ProvSQL: A General System for Keeping Track of the Provenance and Probability of Data

Aryak Sen Silviu Maniu Pierre Senellart

ICDE, May 2026

Motivation

Context: Data is increasingly being used for decision making across a wide range of domains.

Motivation

Context: Data is increasingly being used for decision making across a wide range of domains.

- Sources matter

Motivation

Context: Data is increasingly being used for decision making across a wide range of domains.

- Sources matter
- Data is uncertain

Motivation

Context: Data is increasingly being used for decision making across a wide range of domains.

- Sources matter
- Data is uncertain

Practical solution: There's a need for systems that can jointly track data provenance and reason about uncertainty.

Well what exactly is data provenance when it comes to relational databases?

Example

Table: *Personnel*

id	**name**	**position**	**city**
1	Juma	Director	Nairobi
2	Paul	Janitor	Nairobi
3	David	Analyst	Paris
4	Ellen	Field agent	Beijing
5	Aaheli	Double agent	Paris
6	Nancy	HR	Paris
7	Jing	Analyst	Beijing

Example

Table: *Personnel*

id	**name**	**position**	**city**
1	Juma	Director	Nairobi
2	Paul	Janitor	Nairobi
3	David	Analyst	Paris
4	Ellen	Field agent	Beijing
5	Aaheli	Double agent	Paris
6	Nancy	HR	Paris
7	Jing	Analyst	Beijing

Query:

"What are the cities where at least two persons are working?"

Example

Result:

Nairobi
Paris
Beijing

Example

Q: What if we add annotations to *Personnel*?

Example

Q: What if we add annotations to *Personnel*?

Table: *Personnel* with an added column for annotations

id	**name**	**position**	**city**	**annotation**
1	Juma	Director	Nairobi	t_1
2	Paul	Janitor	Nairobi	t_2
3	David	Analyst	Paris	t_3
4	Ellen	Field agent	Beijing	t_4
5	Aaheli	Double agent	Paris	t_5
6	Nancy	HR	Paris	t_6
7	Jing	Analyst	Beijing	t_7

Example

Semiring provenance : And now if the tuple annotations t_i's are interpreted as elements of a *semiring* $(\mathbb{K}, \oplus, \otimes, \mathbb{0}, \mathbb{1})$

Example

Semiring provenance : And now if the tuple annotations t_i's are interpreted as elements of a *semiring* $(\mathbb{K}, \oplus, \otimes, \mathbb{0}, \mathbb{1})$

Result of our query on the annotated *Personnel* relation:

Nairobi	$t_1 \otimes t_2$
Paris	$(t_3 \otimes t_5) \oplus (t_5 \otimes t_6) \oplus (t_3 \otimes t_6)$
Beijing	$t_4 \otimes t_7$

Example

Semiring provenance : And now if the tuple annotations t_i's are interpreted as elements of a *semiring* $(\mathbb{K}, \oplus, \otimes, \mathbb{0}, \mathbb{1})$

Result of our query on the annotated *Personnel* relation:

Nairobi	$t_1 \otimes t_2$
Paris	$(t_3 \otimes t_5) \oplus (t_5 \otimes t_6) \oplus (t_3 \otimes t_6)$
Beijing	$t_4 \otimes t_7$

The general idea:

alternative derivations are interpreted as $\oplus$

joint derivations are interpreted as $\otimes$

Example

Note: If instead of $\mathbb{K}$ we choose the semiring of Boolean functions $(\mathcal{B}[X], \vee, \wedge, \bot, \top)$ where $X = \{t_1, \ldots, t_7\}$

Example

Note: If instead of $\mathbb{K}$ we choose the semiring of Boolean functions $(\mathcal{B}[X], \vee, \wedge, \bot, \top)$ where $X = \{t_1, \ldots, t_7\}$

Result:

Nairobi	$t_1 \wedge t_2$
Paris	$(t_3 \wedge t_5) \vee (t_5 \wedge t_6) \vee (t_3 \wedge t_6)$
Beijing	$t_4 \wedge t_7$

Example

Note: If instead of $\mathbb{K}$ we choose the semiring of Boolean functions $(\mathcal{B}[X], \vee, \wedge, \bot, \top)$ where $X = \{t_1, \ldots, t_7\}$

Result:

Nairobi	$t_1 \wedge t_2$
Paris	$(t_3 \wedge t_5) \vee (t_5 \wedge t_6) \vee (t_3 \wedge t_6)$
Beijing	$t_4 \wedge t_7$

Explanation: Paris is in the result iff either both tuples representing David and Aaheli, or Aaheli and Nancy, or David and Nancy, are present.

