# Homework Assignment #4

**CS 589 Principles of Database Systems: Spring 2016**

This assignment is due Thursday, 2 June, at the beginning of class. You may do this assignment with a partner. If you do so, turn in a single submission with both names. You may seek help from your partner, the instructor and the class mailing list, but not other sources.

4A (10 points): Give a relational algebra expression that computes $I_P$ for the Datalog program P for `Local(In, Loc, Num)` given in class. You can restrict your attention to `Local` facts. That is, given a set of facts f, your expression should return all `Local` facts in $I_P(f)$.

Note: You may use expressions in project, for example $\pi_{A,B,E=A+2B}(r)$.

4B a. (10 points) Bancilhon and Ramakrishnan provide a syntactic condition for "strong safety" of Datalog programs with evaluable predicates that guarantees a program returns only a finite number of facts (equivalently, has a finite minimum model). However, there are programs with evaluable predicates that are finite even though they are not strongly safe. For example,

```
TotalPay(P, S)
    :- Salary(P, S1), Bonus(P,S2), Sum(S1, S2, S).
```

fails their definition, since the variable S doesn't appear in a base predicate. Give a more relaxed definition of strongly safe that would permit rules such as the one above. Explain how your condition guarantees finiteness.

b. (5 points) Show that the program for `Local(In, Loc, Num)` fails to satisfy your condition.

c. (10 points) Give a variant of the `Local` program that is strongly safe under your condition.

4C. Consider the following program $P_{Fill}$ that determines if tiles can fill an M-by-N area.

```
Fill(M, N) :- Tile(M, N).
Fill(M, N) :- Fill(N, M).
Fill(M, N) :- Fill(M, N1), Fill(M, N2),
  Sum(N1, N2, N), LessEq(M, 10), LessEq(N, 10).
```

a. (10 points) Suppose the extensional database is

```
Tile(2, 4).
Tile(1, 3).
```

Show that $P_{Fill}$ with this database implies `Fill(2, 7)` by giving a set of ground instances of rules that derive this fact.

b. (10 points) Explain why $P_{Fill}$ with the facts above cannot derive `Fill(5, 5)`.

c. (5 points) Show that there is a way to arrange tiles of the sizes above into a 5-by-5 area with no gaps.

4D. Refer to program $P_{Fill}$ above.

a. (10 points) If we remove

```
Fill(M, N) :- Fill(N, M).
```

and add

```
Fill(M, N) :- Tile(N, M).
```

do we get an equivalent program? Support your answer.

b. (15 points) If we remove

```
Fill(M, N) :- Fill(N, M).
```

and add

```
Fill(M, N) :- Tile(N, M).
Fill(M, N) :- Fill(M1, N), Fill(M2, N),
   Sum(M1, M2, M), LessEq(M, 10), LessEq(N, 10).
```

do we get an equivalent program? Support your answer.

4E There is an underground network of rooms and tunnels (described by predicates `Room` and `Tunnel`). Some rooms have `Gold` and some rooms have `Monsters`. You can `WinFrom` a room if you can go from that room to some gold without encountering monsters. If you can't, there is `NoWin` from that room. We can describe the situation with the following program $P_{Game}$.

```
WinFrom(R) :- Gold(R), ¬Monsters(R).
WinFrom(R) :- Tunnel(R, S), WinFrom(S), ¬Monsters(R).

NoWin(R) :- Room(R), ¬WinFrom(R).
```

a. (5 points) Give a stratification of the program $P_{Game}$. Include extensional predicates.

b. (15 points) Evaluate $P_{Game}$ with the following extensional database, using your stratification. Show the facts that you calculate at each stratum.

```
Room(treasury).      Tunnel(batcave, treasury).
Room(batcave).       Tunnel(lair, batcave).
Room(den).           Tunnel(den, treasury).
Room(lair).          Tunnel(armory, den).
Room(armory).        Tunnel(lair, armory).
Room(prison).        Tunnel(prison, armory).
Gold(treasury).
Monsters(den).
```