EXERCISE 9

Andrea Crotti (#299466), Can Liu (#286105), Sebastian Roidl (#281941)

22 Dicembre 2009

Exercise 9.1[Tableau Containment] :

- D1 :{<a1,a2> | $\exists b1 \exists b2(R(a1, b1) \wedge R(b2, a2) \wedge R(b1, 5) \wedge R(5, b2))$}
  $D2 : \{< a1, a2 > | \exists b1 \exists b2 \exists b3 \exists b4(R(a1, b2) \wedge R(a1, b4) \wedge R(b1, a2) \wedge R(b2, b3) \wedge R(b4, b3) \wedge R(b3, b1))\}$

1.

T1

a1	a2
a1	b1
b2	a2
r1	5
5	b2

T2

a1	a2
a1	b2
a1	b4
b1	a2
b2	b3
b4	b3
b3	b1

2.

To check if one is included in the other we must check that:

- T1, T2 have the same columns and entries in result rows.

- the relation computed from T1 is a subset of the one from T2 for all valid assignments of relations to rows.

So we check the 2 possible cases and looking for a mapping. Where not specified otherwise h is the identity (maps a1 to a1 for example).

T1 $\subseteq T2$:

We must find a function h that maps all the rows from T1 to T2. h(resulting$_{\text{row}}$(T2)) = resulting$_{\text{row}}$(T1) I can start to define the following values for the function.

- h(b2) = b1
- h(b4) = b1
- h(b1) = b2
- h(b3) = a2
- h(b4) = b2
- h(b3) = a1

But then applying the function h I would get the tuples (b1, a2) and (a1, b2) which are not in T1.

a1	a2
a1	b1
a1	b1
b2	a2
b1	a2
b2	a2
a1	b2

So $T1 \nsubseteq T2$

T2 $\subseteq T1$:

We start by defining

- h(b1) = b2
- h(a2) = b3

But there are no ways to map the constant value in T2, so
T2 $\nsubseteq T1$.

Exercise 9.2[Tableau Minimization] :

$\pi_{A,B}(\sigma_{B=5}(R)) \bowtie \pi_{B,C}\ (\pi_{A,B}(R) \bowtie \pi_{A,C}(\sigma_{B=5}(R)))$

This query is redundant since as we can see the condition on B=5 is checked twice. We will end up doing a *Join* of two identical set of tuples, the second part of the query is useless.

1. Translate the RA into a tableau query

To write the corresponding tableau we need first to translate the RA query in DRC.
The DRC formula is:
$\{\langle A, B\rangle | \exists(a1, a2, a3, b1, b2, b3, b4, c1, c2) \wedge R(a1, b1) \wedge R(b2, c1) \wedge R(a2, b3) \wedge R(a3, c2) \wedge \{b1 = 5 \wedge b4 = 5\}\}$

And the tableau:

A	B
a1	b1
b2	c1
a2	b3
a3	c2
b1	5
b5	5

2. Perform tableau minimization over the result obtained from the previous task.

To minimize the tableau we iteratively delete one row and check if it's still equal to the original tableau.

After a few steps we get:

A	B
a1	5

3. Translate the minimized tableau back to RA.

$\pi_{A,B}(\sigma_{B=5}(R))$

Exercise 9.3[Join Ordering] :

Sailors(sid, sname, rating, age) Boats(bid, bname, color) Reserves(sid, bid, day, rname)

We assume that each tuple of Reserves is 40 bytes long, a page can hold 100 Reserves tuples, and we have 1000 pages of such tuples.
In addition, we know the following statistics:

- There are five different colors for boats.
- V(Reserves, sid) = 1500 and V(Reserves, bid) = 1000

We also know that sid is a key for Sailors, while bid is a key for Boats.

```
SELECT * FROM Boats B, Reserves R, Sailors S
WHERE R.sid = S.sid AND B.bid=R.bid AND B.color='red'
```

1. What is the selectivity of joining Reserves and Sailors?

2. What is the selectivity of joining Reserves and Boats?

3. Given is the code for searching for optimal bushy plans for join ordering, please modify it to produce only left-deep plans:

4. Search for optimal left deep plans using the dynamic programming algorithm obtained above.

For the cost estimation, use the sum of the sizes of all intermediate relations, not including the base relations themselves or the join result of the current full subset of relations under consideration for each subplan. For simplicity, size of an intermediate relation is assumed to be the cardinality of it, i.e., ignoring the differences in tuple length. We further assume selection is pushed down the plan before joining1.