Example

Note: If instead of $\mathbb{K}$ we choose the semiring of Boolean functions $(\mathcal{B}[X], \vee, \wedge, \bot, \top)$ where $X = \{t_1, \ldots, t_7\}$

Result:

Nairobi	$t_1 \wedge t_2$
Paris	$(t_3 \wedge t_5) \vee (t_5 \wedge t_6) \vee (t_3 \wedge t_6)$
Beijing	$t_4 \wedge t_7$

Explanation: Paris is in the result iff either both tuples representing David and Aaheli, or Aaheli and Nancy, or David and Nancy, are present.

Further: This represents Boolean provenance and can be used to compute probabilities.

ProvSQL

- existing tools fall short

ProvSQL

- existing tools fall short
- ProvSQL, a PostgreSQL extension, fills this gap with a generic, scalabe, and easy-to-deploy solution

ProvSQL

- existing tools fall short
- ProvSQL, a PostgreSQL extension, fills this gap with a generic, scalabe, and easy-to-deploy solution
- does provenance tracking and probability computation over relational databases

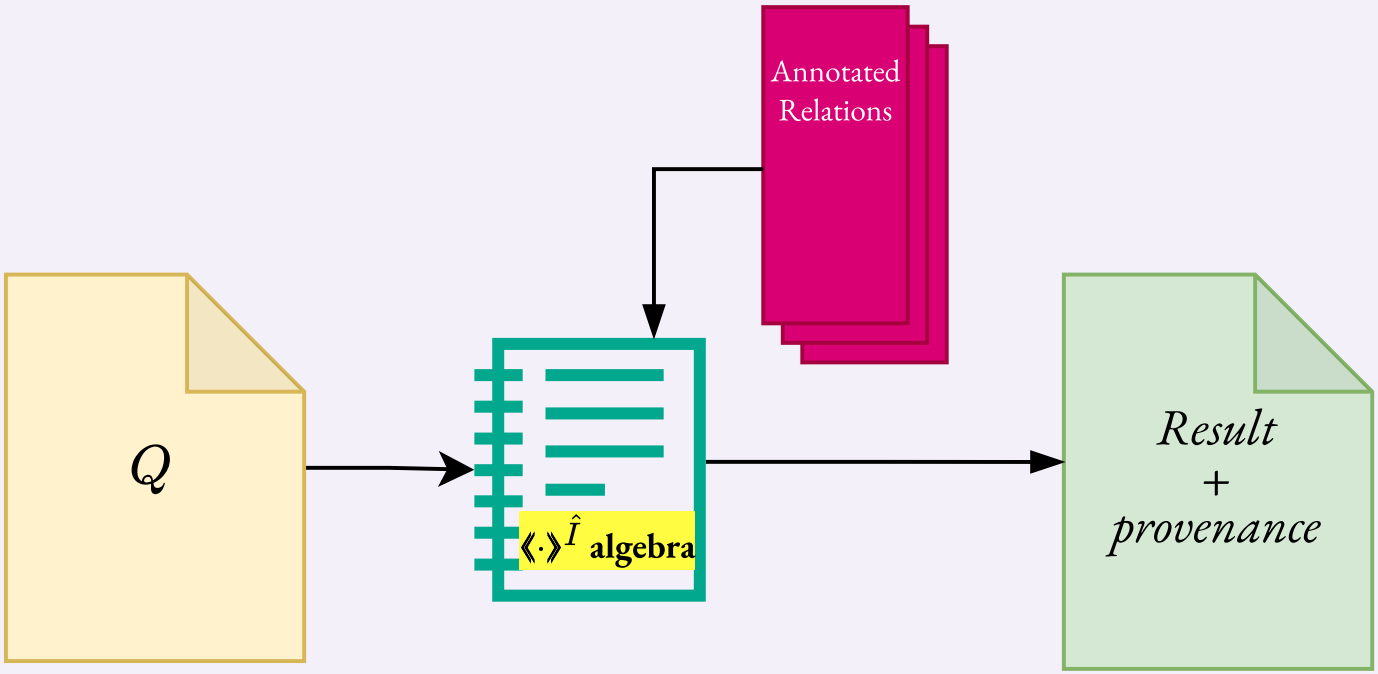
ProvSQL

- existing tools fall short
- ProvSQL, a PostgreSQL extension, fills this gap with a generic, scalabe, and easy-to-deploy solution
- does provenance tracking and probability computation over relational databases
- supports a broad SQL subset

