **C1** and **C2** on Pompeian with {x/marcus}: Roman(marcus)

R2. From **C9** and **C8** on Try with {x/marcus, y/caesar}: ¬Loyal(marcus, caesar)

R3. From **C4** and **R1** on Roman with {x/marcus}: Loyal(marcus, caesar) ∨ Hates(marcus, caesar)

R4. From **R3** and **R2** on Loyal(marcus, caesar): Hates(marcus, caesar)

R5. From **R4** and **C10** on Hates(marcus, caesar): empty clause

So, by refutation, Hates(marcus, caesar) is entailed.

5. Write a KB in First-Order Logic with rules/axioms for…

a. **Map-coloring** – every state must be exactly 1 color, and adjacent states must be different colors. Assume possible colors are states are defined using unary predicate like color(red) or state(WA). To say a state has a color, use a binary predicate, e.g. 'color(WA,red)'.

state(s): s is a state
color(c): c is a valid color
hasColor(s,c): state s has color c
adjacent(s_1,s_2): states s_1 and s_2 share a border

Axioms:

1. Every state has at least one color: $\forall s\ (state(s) \rightarrow \exists c\ (color(c) \land hasColor(s, c)))$
2. Every state has at most one color: $\forall s\ \forall c_1\ \forall c_2\ ((hasColor(s, c_1) \land hasColor(s, c_2)) \rightarrow c_1 = c_2)$
3. Adjacent states must have different colors: $\forall s_1\ \forall s_2\ \forall c\ ((adjacent(s_1, s_2) \land hasColor(s_1, c) \land hasColor(s_2, c)) \rightarrow \bot)$

b. **Sammy's Sport Shop** – include implications of facts like obs(1,W) or label(2,B), as well as constraints about the boxes and colors. Use predicate 'cont(x,q)' to represent that box x contains tennis balls of color q (where q could be W, Y, or B).

obs(x,q): box x was observed to have a ball of color q
label(x,q): the label on box x says color q
cont(x,q): box x actually contains balls of color q
box(x): x is a valid box
color(q): $q \in \{ W, Y, B \}$

Axioms:

1. Each box contains exactly one color

   (a) At least one : $\forall x\ (box(x) \rightarrow \exists q\ cont(x, q))$

   (b) At most one : $\forall x\ \forall q_1\ \forall q_2\ ((cont(x, q_1) \land cont(x, q_2)) \rightarrow q_1 = q_2)$

2. Observations are correct: $\forall x\ \forall q\ (obs(x, q) \rightarrow cont(x,q))$
3. Labels are wrong: $\forall x\ \forall q\ (label(x, q) \rightarrow \neg cont(x, q))$

4. If you want the version where labels are correct, flip the previous rule to: $\forall x\ \forall q\ (label(x, q) \rightarrow cont(x, q))$

c. **Wumpus World** - (hint start by defining a helper concept ‘adjacent(x,y,p,q)’ which defines when a room at coordinates (x,y) is adjacent to another room at (p,q). Don’t forget rules for ‘stench’, ‘breezy’, and ‘safe’.

room(x,y): coordinate (x,y) is a room
adjacent(x,y,p,q): rooms (x,y) and (p,q) share a wall
pit(x,y): there is a pit in (x,y)
wumpus(x,y): the Wumpus is in (x,y)
breezy(x,y): room (x,y) is breezy
stench(x,y): room (x,y) smells of a Wumpus)
safe(x,y): room (x,y) is safe

Adjacency: $\forall x\ \forall y\ \forall p\ \forall q\ (adjacent(x,y,p,q) \leftrightarrow ((x = p \land |y - q| = 1) \lor (y = q \land |x - p| = 1)))$

Rules:

1. A room is breezy iff some adjacent room has a pit: $\forall x\ \forall y\ (breezy(x, y) \leftrightarrow \exists p\ \exists q\ (adjacent(x, y, p, q) \land pit(p, q)))$
2. A room has a stench iff some adjacent room has the Wumpus: $\forall x\ \forall y\ (stench(x, y) \leftrightarrow \exists p\ \exists q\ (adjacent(x, y, p, q) \land wumpus(p, q)))$
3. A room is safe iff it has neither pit nor Wumpus: $\forall x\ \forall y\ (safe(x, y) \leftrightarrow (\neg pit(x, y) \land \neg wumpus(x, y)))$

d. **4-Queens** – assume row(1)…row(4) and col(1)…col(4) are facts; write rules that describe configurations of 4 queens such that none can attack each other, using ‘queen(r,c)’ to represent that there is a queen in row r and col c.

Facts: row(1)…row(4); col(1)…col(4)

Predicate: queen(r, c) – queen at row r, col c

Rules:

1. Each row has **exactly one** queen:

$$\forall r\ (row(r) \rightarrow \exists!\ c\ queen(r, c))$$

(Existence + Uniqueness: $\forall r\ (row(r) \rightarrow \exists c\ queen(r, c))$ and $\forall r\ \forall c_1\ \forall c_2\ ((queen(r, c_1) \wedge queen(r, c_2))$

$\rightarrow c_1 = c_2))$

2. No two queens share the same column: $\forall r_1\ \forall r_2\ \forall c\ ((queen(r_1, c) \wedge queen(r_2, c) \wedge r_1 \neq r_2) \rightarrow \bot)$
3. No two queens share the same diagonal: $\forall r_1\ \forall c_1\ \forall r_2\ \forall c_2\ ((queen(r_1, c_1) \wedge queen(r_2, c_2) \wedge (r_1 \neq r_2 \vee c_1 \neq c_2) \wedge (|r_1 - r_2| = |c_1 - c_2|)) \rightarrow \bot)$