

## EXERCISE 9

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

21 Dicembre 2009

### Exercise 9.1[Tableau Containment] :

- $D1 : \{ \langle a1, a2 \rangle \mid \exists b1 \exists b2 (R(a1, b1) \wedge R(b2, a2) \wedge R(b1, 5) \wedge R(5, b2)) \}$   
 $D2 : \{ \langle a1, a2 \rangle \mid \exists b1 \exists b2 \exists b3 \exists b4 (R(a1, b2) \wedge R(a1, b4) \wedge R(b1, a2) \wedge R(b2, b3) \wedge R(b4, b3) \wedge R(b3, b1)) \}$

1.

**T1**

	a1	a2
	a1	b1
	b2	a2
•	r1	5
	5	b2
	5	5

**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.

**T1**  $\subseteq$  **T2** :

**T1**  $\subseteq$  **T2** :

### Exercise 9.2[Tableau Minimization] :

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

1. Translate the RA into a tableau query
2. Perform tableau minimization over the result obtained from the previous task.
3. Translate the minimized tableau back to RA.

### 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. •  $V(\text{Reserves}, \text{sid}) = 1500$  and  $V(\text{Reserves}, \text{bid}) = 1000$

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 joining<sup>1</sup>.