Query language

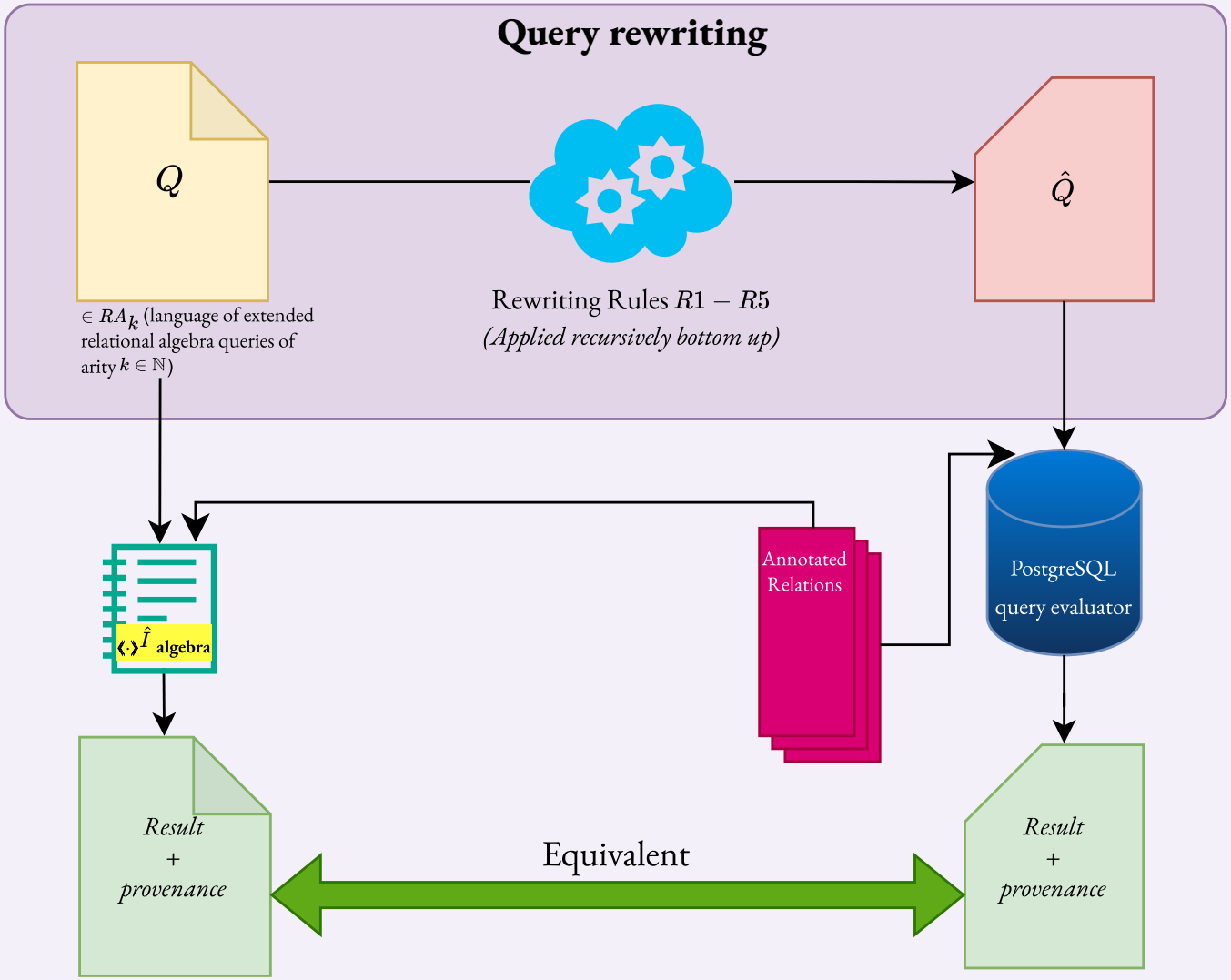
relational algebra + multiset semantics + terminal aggregates

