Weekly exercises IN3020&4020 (2024)
Week 4: Set and Bag Relational Algebra

1 Joins and Value Invention (for sets)

Consider two relations:

$$T_1\,[\,A,\,B\,] : \{\langle 1, a\rangle, \langle 2, b\rangle, \langle 3, b\rangle\}$$
$$T_2\,[\,B,\,C\,] : \{\langle b, 2\rangle, \langle c, 4\rangle\}$$

1. Discuss the types of joins over set relations considered in the lectures: Cartesian product, natural join, theta-join, equi-join, semi-joins, anti-joins, outer joins. In particular,

   - give the brief descriptions of each operation
   - illustrate it with an example using relations T_1 and T_2 given above
   - briefly consider its expressibility in our core relational algebra
   - think about its usefulness for SQL

   Solution:

   - Cartesian product: $R \times S$ is the relation that is defined when R and S have no attributes in common and contains all tuples in R 'combined' with every tuple in S in all possible ways

     Example: $T_1 \times T_2$ is not defined because the relations have a common attribute B; however, $T_1 \times \rho_{B\to B'}(T_2)$ is defined and consists of all tuples with a pair from T_1 as the first two elements and a pair from T_2 as the second two elements (6 in total; e.g., $\langle 1, a, b, 2\rangle$)

     Expressivity: Cartesian product is a part of our core relational algebra, but it is not expressible via other 5 operations in this algebra

     Usefulness for SQL: by itself, it is rarely used, but can be written in SQL using ','; for example, `SELECT * FROM R, S` is the Cartesian product of R and S

- Natural join: $R \bowtie S$ generalises Cartesian Product to the case when R and S may have attributes in common; it contains all tuples in R joined with all possible tuple in S (a tuple in R joins with a tuple in S when they agree on common attributes)

  Example: $T_1 \bowtie T_2 : \{\langle 2, b, 2\rangle, \langle 3, b, 2\rangle\}$

  Expressivity: expressible in the core relational algebra, see the lectures

  Usefulness for SQL: one of the most common operations in SQL (note, however, that `SELECT * FROM R INNER JOIN S ON R.B = S.B` gives (slightly) different result, because it has two copies of B attribute in each tuple)

- Theta-join: $\bowtie_\theta$ for condition θ is join + selection with θ—that is, $R \bowtie_\theta S := \sigma_\theta(R \bowtie S)$

  Example: $T_1 \bowtie_{A>C} T_2 = \{\langle 3, b, 2\rangle\}$

  Expressivity: expressible in the core relational algebra essentially by definition

  Usefulness for SQL: cannot be written as a single operator in SQL, but is useful to combine selection and join (and then have a dedicated efficient algorithm for its evaluation)

- Equi-join: is $\bowtie_\theta$ with all conditions in θ being equality between attributes

  Example: $T_1 \bowtie_{A=C} T_2 = \{\langle 2, b, 2\rangle\}$

  Expressivity: expressible in the core relational algebra essentially by definition

  Usefulness for SQL: essentially, underlies `INNER JOIN` (but formally requires also renaming if the arguments have attributes in common)

- Left semi-join: $R \ltimes S$ is the relation that contains all tuples in R that join with S

  Example: $T_1 \ltimes T_2 = \{\langle 2, b\rangle, \langle 3, b\rangle\}$

  Expressivity: expressible in the core relational algebra in direct way by join and projection

  Usefulness for SQL: e.g., underlies `EXISTS` subqueries

- Right semi-join: $R \rtimes S$ is symmetric to left semi-join (i.e., the relation that contains all tuples in S that join with R)

  Example: $T_1 \rtimes T_2 = \{\langle b, 2\rangle\}$

Expressivity/Usefulness: same

- Left anti-join: $R \triangleright S$ is the relation that contains all tuples in R that do not join with S

  Example: $T_1 \triangleright T_2 = \{\langle 1, a\rangle\}$

  Expressivity: expressible in the core relational algebra in direct way by difference, join, and projection

  Usefulness for SQL: e.g., underlies `NOT EXISTS` subqueries

- Right anti-join: $R \triangleleft S$ is symmetric to left semi-join (i.e., the relation that contains all the tuples in S that do not join with R)

