Circuits and Formulas for Datalog over Semirings

Austen Fan Paris Koutris Sudeepa Roy

PODS 2025, Berlin, Germany

Datalog - Refresher

A Datalog program consists of

IDBs (intermediate / final relations)
and EDBs (input relations)

```
// Transitive Closure (TC)
T(x, y) :- E(x, y)
T(x, z) :- T(x, y), E(y, z)
```

```
// single source reachability
reach(x) :- source(x)
reach(x) :- reach(y), E(y, x)
```

A Datalog program is

- **Monadic** if every IDB is unary
- **Rulewise-acyclic** if the body of every rule is acyclic
- **Linear** if every rule has at most one IDB
- **A basic chain program** if every rule is a "path rule" -- S(x, y), T(y, z), U(z, w), …
- A basic chain program corresponds naturally to a **Context-free Grammar (CFG)**
- When the CFG is regular, we get **Regular Path Queries (RPQs)**

```
// basic chain and non-linear
t(x, y) :- a(x, y)
t(x, u) :- t(x, y), a(y, z), b(y, w), t(w, u)
```

t ← a
t ← tabt

Semirings - Refresher

(commutative naturally-ordered) semiring $(\mathbb{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$

```
// Transitive Closure (TC)
T(x, y) :- E(x, y)
T(x, z) :- T(x, y), E(y, z)
```

- $\oplus$, $\otimes$: associative, commutative
- $\mathbf{0}, \mathbf{1}$ neutral elements of $\oplus$, $\otimes$
- $\otimes$ distributes over $\oplus$
- $\mathbf{0} \otimes a = \mathbf{0}$

Boolean Semiring / Set semantics $(\mathbb{B}, \vee, \wedge, \bot, \top)$	----------→	reachability
Tropical Semiring / weighted graphs $(\mathbb{N}, min, +, \infty, 0)$	----------→	shortest paths
Bag semantics $(\mathbb{Z}, +, \times, 0, 1)$	----------→	counting paths

Focus of our paper:

- Absorptive semiring: $a \oplus (a \otimes b) = a$ or, $1 \oplus a = 1$ implies idempotence of $\oplus$: $a \oplus a = a$

Provenance Semirings - Refresher

	R
x_1	Ann
x_2	Bob

	S	
y_1	Ann	Joe
y_2	Ann	Tom
y_3	Bob	Tom

	T
z_1	Joe
z_2	Tom

Annotated Database D

Boolean query **Q:**

$\exists x \exists y\, R(x) \wedge S(x, y) \wedge T(y)$

$F_{Q,D} = (x_1 \wedge y_1 \wedge z_1) \vee (x_1 \wedge y_2 \wedge z_2) \vee (x_2 \wedge y_3 \wedge z_2)$

Boolean provenance
= PosBool[X]

$F_{Q,D} = (x_1 \otimes y_1 \otimes z_1) \oplus (x_1 \otimes y_2 \otimes z_2) \oplus (x_2 \otimes y_3 \otimes z_2)$

General Provenance Semiring

For UCQ / RA^+, $F_{Q,D}$ is poly-size and poly-time computable in |D|

Question:

What about Provenance Semiring for Recursive Datalog

Definition, Computation, Storage

Provenance Semirings, Green-Karvounarakis-Tannen, PODS'07

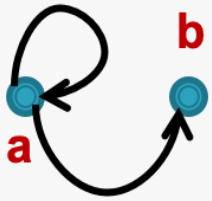
Why Study Provenance for Datalog & Semirings?