Algebra over annotated relations

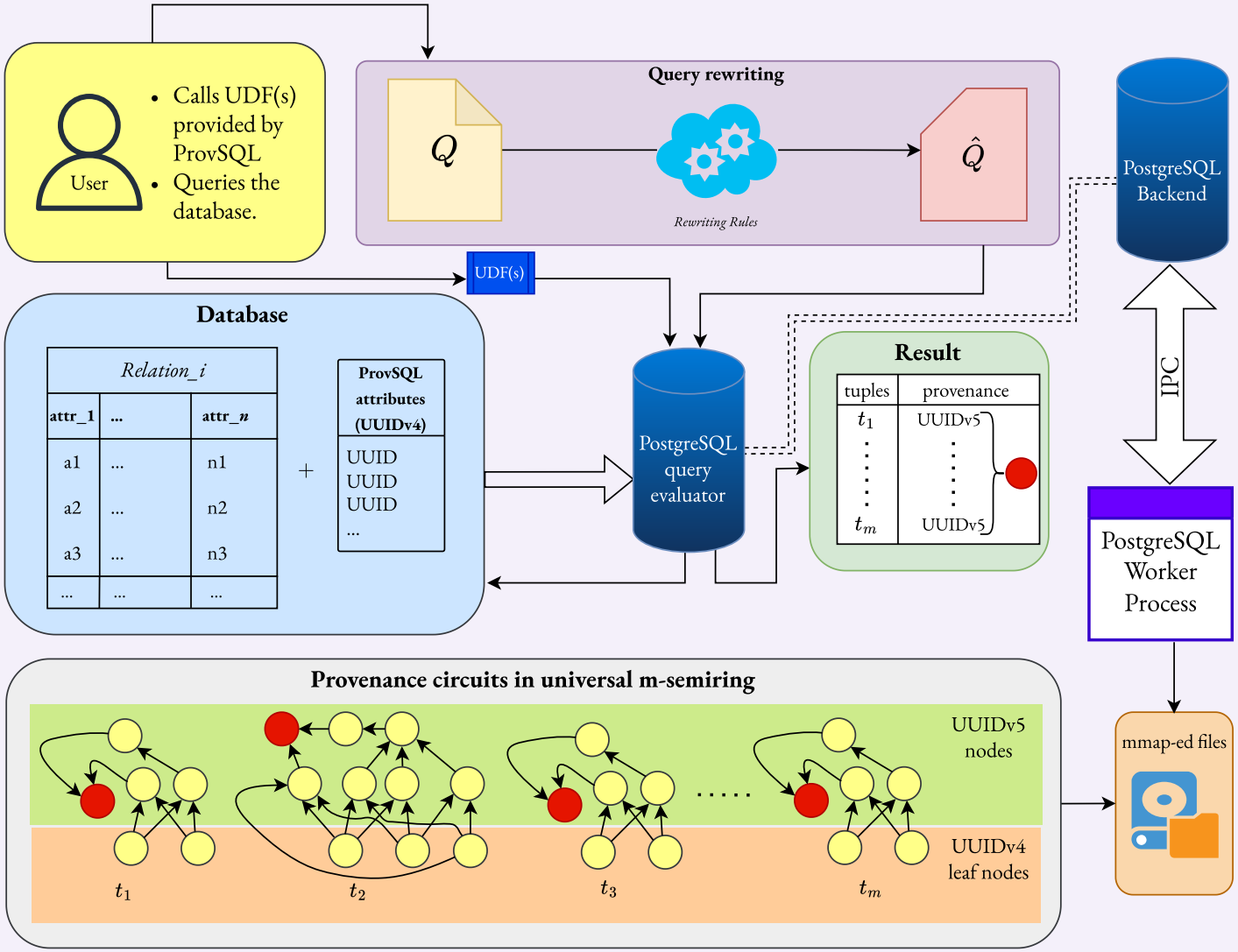


$\in RA_k$ (language of extended relational algebra queries of arity $k \in \mathbb{N}$)

