CS540: Assignment 1

Ga Young Lee
Department of Electrical Engineering and Computer Science
Oregon State University
leegay@oregonstate.edu

Jeffrey M. Young
Department of Electrical Engineering and Computer Science
Oregon State University
youngjef@oregonstate.edu

January 21, 2021

1 Problem 1

1.a

Problem Description:
Return names of every employee who works in the ”Hardware”, ”Software”, and ”Research” departments.

Datalog

```
f(Employee,DEmp) :- emp(Empid,Employee,A,B)
                   , works(Empid,Dept,C)
                   , dept(Dept,DEmp,D,E)
                   .
one(Employee) :- f(Employee, "Research")
                , f(Employee, "Hardware")
                , f(Employee, "Software")
                .
```

Relational Algebra

$$\pi_{ename}(\sigma_{dname="Research"}(\text{works} \bowtie_{works.did=dept.did} \text{dept} \bowtie_{dept.eid=emp.eid} \text{emp})$$
$$\cap$$
$$(\sigma_{dname="Software"}(\text{works} \bowtie_{works.did=dept.did} \text{dept} \bowtie_{dept.eid=emp.eid} \text{emp})$$
$$\cap$$
$$(\sigma_{dname="Hardware"}(\text{works} \bowtie_{works.did=dept.did} \text{dept} \bowtie_{dept.eid=emp.eid} \text{emp}))))$$

SQL

```sql
with tmp as (select ename, dname
from emp e
join works w on w.eid = e.eid
join dept  d on d.did = w.did)
select ename from tmp
where dname = "Hardware"
intersect
select ename from tmp
where dname = "Software"
intersect
select ename from tmp
where dname = "Research";
```

1.b

{Hi TA, this is a comment on the assignment. We thought this question was strange because every department has a manager, and by the definition given in the assignment every manager is an employee (and has an employee id). Thus, there is no relation on this instance which satisfies this problem description. We went ahead and wrote what we thought was the answer anyway.}

Problem Description:
Return the names of every department without any employee.

Datalog

```
R(x,y,w,z) :- dept(x,y,w,z)
            , works(s,x,t)
Q(y) :- dept(x,y,w,z), not R(x,y,w,z)
```

Relational Algebra

$\pi_{dname}(\texttt{dept}) - \pi_{dname}(\texttt{dept} \bowtie_{dept.did=works.did} \texttt{works})$

SQL

```sql
select dname
from dept d
where d.did not in
(select did from works)
;
```

1.c

Problem Description:
Print the managerids of managers who manage only departments with budgets greater than $1.5 million.

```
select managerid
from dept d
where managerid not in
(select managerid from dept where budget <= 1500000)
;
```

1.d

Problem Description:
Print the name of employees whose salary is less than or equal to the salary of every employee.

```
select ename
from emp
where salary <= ALL(select salary from emp)
;
```

1.e

Problem Description:
Print the enames of managers who manage the departments with the largest budget.

```
select ename
from emp e
where e.eid = (select managerid
from dept
where budget = (select MAX(budget) from dept));
```

1.f

Problem Description:
Print the name of every department and the average salary of the employees of that department. The department must have a budget more than or equal to $50

```
select D.dname, AVG(E.salary) AS 'average employee salary'
from dept D
join works W on W.did = D.did
join emp E on E.eid = W.eid
where D.budget >= 50
group by D.did
```

1.g

Problem Description:
Print the managerids of managers who control the largest amount of total budget. As an example, if a manager manages two departments, the amount of total budget for him/her will be the sum of the budgets of the two departments. We want to find managers that have max total budget

```sql
select D.managerid
from dept D
group by D.managerid
order by SUM(D.budget) DESC
limit 1
```

1.h

Problem Description:
Print the name of every employee whose salary is less than or equal to the average salary of all employees in his/her departments.

```sql
create view dept_avg AS(
  select did, AVG(salary) as AVG
  from emp E, works W
  where E.eid = W.eid
  group by W.did
);
select distinct ename
from emp E, works W, dept_avg A
where
  E.eid = W.eid and W.did = A.did and E.salary <= A.avg;
```

2 Problem 2

2.a

{Hello again, we wanted to point out that `coloneq`, $\coloneq$ is a good macro for type setting the $:-$ in datalog/prolog but it seems the assignment pdf doesn't use it. We also wanted to point out that the datalog for both 2a and 2b, as written, are only composed of free variables. For example, in 2a the `s` in the head is not bound in the body and thus for 2a homomorphism trivially holds. We're pretty sure this was a mistake so we answered the questions in the spirit of the assignment anyway. To be consistent we used the same notation (the `s` in the head of our queries) and specifically did *not* bind it to a body variable.}

Problem Description:
Explain whether the homomorphism theorem holds for conjunctive queries with equality, e.g., (T(s):-R(x,y), x=y). If your answer is no, then modify the theorem to hold for these queries.

To show that the homomorphism theorem holds for any two queries q_1 and q_2 for any database instance $\mathbf{I}$, we need to show query containment and that a morphism from q_2 to q_1 exists, or more succinctly:

1. $q_1(\mathbf{I})$ is contained in $q_2(\mathbf{I})$, i.e., $q_1(\mathbf{I}) \subseteq q_2(\mathbf{I})$
2. A structure preserving morphism exists that maps elements in q_2 to q_1, i.e., $\forall \mathbf{I} \,.\, \exists h \,.\, h : q_2 \to q_1$
3. Finally, q_2 is true on the canonical database of q_1

Let $q_1 = T(s) \coloneq R(x, y), x = y$, q_2 can be any query of the form
$q_2 = T(s) \coloneq S_0(w_i, z_i), w_0 = x_0 \ldots, S_i, w_i = x_i$, where $i \in \mathbb{N}$.

We can proceed by cases: either $q_1 = \emptyset$ on $\mathbf{I}$ or $q_1 \neq \emptyset$. In the first case, $q_1 \subseteq q_2$ since an empty set is a subset of any set. Therefore, the homomorphism theorem holds. For the second case, a homomorphism exists for all $\mathbf{I}$ because by assumption, q_1 is non-empty. This implies that over the database instance $\mathbf{I}$, the equality of $x = y$ exists. Because q_2 will be over the *same* instance, there will always exist a morphism h such that: $h : q_2 \to q_1$, $h(w_0) = x, h(z_0) = y, \ldots, h(w_i) = x, h(z_i) = y$, for all i, which is trivially an epimorphism (or surjective function) by $=$ and therefore a homomorphism in this domain.

2.b

Problem Description:
Explain whether the homomorphism theorem holds for conjunctive queries with $\leq$, e.g., (T(s):-R(x,y), x$\leq$,y).

Inequalities are problematic for conjunctive queries because they are not structure preserving morphisms. Consider a counterexample: the case query $q_1 = T(s) \coloneq R(x, y), x \leq y$ and $q_2 = T(s) \coloneq R(x, y), y \leq x$; notice that x and y has swapped places. For some database instances, specifically where $x = y$, we will have $q_1 \subseteq q_2$ because $x = y$ can form a homomorphism as demonstrated in 2.a. However, the homomorphism theorem states *for all* database instances, and thus for database instances where $x \neq y$, which will be necessarily included due to the inequality, we will *not* be able to find a homomorphism such that $h(x) = y$ by the assumption over the database instance that $x \neq y$.