- Theory:
  - Provenance on PosBool[X] and other semirings for UCQ is well understood
  - Naturally extends to Recursive Datalog
  - A fundamental problem theoretically

- Applications:
  - Provenance is important for trust, reproducibility
  - Provenance allows efficient updates in results with updated inputs (e.g., deletion or annotation propagation)
  - Recursive computation needed over other semirings in industry
  - Circuits are used in provenance tracking systems with semirings for SQL [ProvSQL, SJMR'18]

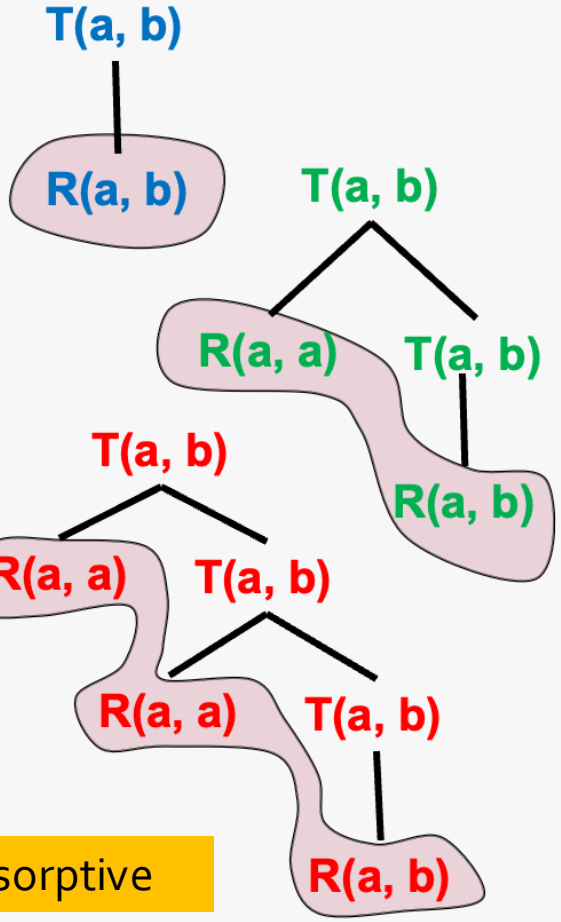
[SJMR'18] ProvSQL: Provenance and Probability Management in PostgreSQL, Senellart-Jachiet-Maniu-Ramusat, PVLDB'18

Prior work on provenance for Datalog and Semirings



R		
p	a	a
q	a	b

T(x, y):- R(x, y)
T(x, y):- R(x, z), T(z, y)



$$\bigvee_{\text{Trees } \tau} \bigwedge_{\text{Leaves t of } \tau} \text{Annot}(t)$$

$$= (q) \vee (p \wedge q) \vee (p \wedge p \wedge q) \vee \ldots$$

- Semantics using Derivation trees or Proof Trees
- Infinitely many trees with a finite form for PosBool[X]: q

May not be well defined for general semiring: infinite sum may not converge [ANPSW'22]

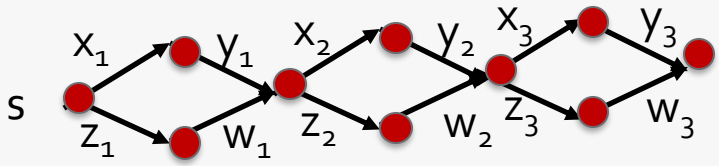
$$(q) \oplus (p \otimes q) \oplus (p \otimes p \otimes q) \oplus \ldots.$$

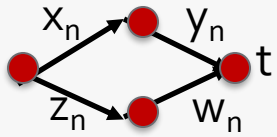
Still = q for absorptive

- keep polynomials that are not "absorbed" by the others – Sorp[X] in [DMRS'14]
  - e.g. $pq + p^2q^3 = pq$ $p^2q + pq^2 = p^2q + pq^2$

[GKT'07] Provenance Semirings, Green-Karvounarakis-Tannen, PODS'07
[DMRS'14] Circuits for Datalog Provenance, Deutch-Milo-Roy-Tannen, ICDT'14
[ANPSW'22] Convergence of Datalog over (Pre-) Semirings, Abo Khamis- Ngo-Pichler-Suciu Wang, PODS'22

Circuits for Datalog Provenance over Semirings





Exponential DNF
but still poly-size formula for st-reachability in CNF
$(x_1y_1 + z_1w_1)\ldots (x_ny_n+z_nw_n)$

Theorem – Lower Bound [DMRT'14]:
Given a PosBool(X)-database D and a Datalog program P, the provenance of tuples in P(D) cannot have a "faithful representation" using Boolean formulas of size polynomial in |D|