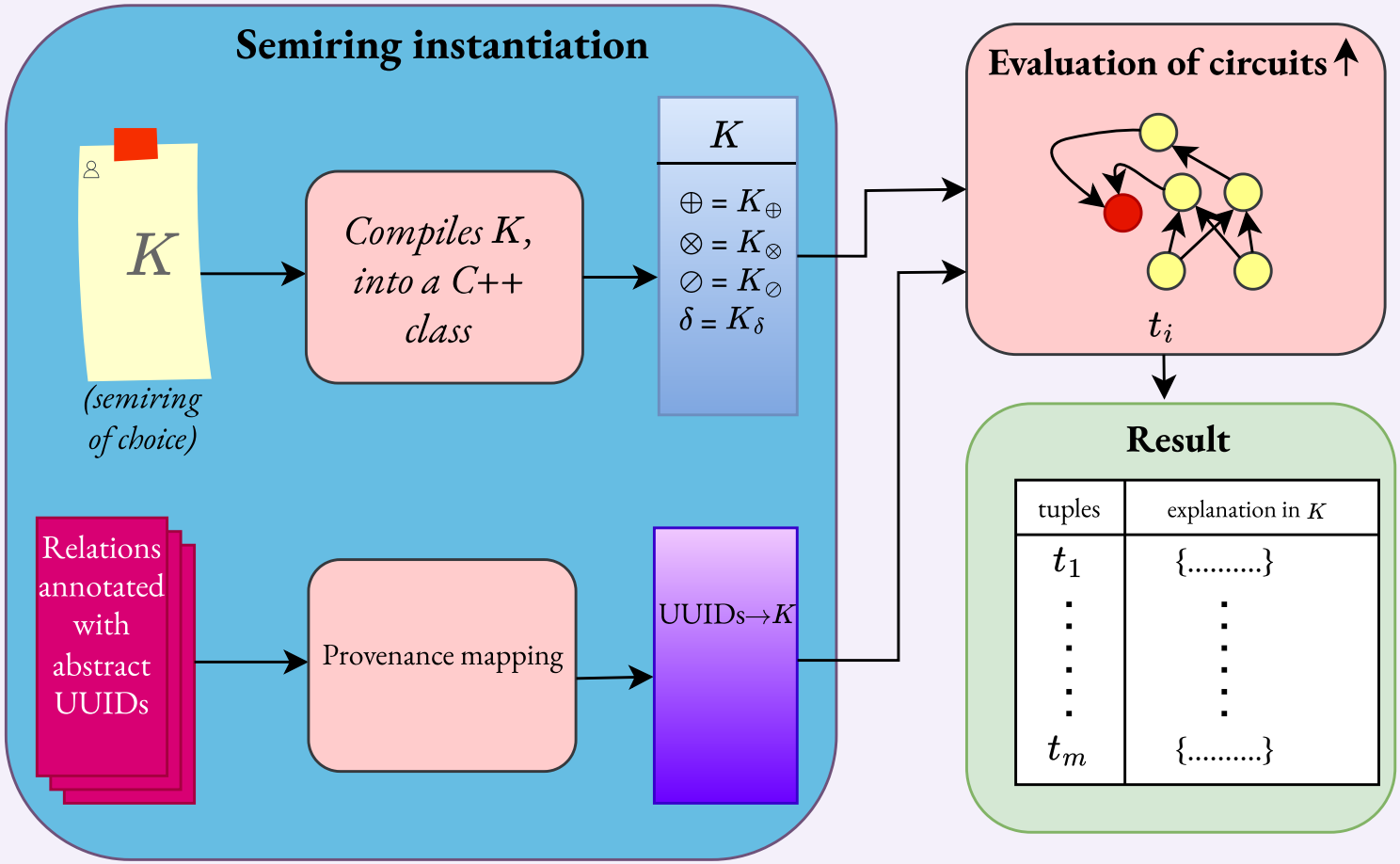
Query rewriting



Provenance tracking



Semiring instantiation



Experimental setup

System: PostgreSQL 16 on i9 16-core Dell Precision with 64 GB RAM, running Debain 12

