

CSCE 420, Assignment A5

due: **Tues, Apr 26, 3:55pm** – **THIS IS A HARD DEADLINE – NO GRACE PERIOD - LATE ASSIGNMENTS TURNED IN AFTER Tues, Apr 26, 3:55pm WILL BE WORTH 0.**
turn-in answers as a Word document (A5.docx) and commit/push it to your TAMU github

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: " $\exists x \text{ tomato}(x) \wedge \text{red}(x) \wedge \text{taste}(x, \text{delicious})$ ". See the lecture slides for more examples and guidance.

- Tomatos are either a fruit or vegetable.
- Some mushrooms are poisonous.
- No student at TAMU likes every class.
- Every king has a crown and some subjects. (how many is not specified)
- An isosceles triangle is a triangle with 2 sides of equal length (but not 3).
- The left front tire of Henry's car is flat.
- All disk drives are electronic; they are either mechanical (HDDs) or solid-state (SSDs). Solid-state drives are faster than mechanical drives, in terms of read-access time (e.g. ms per byte), but SSDs are also more expensive (in dollars per Gb).
- All laptops sold by Dell in 2022 have at least 4GB of RAM and cost at least \$1000.
- John's favorite Sci-Fi movie is Star Wars (hint: instead of saying *favoriteSciFiMovie(John,StarWars)* which is too complex, break it down into simpler concepts such as predicate 'likes(P,M)', and function 'score(P,M)', which denotes P's internal rating system for things (e.g. 1-100). Ensure you are comparing SciFi movies, because a person might use a different rating system for other items, such as toasters.

2. 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. Terms that are variables are in capital letters; constants and function names are lowercase. For example, 'loves(A,hay)' and 'loves(horse,hay)' are unifiable, the unifier is $u = \{A/\text{horse}\}$, and the unified expression is 'loves(horse,hay)' for both.

- | | | | | | |
|----|---------------------|------------------|-----|---|------------------|
| 0) | loves(A,hay) | loves(horse,hay) | yes | $u = \{A/\text{horse}\}$ | loves(horse,hay) |
| 1) | p(a,X) | p(b,X) | no | reason: regardless of X, a and b conflict | |
| a) | p(a, X, f(g(Y))) | p(Z, f(Z), f(U)) | | | |
| b) | q(f(a), g(X)) | q(Y, Y) | | | |
| c) | r(f(Y), Y,X) | r(Z, f(a), f(V)) | | | |
| d) | p(a, Y, f(X)) | p(Z, f(b), f(b)) | | | |
| e) | q(g(f(a)), g(X), Z) | q(Y, Y, f(W)) | | | |
| f) | p(x,f(X),X) | p(Y,f(a),b) | | | |
| g) | q(f(a,a), V ,Z) | q(X, f(X,X), Y) | | | |

3. Caesar

Consider the following situation:

Marcus is a Pompeian.

All Pompeians are Romans.

Caesar 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.
- b) Prove that **Marcus hates Caesar** using Natural Deduction. Label all derived sentences with the ROI and prior sentences and unifier used.
- c) Convert all the sentences into CNF
- d) Prove that **Marcus hates Caesar** using Resolution Refutation.

4. Writing KBs in FOL

Write a KB (rules/axioms) in FOL 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)`'.
- 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`).
- 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'.
- 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`.

Don't forget to quantify all your variables.