CSE-6421: Assignment #1

1. (5 points) **First-order Predicate Calculus.** *In theory, at least.*

   Consider first-order theory (program) $\mathcal{T}$.

   a. Is it possible that $\mathcal{T} \models \mathit{student}\,(\mathit{parke})$ *and* $\mathcal{T} \models \neg \mathit{student}\,(\mathit{parke})$?
   If it is possible, provide an example $\mathcal{T}$ such that it is the case. Otherwise, explain why it is not possible.

   b. Is it possible that $\mathcal{T} \not\models \mathit{student}\,(\mathit{parke})$ *and* $\mathcal{T} \not\models \neg \mathit{student}\,(\mathit{parke})$?
   If it is possible, provide an example $\mathcal{T}$ such that it is the case. Otherwise, explain why it is not possible.

   Consider a datalog database $\mathcal{D}$.

   c. Is it possible that $\mathcal{D} \models \neg \mathit{student}\,(\mathit{parke})$?
   Why or why not?

   d. Do $\mathcal{D} \vdash \neg \mathit{student}\,(\mathit{parke})$ $\mathcal{D} \not\vdash \mathit{student}\,(\mathit{parke})$ mean the same thing?
   Why or why not?

   e. Is every first-order propositional theory ("program") equivalent to some CNF propositional theory ("program")?
   If no, show an example of a first-order theory that cannot be written in CNF. If yes, are there any drawbacks to representing first-order theories in CNF?

2. (5 points) **Rules.** *Bar-flies.* (Thanks to Zahir Tari.)

   Suppose that we have the following predicates.

   - *frequents* (*Drinker, Bar*): The drinker frequently visits this bar.
   - *serves* (*Bar, Beer*): The bar serves this type of beer.
   - *likes* (*Drinker, Beer*)The drinker likes this type of beer.

   Define the following predicates via rules using the predicates above (and any that you define).

   a. *happy* (*D*): The drinker *D* frequents at least one bar which serves a beer that he / she likes.

   b. *very_happy* (*D*): Every bar that the drinker *D* visits serves at least one beer he / she likes.

   c. *should_visit* (*D, B*): The bar *B* serves at least one beer that the drinker *D* likes.

   d. *sad* (*D*): No bar that the drinker *D* visits serves a beer that he / she likes.

   e. *very_sad* (*D*): No bar serves a beer that he / she likes.

   You may assume that each drinker at least frequents one bar. Make certain that your rules are safe.

3. (5 points) **Queries in Datalog & Datalog¬.** *Enrol now in Datalog U.!* EXERCISE

Consider the following schema.

student(<u>s#</u>, sname, dob, d#)
  FK (d#) refs **dept** // Student's major
prof(<u>p#</u>, pname, d#)
  FK (d#) refs **dept** // Professor's home deparmtent
dept(<u>d#</u>, dname, building, p#)
  FK (p#) refs **prof** // Department's chair
course(<u>d#</u>, <u>no</u>, title)
  FK (d#) refs **dept** // Course offered by this deparmtent
class(<u>d#</u>, <u>no</u>, <u>term</u>, <u>year</u>, <u>section</u>, room, time, p#)
  FK (d#, no) refs **course** // Class is an offering of this course
  FK (p#) refs **prof** // Instructor of class
enrol(<u>s#</u>, <u>d#</u>, <u>no</u>, <u>term</u>, <u>year</u>, <u>section</u>, grade)
  FK (s#) refs **student** // This student is enrolled in
  FK (d#, no, term, year, section) refs **class** // this class

'FK' above stands for *foreign key.* These indicate foreign-key constraints in the schema.

Write the following queries in Datalog (and Datalog¬). You may use auxiliary predicates and rules. (You may reuse auxiliary predicates and rules in following sub-questions.)

A common convention is to use '_' as a variable name when the variable is unimportant for the query; e.g., *class* (*D, N,* _, _, _, _, _, _). By convention, two occurrences of '_' are *different* variables and may take on different values (even though they seem to have the same "name"). You may find this convention useful.

Be careful that all your rules are *safe*, including rules that you write that use negation.

a. Which students have taken some course twice?

b. Which students have taken a course with *a* department chair?

   Note that a professor may teach classes outside of his or her department. Also note that a student may take classes in a department outside of his or her major's department.

c. Which students have never taken a course in his or her major (dept)?

d. Which students have taken all of the courses offered by a department?

e. Which students have taken at least five courses in their major (dept)?

   You shall need to use arithmetics (e.g., '$\neq$', '$<$') here. Assume that course numbers (no) can be compared; e.g., $M < N$. Use the predicate *is* to equate numbers; e.g., J is $I + 1$.

4. (5 points) **Logic Program.** *The game of stones.*

   The game of *stones* is played as follows. It is played by two players who alternate turns. The game starts with an odd, finite number (k) of stones. When it is one's turn, one may remove one, two, or three stones from the collection. If there is only one stone remaining and it is one's turn, one must take the last stone. The game ends when no stones remain. The player who has taken an odd number of stones in total throughout the game is the winner. (Since we always start with an odd number of stones, the *other* player must have taken an *even* number. So both players do not win!)

   Model this as a logic program with a *win* predicate and the `not` operator.