[Karchmer-Wigderson, 1988] st-connectivity on n nodes requires $\Omega(\log^2 n)$-depth monotone circuit = $n^{\Omega(\log n)}$-size monotone Boolean formula

Solution: Use Circuits for Provenance!
Formula ⟶ circuit, leaves = EDB facts, internal nodes = $\oplus$ and $\otimes$ gates
Size = # gates
Depth = length of longest root ⟶ leaf path
Formula ≡ a circuit with fan out 1

Theorem – Upper Bound [DMRT'14]: Datalog on Absorptive Semirings Sorp[X] has a poly-size circuit computable in polynomial time (also shows its convergence)

Trace naïve evaluation --- but, bounds were not tight

[DMRT '14]: Circuits for Datalog Provenance, Deutch-Milo-Roy-Tannen, ICDT'14

Questions

m = input size, n = domain size
Circuit size = #gates, Fan-in = 2

- Observation: It suffices to focus on depth – O(log m) vs. $\Omega(\log^{1+\epsilon} m)$
- [Wegener'83]. Let F be a formula over the Boolean semiring of size $|F|$. Then, there exists an equivalent formula (circuit) of depth $O(\log |F|)$.

- Which Datalog programs admit polynomial-size formulas and which do not?
- Which Datalog programs admit circuits of depth $O(\log m)$, and which require circuits of super-logarithmic depth?

- Is Transitive Closure (TC) a canonical case for upper and lower bounds?

We only focus on Absorptive semirings
- Poly-size circuits always possible for any datalog program on absorptive semirings [DMRT'14]
- But, poly-size circuits possible for some more general semirings [DMRT'14]

All questions are open for general semirings

This paper: Partial understanding of these questions
This talk: Overview and ideas of some of the results and ideas

[Wegener'83] Relating Monotone Formula Size and Monotone Depth of Boolean Functions. Ingo Wegener, Inf. Process. Lett.,'83
[DMRT '14]: Circuits for Datalog Provenance, Deutch-Milo-Roy-Tannen, ICDT'14

Basic Chain Datalog

- $P(x,y)$:- $Q_0(x,z_1) \wedge Q_1(z_1,z_2) \wedge \ldots \wedge Q_k(z_k,y)$

- Corresponds to a CFG

- If all rules are left (or right)-linear, then a regular language (RL)

Basic Chain Datalog: Transitive Closure

Theorems: Transitive closure on <u>any absorptive semirings</u>
(1) O(mn)-size and O(n log n)-depth circuits
(2) $O(n^3 \log n)$-size and $O(\log^2 n)$-depth circuits

- Idea (1): Simulate Bellman-Ford for absorptive semirings
  - convert to fan-in 2 with O(log n) $\bigoplus$-gates
- Idea (2): By repeated matrix multiplication
  - M_{ij} = E(i, j) if an edge exists, = 1 for i = j, else = 0
  - M^p computes walks of length $\leqslant$ p
  - $M_{ij}^{p+1} = \sum_{k=1}^{n} M_{ik}^{p} M_{kj}^{p}$
  - Two n x n matrix multiplication by $O(n^3)$ many $\bigotimes$-gates and $O(n^2 \log n)$ many $\bigoplus$-gates
  - Compute M^n by log n times doubling matrix multiplication
  - Absorption matters – (i,i) stays 1 and cycles are absorbed
  - Better bound for dense graphs and parallelization

Basic Chain Datalog: Infinite Regular Languages ≡ TC

Question: Is Transitive Closure (TC) a canonical case for upper and lower bounds of circuit size and depth?

Theorem:
Let Π be a basic chain Datalog program that corresponds to an infinite regular language L.
Then, the provenance polynomial for Π has the same circuit depth and size complexity as TC over any absorptive semiring S.