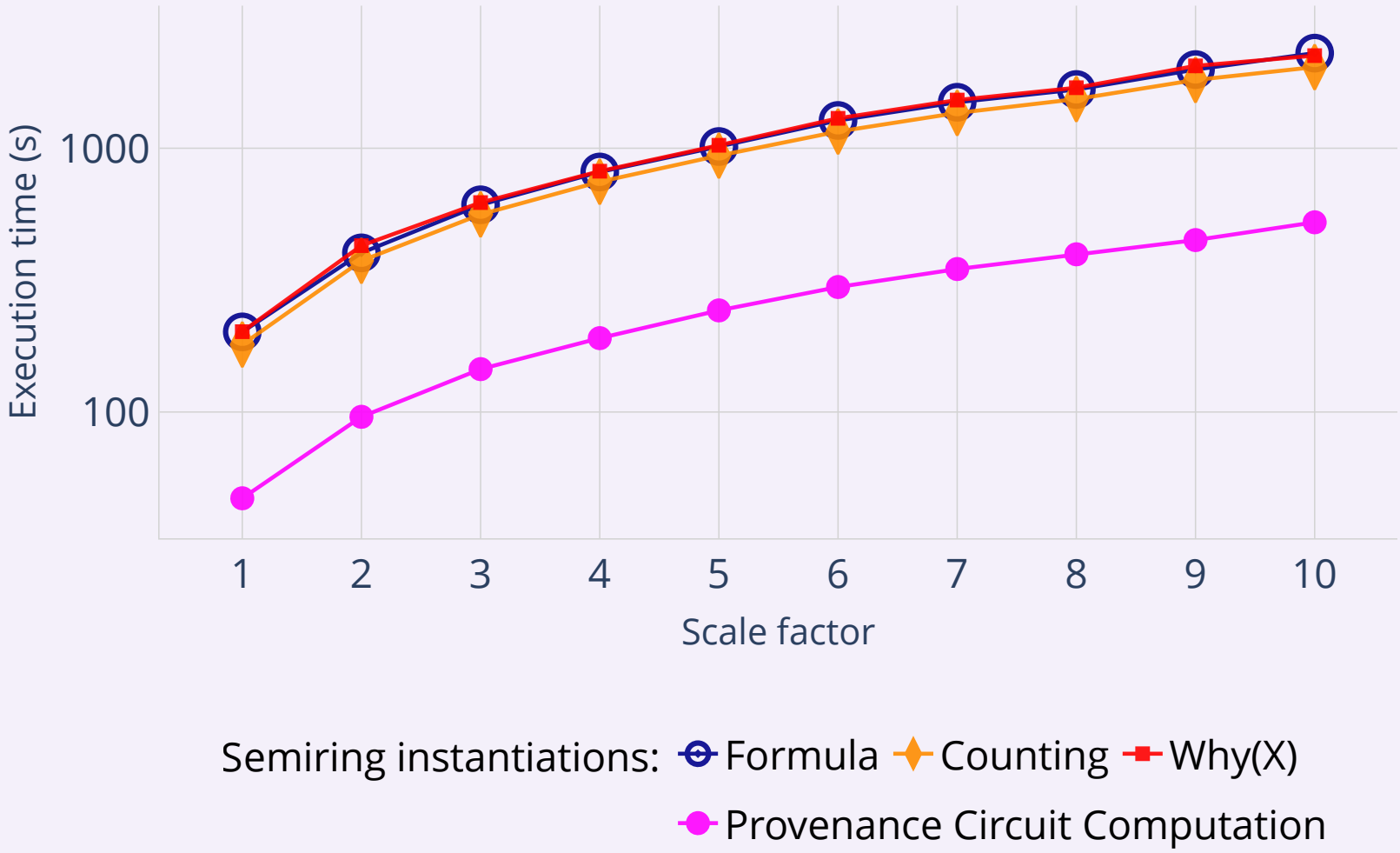
Dataset: TPC-H 3.0.1 (scale factor 1 to 10)

Queries: Selected TPC-H queries + a custom query-load of handwritten queries + modified TPC-H queries

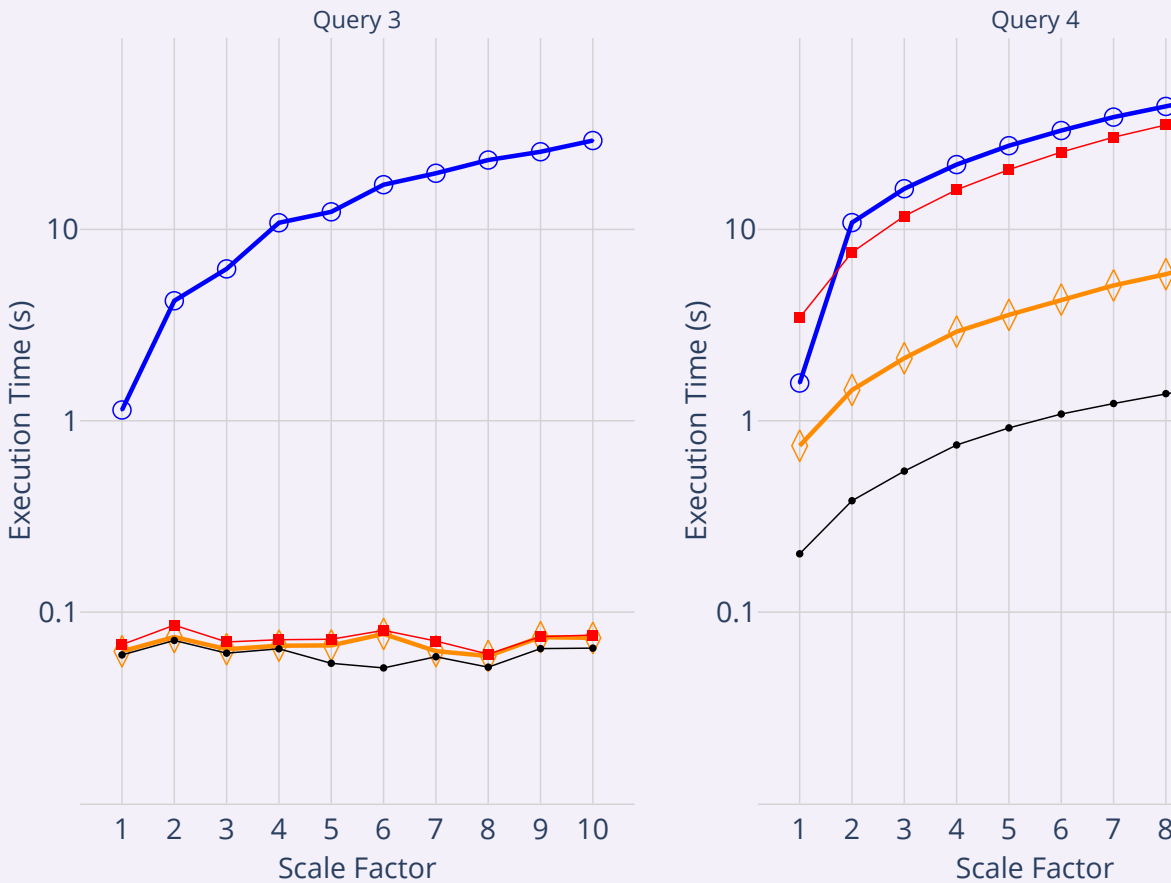
Systems compared with: GProM, MayBMS[1]