  Example: $T_1 \triangleleft T_2 = \{\langle c, 4\rangle\}$

  Expressivity/Usefulness: same

- Left outer join: $R \mathbin{⟕} S$ is the relation that contains all tuples in R joined with all possible tuples in S (i.e., all the tuples in $R \bowtie S$) and all dangling tuples in R extended with $NULL$ as the values of attributes of S that are not in R (a dangling tuple is a tuple in R that does not agree with any tuple in S on common attributes)

  Example: $T_1 \mathbin{⟕} T_2 : \{\langle 1, a, NULL\rangle, \langle 2, b, 2\rangle, \langle 3, b, 2\rangle\}$

  Expressivity: in not expressible in the core relational algebra, because it requires value invention

  Usefulness for SQL: underlies `LEFT OUTER JOIN` operation, which is commonly used

- Right outer join: $R \mathbin{⟖} S$ is symmetric to left outer join—that is, equal to $S \mathbin{⟕} R$

  Example: $T_1 \mathbin{⟖} T_2 : \{\langle 2, b, 2\rangle, \langle 3, b, 2\rangle, \langle NULL, c, 4\rangle\}$

  Expressivity/usefulness: same

- Full outer join: $R \mathbin{⟗} S$ is the union of the left and right outer join

  Example: $T_1 \mathbin{⟗} T_2 : \{\langle 1, a, NULL\rangle, \langle 2, b, 2\rangle, \langle 3, b, 2\rangle, \langle NULL, c, 4\rangle\}$

  Expressivity/usefulness: same

2. Evaluate expression $\pi_{A\times A\to Asq, B, NULL\to D, A-C\to E}(T_1 \bowtie T_2)$ over relations T_1 and T_2.

   Solution: $\{\langle 4, b, NULL, 0\rangle, \langle 9, b, NULL, 1\rangle\}$

2 Core operations for sets and bags

Consider the relations

$$R[A, B] : \{\langle 1, 2\rangle, \langle 3, 4\rangle, \langle 3, 5\rangle, \langle 4, 5\rangle\}$$
$$S[A, B] : \{\langle 1, 2\rangle, \langle 3, 5\rangle, \langle 3, 6\rangle, \langle 4, 5\rangle, \langle 1, 3\rangle\}$$

and the following bag relations (which are the same as R and S except that one tuple in each is duplicated):

$$Rb[A, B] : \{\{\langle 1, 2\rangle, \langle 3, 4\rangle, \langle 1, 2\rangle, \langle 3, 5\rangle, \langle 4, 5\rangle\}\}$$
$$Sb[A, B] : \{\{\langle 1, 2\rangle, \langle 3, 5\rangle, \langle 4, 5\rangle, \langle 3, 6\rangle, \langle 4, 5\rangle, \langle 1, 3\rangle\}\}$$

Compute the following expressions:

1. $\sigma_{A=3}(Rb \cup^b Sb)$

   Solution:
   $\{\{\langle 3, 4\rangle, \langle 3, 5\rangle, \langle 3, 5\rangle, \langle 3, 6\rangle\}\}$

2. $\sigma_{A=3}(R \cup S)$

   Solution: $\{\langle 3, 4\rangle, \langle 3, 5\rangle, \langle 3, 6\rangle\}$

3. $\sigma_{A=3}(R \cup^b S)$

   Solution: $\{\{\langle 3, 4\rangle, \langle 3, 5\rangle, \langle 3, 5\rangle, \langle 3, 6\rangle\}\}$

4. $Sb \setminus Rb$

   Solution: $\{\{\langle 3, 6\rangle, \langle 1, 3\rangle, \langle 4, 5\rangle\}\}$

5. $S \setminus R$

   Solution: $\{\langle 3, 6\rangle, \langle 1, 3\rangle\}$

6. $\pi^b_A(Rb)$

   Solution: $\{\{\langle 1\rangle, \langle 3\rangle, \langle 1\rangle, \langle 3\rangle, \langle 4\rangle\}\}$

7. $\pi_A(Rb)$

   Solution: This is undefined because R is not a set.