Size and depth-preserving circuit reductions:

- Reduction from TC to Π
- Reduction from Π to TC

1. Reduction from TC to Π (infinite RL L)

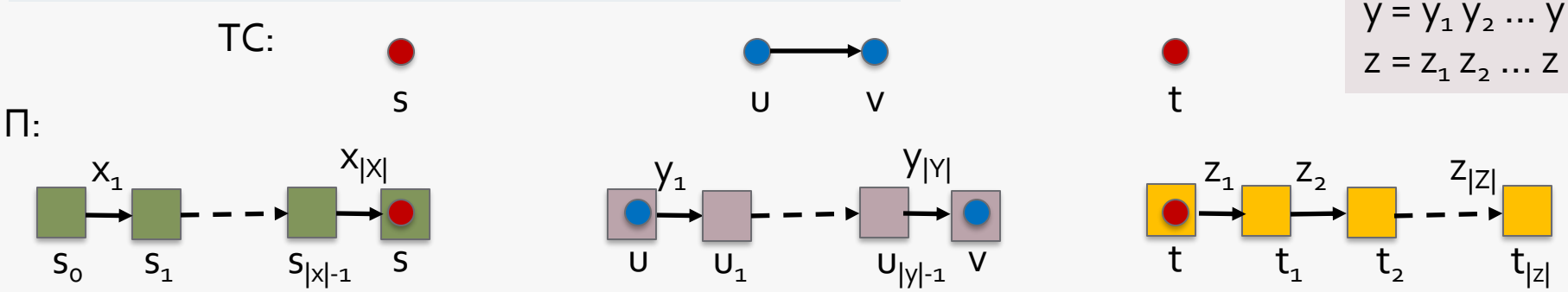
Pumping Lemma for RL:

RL L infinite ⇒ there exists an integer $p \geq 1$ such that every string of length at least p can be written as xyz such that:

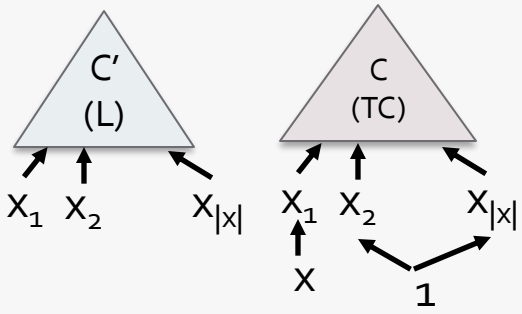
(1) $|y| \geq 1$; (2) $|xy| \leq p$; (3) $(\forall n \geq 0).\ (xy^n z)$ is accepted by L

Encode T(s, t), D for TC to T′(s_0, $t_{|z|}$), D′ for Π

$x = x_1 x_2 \ldots x_{|X|}$
$y = y_1 y_2 \ldots y_{|Y|}$
$z = z_1 z_2 \ldots z_{|Z|}$



- Take a circuit C of Π ⟶ construct circuit C′ for TC
  - Use C as C′
  - have all but one input to connect to 1
  - Fan out > 1, Circuit reduction not a formula reduction
- Circuit C and C′ have the ~same depth and size



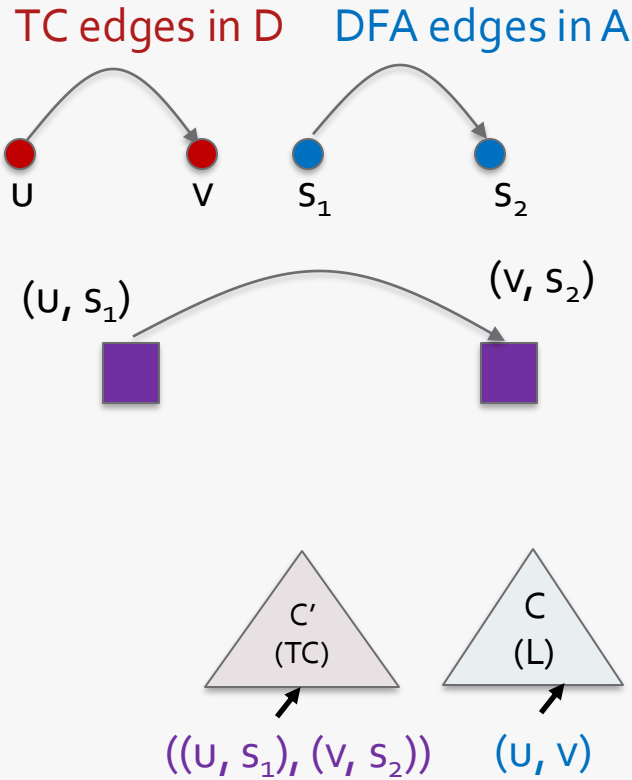
2. Reduction from Π (infinite RL L) to TC

