Solutions to Homework 3

3.4.1 [10 points]

Let
- A = street Address
- C = City
- D = area coDe
- N = Name
- P = Phone number (assumed not to include the area code)
- S = SS number
- T = sTate
- Z = Zip

S is assumed to be equivalent to the person; that is, every person is assumed to have exactly one and no two people have the same.

It is assumed that every person has a unique legal name.

The combination ACT determines the location uniquely, and will often appear grouped as a unit.

The combination DP determines a phone location uniquely (assuming we are not concerned with cell phones).

Basic FD's:

$ACT \rightarrow Z$	The exact location determines the Zip.
$ACT\ N \rightarrow S$	Assumes that there is at most one person with a given name living at a given location (e.g. the names of a father and son are distinguishable somehow).
$D \rightarrow T$	Area codes are not split between states.
$DP \rightarrow ACT$	An area code and phone number determine a unique location.
$S \rightarrow ACT\ N$	It is assumed that the SS number uniquely determines the person. Hence it also determines the person's name, and primary address.
$Z \rightarrow T$	The Zip determines the state, but not the city. (The Zip determines *a* city, the one in which the person's post-office is located, but not necessarily the one in which the person resides, and their could be several cities with the same zip.) It is assumed that the person's post-office is always within his/her state.

Keys: S DP, N DP

> Since the person alone does not determine the phone number, we need DP in every key. S determines everything else. N together with DP determines ACT and thus N with DP determines S.

3.5.2 **Note**: When we list an FD of form X → Y, we do not also list the obvious implied

FD's of the form: X' → Y where X ⊂ X'.

3.5.2 part i) [5 points] Given A → B, B → C, B → D.

Non-trivial FD's with singletons on the right-hand side, in addition to the above:

A → C, A → D

Keys: A

Superkeys that are not keys: AB, AC, AD, ABC, ACD, ABCD

3.5.2 part ii) [5 points] Given AB → C, BC → D, CD → A, and AD → B.

Non-trivial FD's with singletons on the right-hand side, in addition to the above:

AB → D, AB → C, BC → A, BC → D, CD → B, AD → B, AD → C

Keys: AB, AD, BC, CD (*not* AC, BD)

Superkeys that are not keys: ABC, ABD, ACD, BCD, ABCD

3.5.2 part iii) [5 points] Given A → B, B → C, C → D, and D → A.

Non-trivial FD's with singletons on the right-hand side, in addition to the above:

A → C, A → D, B → A, B → D, C → A, C → B, D → B, D → C

Keys: A, B, C, D

Superkeys that are not keys: AB, AC, AD, BC, BD, CD,
ABC, ABD, ACD, BCD, ABCD.

3.5.6 [5 points] Show that if $X \subseteq Y$, then $X^+ \subseteq Y^+$.

Consider the algorithm that determines X^+ from X.
All the elements of X are available in the computation of Y^+. Hence any attribute that is added in the computation of X^+ would also have been added in the computation of Y^+.

More specifically, if we define $X_0 = X$, and consider the determination of X_{i+1} from X_i in the i^{th} step of the algorithm, we see by induction that for each i, $X_i \subseteq Y_i$. It follows that when the algorithm stops with $Y_i = Y^+$, i will be such that the algorithm for X stopped no later than the i^{th} step, with $X_i = X^+$. Therefore $X^+ \subseteq Y^+$.

3.5.7 [5 points] Show that $(X^+)^+ = X^+$.

The algorithm for computing X^+ stops when no new attributes are introduced into the evolving set. However, that condition depends only on the state of the algorithm, not how we got to that state. Hence, if we start with X^+, we will get no new attributes, and immediately terminate. Therefore $(X^+)^+ = X^+$.

3.6.1e [10 points] R(A,B,C,D,E) with $AB \rightarrow C$, $B \rightarrow D$, $DE \rightarrow C$.

Derived FD's: $BE \rightarrow C$

Key: ABE

BCNF violations: $AB \rightarrow C$, $B \rightarrow D$, $BE \rightarrow C$, $DE \rightarrow C$

Decompose on $AB \rightarrow C$:

$(AB)^+ = ABCD$, other component is ABE.

ABE is in BCNF.

ABCD is not.

Decompose on $B \rightarrow D$:

B+ = BD, other component is ABC

Both are in BCNF.

The resulting decomposition is: ABC, ABE, BD.

Another possibility is: ABE, BD, CDE.

3.6.1f [not required, but still instructive]

R(A,B,C,D,E) with AB → C, C→ D, D→ B, D→ E.

Implied: AB → D, AB→ E, AD→ C, C→ E, D→ E.

Keys: AB, AD, AC

i) BCNF violations: C → B, C→ D, C → E, D→ B, D→ E

ii) 4NF decomposition:

Splitting first on C→ D, C^+ = BED, so we split to BED, ACE.

We still have a violation D→ B in BED, so split it to DB, DE, which are in BCNF.

There is a violation C → E in ACE, so split it to CE, AC.

The overall decomposition is AC, BD, CE, DE.

Note that this decomposition does not preserve AB → C however.

iii) 3NF violations: C → D, C → E

iv) 3NF decomposition:

The previous decomposition is 3NF, but not dependency preserving.

A dependency-preserving 3NF decomposition is:

ABC, CD, DE.