8. $\pi^b_A(R)$

   Solution: $\{\{\langle 1\rangle, \langle 3\rangle, \langle 3\rangle, \langle 4\rangle\}\}$

9. $\pi_A(R)$

   Solution: $\{\langle 1\rangle, \langle 3\rangle, \langle 4\rangle\}$

3 Bag-specific operations

Consider the following bag relation:

T:

A	B	C
1	a	*NULL*
1	*NULL*	*NULL*
2	a	*NULL*
NULL	a	*NULL*
NULL	a	*NULL*

Evaluate the following expressions over T:

1. $\delta(T)$

   Solution:

   $\delta(T)$:

A	B	C
1	a	*NULL*
1	*NULL*	*NULL*
2	a	*NULL*
NULL	a	*NULL*

2. $\delta(\pi^b_A(T))$

   Solution: (same as $\pi_A(T)$))

   $\delta(\pi^b_A(T))$:

A
1
2
NULL

3. $\gamma_{\mathsf{COUNT}(B),\mathsf{COUNT_DISTINCT}(B),\mathsf{COUNT}(C),\mathsf{COUNT_DISTINCT}(C)}(T)$

   Solution: $\{\{\langle 4, 1, 0, 0\rangle\}\}$

4. $\gamma_{B,\mathsf{MIN}(A),\mathsf{SUM}(A),\mathsf{AVG}(A),\mathsf{SUM}(C)}(T)$

   Solution: $\{\{\langle a, 1, 3, 1.5, \mathit{NULL}\rangle, \langle \mathit{NULL}, 1, 1, 1, \mathit{NULL}\rangle\}\}$

4 Expression of operations and algebraic laws

1. Express the following bag relational algebra expression using theta-join: $\pi^b_A(\sigma_{A>B \wedge B>4}(R \bowtie S))$

   Solution: $\pi^b_A(R \bowtie_{A>B \wedge B>4} S)$

2. Express δ using other bag relational algebra operations.

   Solution: $\delta(R) := \gamma_{A_1,A_2,\ldots,A_n}(R)$

3. (*) Full outer join of two relations R and S over bags is defined as the bag of tuples that consists of all tuples in the natural join of R and S, all dangling tuples of R extended with *NULLs* (with the same multiplicities as in R), and all dangling tuples of S extended with *NULLs* (with the same multiplicities as in S). Give examples of R and S showing that both $R \rtimes\ltimes S := (R \rtimes S) \cup^b (R \rtimes S)$ and $R \rtimes\ltimes S := (R \rtimes S) \cup (R \rtimes S)$ are incorrect over bags, and give a correct definition of full outer join via other bag relational algebra operations.

   Solution: To falsify

   $$R \rtimes\ltimes S = (R \rtimes S) \cup^b (R \rtimes S),$$

   we can take $R = S = \{\{\langle 1\rangle\}\}$: we have

   $$R \rtimes\ltimes S = \{\{\langle 1\rangle\}\}, \text{ but } (R \rtimes S) \cup^b (R \rtimes S) = \{\{\langle 1\rangle, \langle 1\rangle\}\}.$$

   To falsify

   $$R \rtimes\ltimes S = (R \rtimes S) \cup (R \rtimes S),$$

   we can take $R = S = \{\{\langle 1\rangle, \langle 1\rangle\}\}$: we have

   $$R \rtimes\ltimes S = \{\{\langle 1\rangle, \langle 1\rangle\}\}, \text{ but } (R \rtimes S) \cup (R \rtimes S)$$

   is undefined. On of the correct ways to express full outer join is

   $$R \rtimes\ltimes S := ((R \rtimes S) \cup^b (R \rtimes S)) \setminus (R \ltimes S).$$

5 Constraints

Consider relations with the following schemas (the domains and the meaning of the attributes are as expected):

$$\mathit{Product}\,[\mathit{manufacturer}, \mathit{model}, \mathit{type}]$$
$$\mathit{PC}\,[\mathit{model}, \mathit{speed}, \mathit{RAM}, \mathit{HDD}, \mathit{price}]$$

1. Express using relational algebra that *model* in *PC* is a foreign key referencing *model* in *Product* (you do not need to explicitly express that the latter is a primary key).

   Solution: $\delta(\pi^b_{model}(PC)) \subseteq \delta(\pi^b_{model}(Product))$ Note that the solution works even for bag relations; when *PC* and *Product* are just sets, π may be used instead of π^b and δ is not needed.

2. (*) Express using relational algebra the constraint that *model* in *PC* cannot be *NULL*.

   Solution: $\sigma_{model=NULL}(PC) = \emptyset$.