Lesson 2: Relational algebra

Theme: Relational algebra as a procedural language for querying database.

Syntax. Relational algebra (RA) expressions over vocabulary τ are defined inductively as follows:

- (Atomic expression) Every relation symbol $R \in \tau$ of arity n is an RA-expression of arity n.
- (Union, intersection and difference) If e_1 and e_2 are RA-expressions of arity n, then so are:
$$(e_1 \cup e_2), \qquad (e_1 \cap e_2) \qquad \text{and} \qquad (e_1 - e_2)$$
- (Selection) If e is an RA-expression of arity n and $1 \leqslant i, j \leqslant n$, then so is:
$$\sigma_{i=j}(e)$$
- (Projection) If e is an RA-expression of arity n and $1 \leqslant i_1, \ldots, i_m \leqslant n$, then the following is an RA-expression of arity m:
$$\pi_{i_1,\ldots,i_m}(e)$$
- (Semijoin) If e_1 and e_2 are RA-expressions of arity n_1 and n_2, respectively, and θ is of the form:
$$\bigwedge_{s=1}^{t} (i_s = j_s)$$
where $0 \leqslant t$, and $1 \leqslant i_1, \ldots, i_t \leqslant n_1$, and $1 \leqslant j_1, \ldots, j_t \leqslant n_2$, then the following is an RA-expression of arity n_1:
$$e_1 \ltimes_\theta e_2$$
- (Join) If e_1 and e_2 are RA-expressions of arity n_1 and n_2, respectively, and θ is of the form:
$$\bigwedge_{s=1}^{t} (i_s = j_s)$$
where $0 \leqslant t$, and $1 \leqslant i_1, \ldots, i_t \leqslant n_1$, and $1 \leqslant j_1, \ldots, j_t \leqslant n_2$, then the following is an RA-expression of arity $n_1 + n_2 - t$:
$$e_1 \bowtie_\theta e_2$$

Semijoin algebra (SA) expressions are RA expressions without any join operations.

Semantics. On a relational database DB, an relational algebra expression e defines a set of tuples $e(\mathsf{DB})$ as follows:

- If $e = R$,
$$e(\mathsf{DB}) \ := \ \{\bar{a} \mid R(\bar{a}) \in \mathsf{DB}\}$$
- If $e = (e_1 * e_2)$, where $* \in \{\cup, -, \cap\}$,
$$e(\mathsf{DB}) \ := \ e_1(\mathsf{DB}) * e_2(\mathsf{DB})$$

- If $e = \sigma_{i=j}(e')$,

$$e(\mathsf{DB}) \quad := \quad \{\bar{a} \in e'(\mathsf{DB}) \mid a_i = a_j\}$$

- If $e = \pi_{i_1,\ldots,i_m}(e')$,

$$e(\mathsf{DB}) \quad := \quad \{(a_{i_1}, \ldots, a_{i_m}) \mid \bar{a} \in e'(\mathsf{DB})\}$$

- If $e = (e_1 \ltimes_\theta e_2)$,

$$e(\mathsf{DB}) \quad := \quad \{\bar{a} \in e_1(\mathsf{DB}) \mid \text{ there is } \bar{b} \in e_2(\mathsf{DB}) \text{ such that } \theta(\bar{a}, \bar{b}) \text{ holds}\},$$

  where for $\theta = \bigwedge_{s=1}^{t}(i_s{=}j_s)$,

$$\theta(\bar{a}, \bar{b}) \text{ holds} \quad \text{if and only if} \quad a_{i_s}{=}b_{j_s} \text{ is true for all } 1 \leqslant s \leqslant t$$

- If $e = (e_1 \bowtie_\theta e_2)$,

$$e(\mathsf{DB}) \quad := \quad \{(\bar{a}, \pi_I(\bar{b})) \mid \bar{a} \in e_1(\mathsf{DB}) \ \wedge \ \bar{b} \in e_2(\mathsf{DB}) \text{ such that } \theta(\bar{a}, \bar{b}) \text{ holds}\}$$

  where:

  - $\pi_I(\bar{b})$ denote the projection of $\bar{b}$ to the indices in I and $I = \{1, \ldots, n_2\} - \{j \mid \text{there is } i \text{ s.t. } (i, j) \in \theta\}$
  - $\theta(\bar{a}, \bar{b})$ holds is as above.

Theorem 2.1 [1, Chapter 5][3] FO *and* RA *are equivalent for querying databases.*

- *For every* FO *formula* φ*, there is an* RA *expression* e *such that* $\varphi(DB) = e(DB)$ *for every database* DB*.*
- *For every* RA *expression* e*, there is an* FO *formula* φ *such that* $e(DB) = \varphi(DB)$ *for every database* DB*.*

Appendix

Relational algebra was introduced by Codd [2]. In database context, first-order logic is usually called *relational calculus*. For more details about relational model of databases, see the textbook by Abiteboul, Hull, and Vianu [1]. Note that it is available online.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995. URL: `http://webdam.inria.fr/Alice/`.

[2] E. F. Codd. A relational model of data for large shared data banks. *Communication of the ACM*, 13(6):377–387, 1970.

[3] E. F. Codd. Relational completeness of data base sublanguages. *In: R. Rustin (ed.): Database Systems: 65-98, Prentice Hall and IBM Research Report RJ 987, San Jose, California*, 1972.