CS 594 - HW 2

Huy Truong
Computer Science Department
The University of Illinois at Chicago
thuyng2@uic.edu

October 9, 2024

1 Part 1.1 Foundations of Provenance Models

1.1 Question 1.1.1 Explain the declarative notions of sufficiency and minimality principles that are the foundation of many provenance models.

Let's say we run query Q on table R and get tuple t as the output. Sufficiency describes a situation where a set of elements in R, say s, is enough to produce some elements of interest in t, say e, even if there could be redundant or irrelevant information in s. Minimality pushes sufficiency to its limits where there must also be zero redundancy or irrelevance in s. Therefore, minimality implies sufficiency but not the other way around.

1.2 Question 1.1.2 List the provenance models you have seen so far.

Why-Provenance

1.3 Question 1.1.3 What are the advantages and disadvantages of the declarative and syntactic definitions of provenance?

Declarative provenance identifies the input data responsible for a result (for example, the Why-Provenance). Therefore, it provides quick insights into the causal relationships in data transformations, making it easier for humans to understand. It is often independent of the specific syntax of a querying language and only depends on how those queries behave, making it more transferable across languages. However, determining minimal and sufficient provenance is computationally prohibitive, especially for complex queries. Furthermore, they don't provide traceable steps involved in the data transformations.

Syntactic provenance identifies the detailed steps performed on the data, typically returning a data lineage (for example, the Where-Provenance). Therefore, it provides an easy way for humans to retrace the exact steps in data transformations.

It can reproduce and verify the results of complex data transformations. However, it is verbose, making it harder to extract human-readable insights. It is also tightly coupled to the specific implementation of the querying languages, hindering transferability.

1.4 Question 1.1.4 What is insensitivity to query rewrite and how does this property relate to query equivalence?

Two queries are equivalent if they return the same result over every database. Therefore, for a query to be insensitive to a rewrite, that rewrite must transform that query into a syntactically different but semantically equivalent form.

1.5 Question 1.1.5 What is the definition of a witness used in Why-provenance?

A witness w in Why-provenance must be an element of $Why(Q, D, t)$ where Q is a query in relational algebra, D is a database, and $t \in Q(D)$:

$$Why(R, D, t) = \begin{cases} \{\{t\}\} \text{ if } t \in R \\ \varnothing \text{ otherwise} \end{cases}$$

$$Why(\rho_{AB}(Q), D, t) = Why(Q, D, t.AB)$$

$$Why(\sigma_\theta(Q), D, t) = \begin{cases} Why(Q, D, t) \text{ if } t \vDash \theta \\ \varnothing \text{ otherwise} \end{cases}$$

$$Why(\pi_A(Q), D, t) = \bigcup_{u \in Q(D) : u.A = t} Why(Q, D, u)$$

$$Why(Q_1 \times Q_2, D, t) = \{D' \cup D'' | D' \in Why(Q, D, t.Q_1) \wedge D'' \in Why(Q, D, t.Q_2)\}$$

$$Why(Q_1 \cup Q_2, D, t) = Why(Q_1, D, t) \cup Why(Q_2, D, t)$$

$$Why(Q_1 - Q_2, D, t) = Why(Q_1, D, t) - Why(Q_2, D, t)$$

1.6 Question 1.1.6 What properties have to hold for (K, +, ×, 0, 1) to be a semiring?

Let $a, b, c, 0, 1 \in K$, we need the following properties to hold: Commutativity of addition $a + b = b + a$, commutativity of multiplication $a \times b = b \times a$, associativity of addition $(a + b) + c = a + (b + c)$, associativity of multiplication $(a \times b) \times c = a \times (b \times c)$, neutral element of addition $a + 0 = a$, neutral element of multiplication $a \times 1 = a$, annihilation by zero $a \times 0 = 0$, and multiplication distributes over addition $a \times (b + c) = a \times b + a \times c$.

1.7 Question 1.1.7 Give definitions for the operators of the relational algebra over semiring-annotated relations.

Let a semiring over a set K be $(K, +_K, \times_K, 0_K, 1_K)$ where $\times_K : K \times K \to K$, $+_K : K \times K \to K$, and $0_K, 1_K \in K$. The relational algebra contains the following operators that can be defined over this semiring:

$$\rho_{A \leftarrow B}(R)(t) = R(t.[B \leftarrow A])$$
$$\pi_A(R)(t) = \sum_{t=t'.A} R(t')$$
$$\sigma_\theta(R)(t) = R(t) \times_K \theta(t) \text{ where } \theta(t) = 1_K \text{ if } t \vDash \theta \text{ , otherwise } 0_K$$
$$(R_1 \times R_2)(t) = R_1(t.R_1) \times_K R_2(t.R_2)$$
$$(R_1 \cup R_2)(t) = R_1(t) +_K R_2(t)$$

1.8 Question 1.1.8 Which are important semirings in the semiring annotated relational model and what is their correspondence in the standard relational model?