[1] MayBMS was run on VM with 50 GB ram and 8 CPU-cores due to support issues on modern systems.

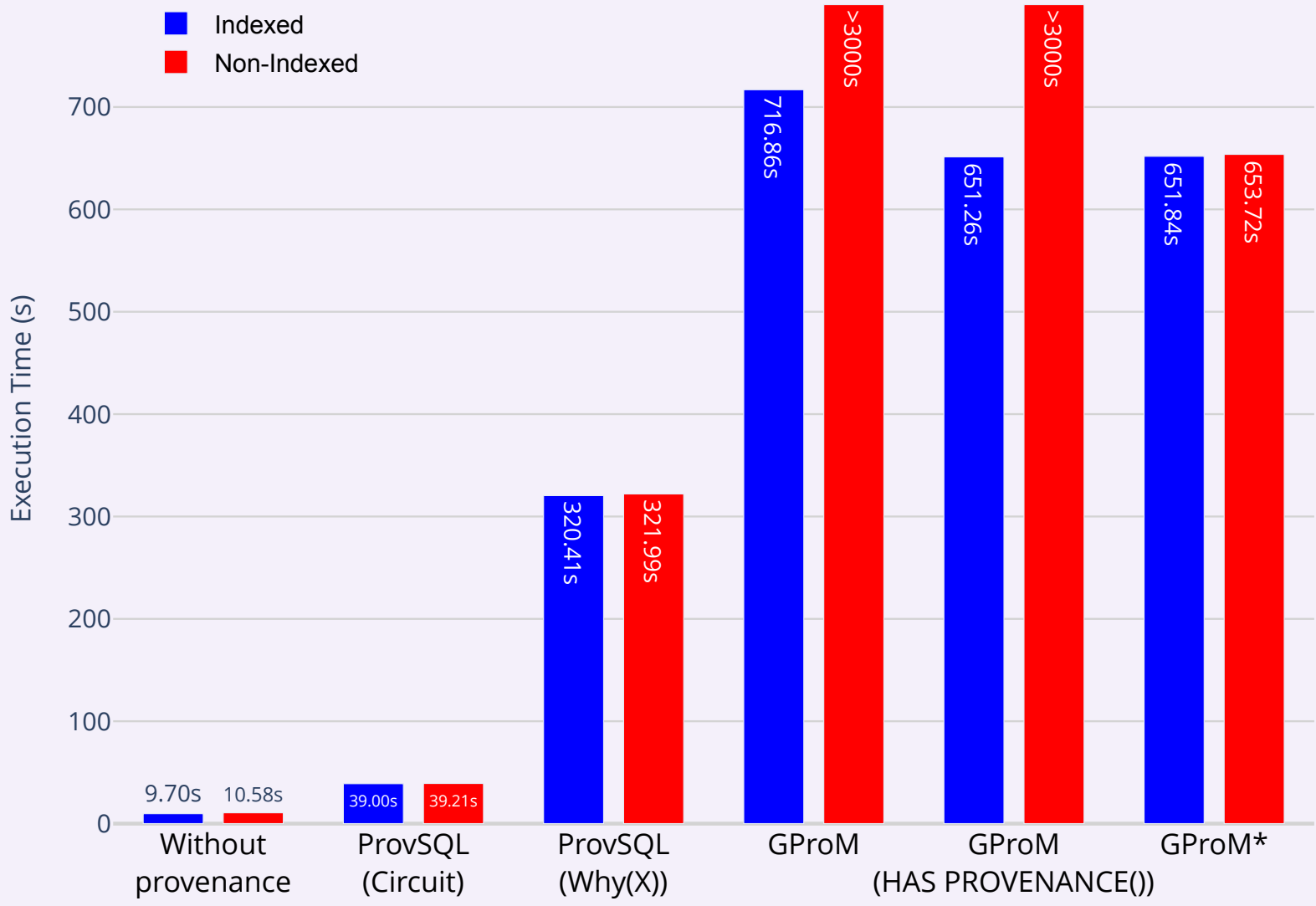
Scalability of semiring instantiation in ProvSQL on a custom query-load



