

## Lesson 9: Acyclic join queries

**Theme:** Evaluation of acyclic join queries.

### 1 Conjunctive queries

A conjunctive query (CQ) is a query of the form:

$$\varphi(\bar{x}) := \exists \bar{y} R_1(\bar{z}_1) \wedge \cdots \wedge R_m(\bar{z}_m)$$

where  $\bar{x} = (x_1, \dots, x_k)$ ,  $\bar{y} = (y_1, \dots, y_l)$ , and each  $\bar{z}_i$  a vector of variables that come from  $\{x_1, \dots, x_k, y_1, \dots, y_l\}$ . A CQ is a Boolean CQ, if it does not have free variables  $\bar{x}$ .

Consider the following two problems.

- CQ-EVALUATION.

**Input:** A database DB and a Boolean CQ  $\varphi$ .

**Task:** Determine whether  $DB \models \varphi$ .

- CQ-EVALUATION<sub>DB</sub>.

**Input:** A Boolean CQ  $\varphi$ .

**Task:** Determine whether  $DB \models \varphi$ .

The difference between the two is that in CQ-EVALUATION<sub>DB</sub> the database DB is fixed, while in CQ-EVALUATION the database is part of the input.

The non-Boolean version of CQ-EVALUATION<sub>DB</sub> CQ-EVALUATION are when the input CQ  $\varphi$  is non-Boolean, i.e., it has free variables  $\bar{x}$ , and the task is to compute  $\varphi(DB)$ .

**Theorem 9.1**

- CQ-EVALUATION is NP-complete.
- There is a database DB such that CQ-EVALUATION<sub>DB</sub> is NP-complete.

### 2 Yannakakis' algorithm on acyclic queries

A CQ  $\varphi(\bar{x}) := \exists \bar{y} R_1(\bar{z}_1) \wedge \cdots \wedge R_m(\bar{z}_m)$  can be viewed as hypergraph  $\mathcal{H}_{\varphi(\bar{x})} = (V, \mathcal{E})$ , where:

- $V$  is the set of variables appearing in  $\varphi$ .
- $\mathcal{E} = \{\bar{z}_1, \dots, \bar{z}_m\}$ , where each  $\bar{z}_i$  is viewed as a “set.”

A CQ  $\varphi(\bar{x})$  is *acyclic*, if  $\mathcal{H}_{\varphi(\bar{x})}$  is acyclic.

**Theorem 9.2** [1] CQ-EVALUATION can be solved in polynomial time, when the input CQ is restricted to acyclic queries.

In fact, Theorem 9.2 also holds for non-Boolean version of CQ-EVALUATION.

### References

- [1] M. Yannakakis. Algorithms for acyclic database schemes. In *Proceedings of 7th International Conference on Very Large Data Base*, pages 82–94, 1981.