Boolean Semiring $(\{T, F\}, \vee, \wedge, F, T)$: The set of boolean values with the or and the and operators. This corresponds to the presence/absence tracking in the standard relational model

Natural Numbers $(\mathbb{N}, +, \times, 0, 1)$: The set of natural numbers with the addition and multiplication operators. This corresponds to the bag semantics in the standard relational model

Possible Worlds Semiring $(\mathcal{P}(X), \cup, \cap, \varnothing, X)$: The powerset of X with the union and intersection operators. This corresponds to the data lineage tracking (provenance semiring) in the standard relational model

1.9 Question 1.1.9 Define the semiring that corresponds to why-provenance and the semiring that corresponds to minimal why-provenance.

Why-provenance Semiring $(\mathcal{P}(\mathcal{P}(X)), \cup, \uplus, \varnothing, \{\varnothing\})$: The powerset of the power-set of X (some set of tuples) with the union operator and the concatenation operator $k1 \uplus k2 = \{w1 \cup w2 | w1 \in k1 \wedge w2 \in k2\}$.

Given the definition of the minimal why-provenance $MWhy(Q, D, t) = \{w | w \in Why(Q, D, t) \wedge \nexists w' \subset w : w' \in Why(Q, D, t)\}$, the Minimal Why-provenance Semiring is $(\mathcal{P}(\mathcal{P}(X)), \cup, \uplus, \varnothing, \{\varnothing\})$. The only difference from the why-provenance semiring is the minimal subset union operator $k1 \cup k2 = \{s | s \in k1 \cup k2 \wedge \nexists w \in k1 \cup k2 : (w \subset s)\}$.

2 Part 1.2 Provenance Computation

For conciseness, we denote the table $lawsuit$ as l, $defendant$ as d, and $plaintiff$ as p. Also, let $Why(Q, D) = \{(t, w) | t \in Q(D) \wedge w \in Why(Q, D, t)\}$, similarly for $MWhy(Q, D)$.

2.1 $\pi_{name}(d) \cup \pi_{name}(p)$

To get why-provenance, we consider:

$$\begin{aligned}
Why(\pi_{name}(d), D) = \{&((Peter), \{\{d1\}, \{d2\}, \{d5\}\}),\\
&((Bob), \{\{d3\}\}), ((Alice), \{\{d4\}\})\}\\
Why(\pi_{name}(p), D) = \{&((Joe), \{\{p1\}, \{p2\}\}), ((Jim), \{\{p3\}\}), ((Peter), \{\{p4\}\}),\\
&((Gerd), \{\{p5\}\}), ((Pferd), \{\{p6\}\})\}
\end{aligned}$$

Therefore,

$$\begin{aligned}
Why(\pi_{name}(d) \cup \pi_{name}(p), D) &= Why(\pi_{name}(d), D) \cup Why(\pi_{name}(p), D)\\
&= \{((Peter), \{\{d1\}, \{d2\}, \{d5\}, \{p4\}\}),\\
&((Bob), \{\{d3\}\}), ((Alice), \{\{d4\}\}),\\
&((Joe), \{\{p1\}, \{p2\}\}), ((Jim), \{\{p3\}\}),\\
&((Gerd), \{\{p5\}\}), ((Pferd), \{\{p6\}\})\}
\end{aligned}$$

To get the minimal why-provenance, we note that the why-provenance for this query is already minimal. Therefore,

$$\begin{aligned}
MWhy(\pi_{name}(d) \cup \pi_{name}(p), D) = \{&((Peter), \{\{d1\}, \{d2\}, \{d5\}, \{p4\}\}),\\
&((Bob), \{\{d3\}\}), ((Alice), \{\{d4\}\}),\\
&((Joe), \{\{p1\}, \{p2\}\}), ((Jim), \{\{p3\}\}),\\
&((Gerd), \{\{p5\}\}), ((Pferd), \{\{p6\}\})\}
\end{aligned}$$

The Provenance Polynomials for every tuple in the query result are as follows:

$$\begin{aligned}
&(Peter) : d1 + d2 + d5 + p4\\
&(Bob) : d3\\
&(Alice) : d4\\
&(Joe) : p1 + p2\\
&(Jim) : p3\\
&(Gerd) : p5\\
&(Pferd) : p6
\end{aligned}$$

2.2 $\pi_{location,dname,pname}((\rho_{id,dname,homestate}(d) \times l \times \rho_{id,pname}(p))$