- DFA A for L
- TC on D with E(u, v)

- Product graph G′ of D and A
  - nodes (v, s), u is a "vertex" in D, s is a state in A
  - Edges E′((u, s_1), (v, s_2)) : E(u, v) and A(s_1, s_2)
  - O(m) edges and O(n) vertices

- Take a circuit C′ of TC → construct circuit C for Π
- Run TC on G′ K times for each accepting state t and build K circuits copying C′
- L(u, v) → TC((u, s_0), (v, t)), s_0 start state of A
- Union over accepting states = Take ⊕ over K circuits
- If input to C′ is ((u, s_1), (v, s_2)) then input to C is (u, v)

- Preserves size and depth --- both circuit and formula reduction



Basic Chain Programs: Overview

CFG Language	Circuit Size		Circuit Depth	
	Upper Bound	Lower Bound	Upper Bound	Lower Bound
finite	$O(m)$	$\Omega(m)$	$O(\log n)$	$\Omega(\log n)$
infinite regular	$O(mn)$ $O(n^3 \log n)$	$\Omega(m)$	$O(n \log n)$ $O(\log^2 n)$	$\Omega(\log^2 n)$
infinite	$O(n^5)$	$\Omega(m)$	$O(n^2 \log n)$	$\Omega(\log^2 n)$

(prior work, [DMRT'14])

m = input size
n = active domain size

Some tight bounds
Some gaps

Some other results and observations

To show lower bounds, PosBool[X] suffices for positive semirings S

- Positive semirings: a $\oplus$ b = o => a = o and b = o
- Semiring homomorphism from S to B

Upper bounds: boundedness suffices for O(log m)-depth circuits = poly-size formula

- Bounded = Naïve evaluation converges after $\leqslant$ constant k steps
- Build circuits by "grounding" rules, use O(log m) depth for $\oplus$
- Undecidable – we give conditions for special cases when bounded and = UCQ

$T(x,y)$:- $E(x,y)$, $\quad T(x,y)$:-$A(x) \wedge T(z,y)$
Bounded for absorptive semirings

Basic chain datalogs / CFG are bounded and O(log m)-depth if and only if finite (positive absorptive S)

If Π satisfies the polynomial fringe property (tight proof trees without repeated IDBs on a root to leaf path), then can construct circuits of poly-size and $O(\log^2 m)$-depth over absorptive S

Conclusions and Open Questions

- Building compact provenance semiring circuits efficiently for Datalog is an important problem with both theoretical and practical significance

- Our work only gives partial answers

- Too many theorems for different cases on programs, semirings, … ☹

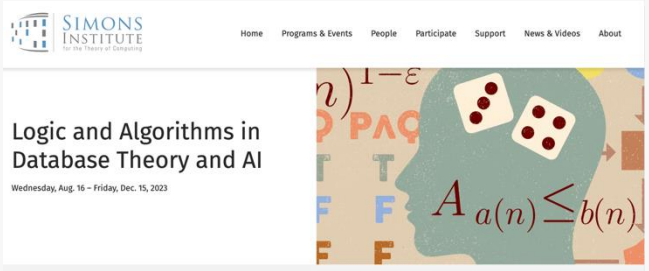
- Can we show a (clean) dichotomy for which Datalog programs admit a polynomial-size formula for absorptive semirings?

- Is every unbounded Datalog program "as hard as" reachability?

- Is there any absorptive semiring for which boundedness differs from boundedness on the Boolean semiring?

- What can we do for semirings beyond absorptive semirings including p-stable semrings?

Thank you



Thanks to Simons Institute, Berkeley for hosting the Fall 2023 program on Database Theory where the research started.