Data Integration: Provenance

Jan Chomicki

University at Buffalo

Provenance

Annotations recording how a tuple in the query result was produced from the database.

Different kinds

- Why-provenance (lineage): return a relevant part of the database
- How-provenance: keeping track of individual derivations
- Where-provenance: keeping track of individual attribute values

Why-provenance

Query language

Relational algebra:

- without set difference (positive RA)
- without renaming (for simplicity)
- with constant singleton relations: $\{u\}$

Notation

- Q: query
- t: tuple
- $R_1, \ldots, R_k$: relation names
- D: database instance consisting of relation instances $r_1, \ldots, r_k$
- $\mathbf{S} \Cup \mathbf{T} = \{S \cup T \mid S \in \mathbf{S} \wedge T \in \mathbf{T}\}$

Why-provenance: definition

Tuple annotations: sets of sets of facts.

Definition

$$
\begin{aligned}
Why(\{u\}, D, t) &= \begin{cases} \{\varnothing\} & \text{if } t = u \\ \varnothing & \text{otherwise} \end{cases} \\
Why(R_i, D, t) &= \begin{cases} \{\{R_i(t)\}\} & \text{if } t \in r_i \\ \varnothing & \text{otherwise} \end{cases} \\
Why(\sigma_C(Q), D, t) &= \begin{cases} Why(Q, D, t) & \text{if } t \text{ satisfies } C \\ \varnothing & \text{otherwise} \end{cases} \\
Why(\pi_X(Q), D, t) &= \bigcup\{Why(Q, D, u) \mid u \in Q(D) \wedge t = u[X]\} \\
Why(Q_1 \cup Q_2, D, t) &= Why(Q_1, D, t) \cup Why(Q_2, D, t) \\
Why(Q_1 \bowtie Q_2, D, t) &= Why(Q_1, D, t[U_1]) \Cup Why(Q_2, D, t[U_2])
\end{aligned}
$$

Why-provenance: properties

Empty provenance

If $\mathit{Why}(Q, D, t) = \varnothing$, then $t \notin Q(D)$.

Nonempty provenance

If $J \in \mathit{Why}(Q, D, t)$, then $J \subseteq D$ and $t \in Q(J)$.

K-semirings

Tuple annotations: values from a special domain $\mathcal{K}$.

The properties of $\mathcal{K}$

$\mathcal{K} = (K, 0, 1, +, \cdot)$ is a commutative semiring:

- addition $(+)$ is associative, commutative and has identity 0
- multiplication $(\cdot)$ is associative, commutative and has identity 1
- for all x: $x \cdot 0 = 0 \cdot x = 0$
- multiplication distributes over addition.

Examples of $\mathcal{K}$

- Booleans: relations as sets
- natural numbers: relations as bags
- polynomials: how-provenance

Tuple annotations

- tuples not in the database: 0
- tuples in the database: tuple identifiers
- tuples in a query result: polynomial expressions encoding tuple derivations

Definition

$$
\begin{aligned}
How(\{u\}, D, t) &= \begin{cases} 1 & \text{if } t = u \\ 0 & \text{otherwise} \end{cases} \\
How(R_i, D, t) &= \begin{cases} V & \text{if } t \in r_i \text{ with annotation } V \\ 0 & \text{otherwise} \end{cases} \\
How(\sigma_C(Q), D, t) &= \begin{cases} How(Q, D, t) & \text{if } t \text{ satisfies } C \\ 0 & \text{otherwise} \end{cases} \\
How(\pi_X(Q), D, t) &= \sum\{How(Q, D, u) \mid How(Q, D, u) \neq 0 \land t = u[X]\}
\end{aligned}
$$

Union

$$How(Q_1 \cup Q_2, D, t) = How(Q_1, D, t) + How(Q_2, D, t)$$

Join

$$How(Q_1 \bowtie Q_2, D, t) = How(Q_1, D, t[U_1]) \cdot How(Q_2, D, t[U_2])$$

Recovering Why-provenance

$$\mathcal{K} = (\wp(\wp(Facts)), \varnothing, \{\varnothing\}, \cup, \Cup)$$

where $\wp$ is the powerset operator and *Facts* is the set of all facts.