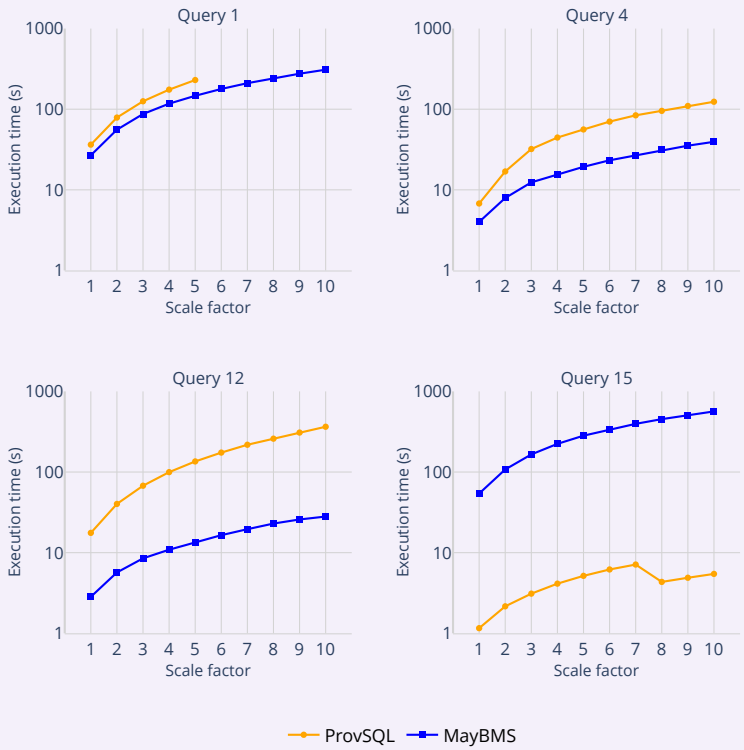
Comparing scalability of GProM vs ProvSQL on modified TPC-H queries



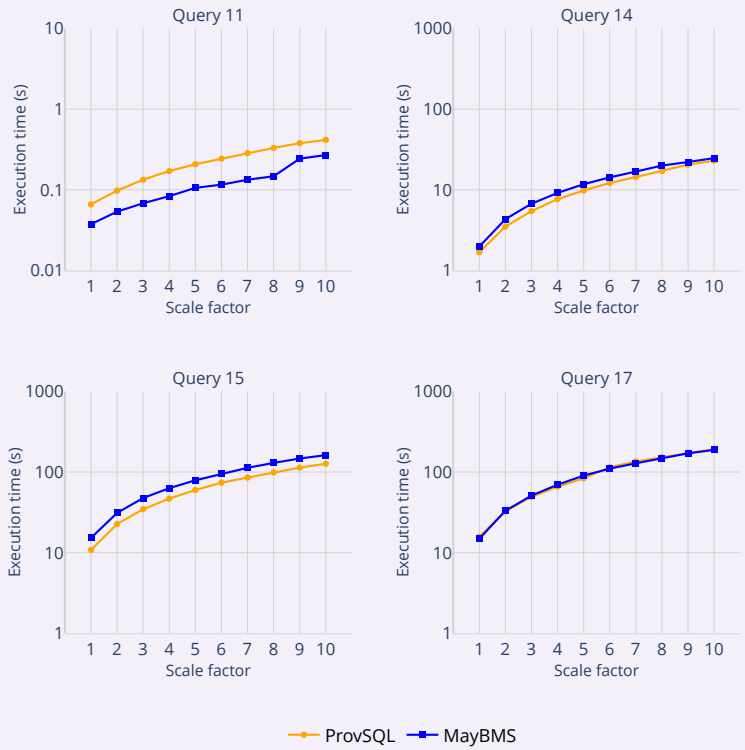
GProM vs ProvSQL on a custom query-load for 1GB TPC-H



MayBMS vs ProvSQL (Probability computation)



(a) Modified TPC-H queries (Safe queries)



(b) Selected queries from our custom query-load (Unsafe queries)

THANK YOU!

ProvSQL: `https://provsql.org/`

Scripts used for running the experiments are available at:

`https://github.com/Aryak320/benchmark_suite.git`