For conciseness, we avoid writing down the why-provenance of the renaming operation in intermediate results and implicitly apply it to the natural join operation:

$$
\begin{aligned}
Why(d \times l, D) = \{&((1, Peter, IL, 01/01, IL, Harassment), \{\{d1, l1\}\}),\\
&((2, Peter, IL, 02/03, TX, Breachofcontract), \{\{d2, l2\}\}),\\
&((3, Bob, TX, 03/06, WA, Taxfraud), \{\{d3, l3\}\}),\\
&((4, Alice, CA, 02/03, IL, Taxfraud), \{\{d4, l4\}\}),\\
&((4, Peter, IL, 02/03, IL, Taxfraud), \{\{d5, l4\}\})\}\\
Why((d \times l) \times p, D) = \{&((1, Peter, IL, 01/01, IL, Harassment, Joe), \{\{d1, l1, p1\}\}),\\
&((2, Peter, IL, 02/03, TX, Breachofcontract, Joe), \{\{d2, l2, p2\}\}),\\
&((2, Peter, IL, 02/03, TX, Breachofcontract, Jim), \{\{d2, l2, p3\}\}),\\
&((3, Bob, TX, 03/06, WA, Taxfraud, Peter), \{\{d3, l3, p4\}\}),\\
&((4, Alice, CA, 02/03, IL, Taxfraud, Gerd), \{\{d4, l4, p5\}\}),\\
&((4, Alice, CA, 02/03, IL, Taxfraud, Pferd), \{\{d4, l4, p6\}\}),\\
&((4, Peter, IL, 02/03, IL, Taxfraud, Gerd), \{\{d5, l4, p5\}\}),\\
&((4, Peter, IL, 02/03, IL, Taxfraud, Pferd), \{\{d5, l4, p6\}\})\}
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&Why(\pi_{location,dname,pname}((\rho_{id,dname,homestate}(d) \times l \times \rho_{id,pname}(p)), D) =\\
&\{((IL, Peter, Joe), \{\{d1, l1, p1\}\}),\\
&((TX, Peter, Joe), \{\{d2, l2, p2\}\}),\\
&((TX, Peter, Jim), \{\{d2, l2, p3\}\}),\\
&((WA, Bob, Peter), \{\{d3, l3, p4\}\}),\\
&((IL, Alice, Gerd), \{\{d4, l4, p5\}\}),\\
&((IL, Alice, Pferd), \{\{d4, l4, p6\}\}),\\
&((IL, Peter, Gerd), \{\{d5, l4, p5\}\}),\\
&((IL, Peter, Pferd), \{\{d5, l4, p6\}\})\}
\end{aligned}
$$

The why-provenance for this query is already minimal. Therefore, $MWhy(Q, D) = Why(Q, D)$ for this query.

The Provenance Polynomials for every tuple in the query result are as follows:

$$(IL, Peter, Joe) : d1 * l1 * p1$$
$$(TX, Peter, Joe) : d2 * l2 * p2$$
$$(TX, Peter, Jim) : d2 * l2 * p3$$
$$(WA, Bob, Peter) : d3 * l3 * p4$$
$$(IL, Alice, Gerd) : d4 * l4 * p5$$
$$(IL, Alice, Pferd) : d4 * l4 * p6$$
$$(IL, Peter, Gerd) : d5 * l4 * p5$$
$$(IL, Peter, Pferd) : d5 * l4 * p6$$

2.3 $\pi_{location}(\sigma_{location=homestate}(l \times \rho_{id,name,homestate}(d))) \cup \pi_{location}(l)$

For conciseness, we avoid writing down the why-provenance of the renaming operation in intermediate results and implicitly apply it to the natural join operation:

$$\begin{aligned}
Why(l \times d, D) = \{&((1, 01/01, IL, Harassment, Peter, IL), \{l1, d1\}),\\
&((2, 02/03, TX, Breachofcontract, Peter, IL), \{l2, d2\}),\\
&((3, 03/06, WA, Taxfraud, Bob, TX), \{l3, d3\}),\\
&((4, 02/03, IL, Taxfraud, Alice, CA), \{l4, d4\}),\\
&((4, 02/03, IL, Taxfraud, Peter, IL), \{l4, d5\})\}\\
Why(\sigma_{location=homestate}(l \times d), D) = \{&((1, 01/01, IL, Harassment, Peter, IL), \{l1, d1\}),\\
&((4, 02/03, IL, Taxfraud, Peter, IL), \{l4, d5\})\}\\
Why(\pi_{location}(\sigma), D) = \{&((IL), \{\{l1, d1\}, \{l4, d5\}\})\}\\
Why(\pi_{location}(l), D) = \{&((IL), \{\{l1\}, \{l4\}\}), ((TX), \{\{l2\}\}), ((WA), \{\{l3\}\})\}\\
Why(\pi(\sigma) \cup \pi, D) = \{&((IL), \{\{l1, d1\}, \{l4, d5\}, \{l1\}, \{l4\}\}),\\
&((TX), \{\{l2\}\}), ((WA), \{\{l3\}\})\}
\end{aligned}$$

To get minimal why-provenance for each tuple, we exclude witnesses with a proper subset(s) being in the why-provenance of that tuple:

$MWhy(\pi(\sigma) \cup \pi, D) = \{((IL), \{\{l1\}, \{l4\}\}), ((TX), \{\{l2\}\}), ((WA), \{\{l3\}\})\}$

The Provenance Polynomials for every tuple in the query result are as follows:

$$(IL) : l1 * d1 + l4 * d5 + l1 + l4$$
$$(TX) : l2$$
$$(WA) : l3$$