CS432/532 Homework 3

1. Consider the following relation schemas transformed from the ER diagram for the Student Registration System (note that some changes have been made due to the creation of a single attribute key for Classes):

Students(sid, firstname, lastname, status, gpa, email)
Courses(dept_code, course#, title, credits, deptname)
Prerequisites(dept_code, course#, pre_dept_code, pre_dept_course#)
Classes(classid, dept_code, course#, sect#, year, semester, start_time, end_time, limit, size, classroom, capacity, fid) /* note: classid is added to serve as a single attribute key */
Classes_days(classid, day)
Faculty(fid, name, rank, office, email, deptname)
Departments(deptname, chair, office)
Registration(sid, classid, lgrade, ngrade)
Student_majors(sid, deptname)

Do the following for each relation schema:

(a) [20%] Identify all non-trivial functional dependencies. Don't make unrealistic assumptions about the data. Should use the union rule to combine the functional dependencies as much as possible. Furthermore, if a functional dependency is redundant (i.e., it can be derived from the ones you keep), it does not need to be included.

(b) [20%] Determine whether the schema is in 3NF or in BCNF. Briefly explain your answer.

(a) [20%] For each schema that is not in 3NF, decompose it into 3NF schemas using AlgorithmLLJD-DPD-3NF. Show the result after each step of the algorithm. Are they decomposed schemas in BCNF? Justify your answer.

2. Prove or disprove the following rules:

   (a) [10%] {X → Y, Z → W} |= {XZ → YW}
   (b) [10%] {X → Y, YZ → W} |= {XY → W}

   When proving a rule, use Armstrong's Axioms (i.e., reflexivity rule, augmentation rule, and transitivity rule) only. To disprove a rule, construct a relation with appropriate attributes and tuples such that the tuples of the relation satisfy the functional dependencies on the left of the rule but do not satisfy the functional dependency on the right of the rule.

3. [20%] Let R be a relation schema and K is a subset of the attributes of R. Prove that if K functionally determines all attributes in R – K, then K is a superkey of R. Note that you are not allowed to cite Theorem 3 in Chapter 5 of the Lecture Notes for the proof. You need to prove it based on the definitions of functional dependency and superkey.