Tight Bounds of Circuits for Sum-Product Queries

Austen Z. Fan, Paris Koutris & Hangdong Zhao

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

PODS 2024

The Sum-Product Polynomial

variables $\mathbf{x} = \{x_1, \ldots, x_n\}$

- Conjunctive Query Q with hypergraph $([n], \mathscr{E})$:

$$Q(\mathbf{x}) \leftarrow \bigwedge_{K \in \mathscr{E}} R_K(\mathbf{x}_K)$$

$\mathbf{x}_K = \{x_i\}_{i \in K}$

- Semiring $\mathbb{S} = (\mathbf{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$
- Instance I

$$p_I^{\mathscr{H}} := \bigoplus_{t \in Q(I)} \bigotimes_{e \in \mathscr{E}} x_{t[e]}^{e}$$

Examples of Polynomials

$$Q(x_1, x_2, x_3) \leftarrow R(x_1, x_2) \wedge S(x_2, x_3)$$

R
(a_1, a_2)
(b_1, a_2)
(c_1, c_2)

S
(a_2, a_3)
(a_2, b_3)
(c_2, c_3)

$x^R_{b_1a_2}$ — (b_1, a_2)

(c_2, c_3) — $x^S_{c_2c_3}$

arithmetic semiring $(\mathbb{N}, +, \cdot, 0, 1)$

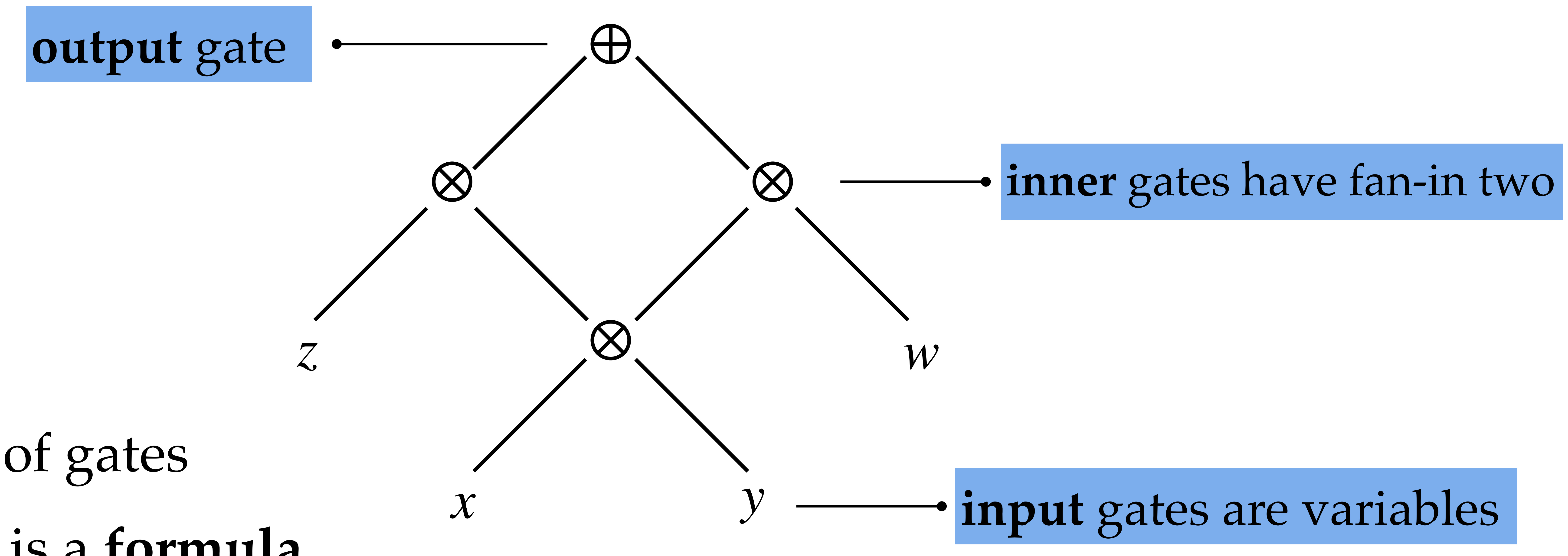
$$p^Q_I = x^R_{a_1a_2} \cdot x^S_{a_2a_3} + x^R_{a_1a_2} \cdot x^S_{a_2b_3} + x^R_{b_1a_2} \cdot x^S_{a_2a_3} + x^R_{b_1a_2} \cdot x^S_{a_2b_3} + x^R_{c_1c_2} \cdot x^S_{c_2c_3}$$

Boolean semiring$(\{0,1\}, \vee, \wedge, 0, 1)$

$$p^Q_I = (x^R_{a_1a_2} \wedge x^S_{a_2a_3}) \vee (x^R_{a_1a_2} \wedge x^S_{a_2b_3}) \vee (x^R_{b_1a_2} \wedge x^S_{a_2a_3}) \vee (x^R_{b_1a_2} \wedge x^S_{a_2b_3}) \vee (x^R_{c_1c_2} \wedge x^S_{c_2c_3})$$

Semiring Circuits

$$p = (x \otimes y \otimes z) \oplus (x \otimes y \otimes w)$$



- **circuit size** := number of gates
- when fan-out is one, it is a **formula**

What is the smallest semiring circuit we can construct for the sum-product polynomial?

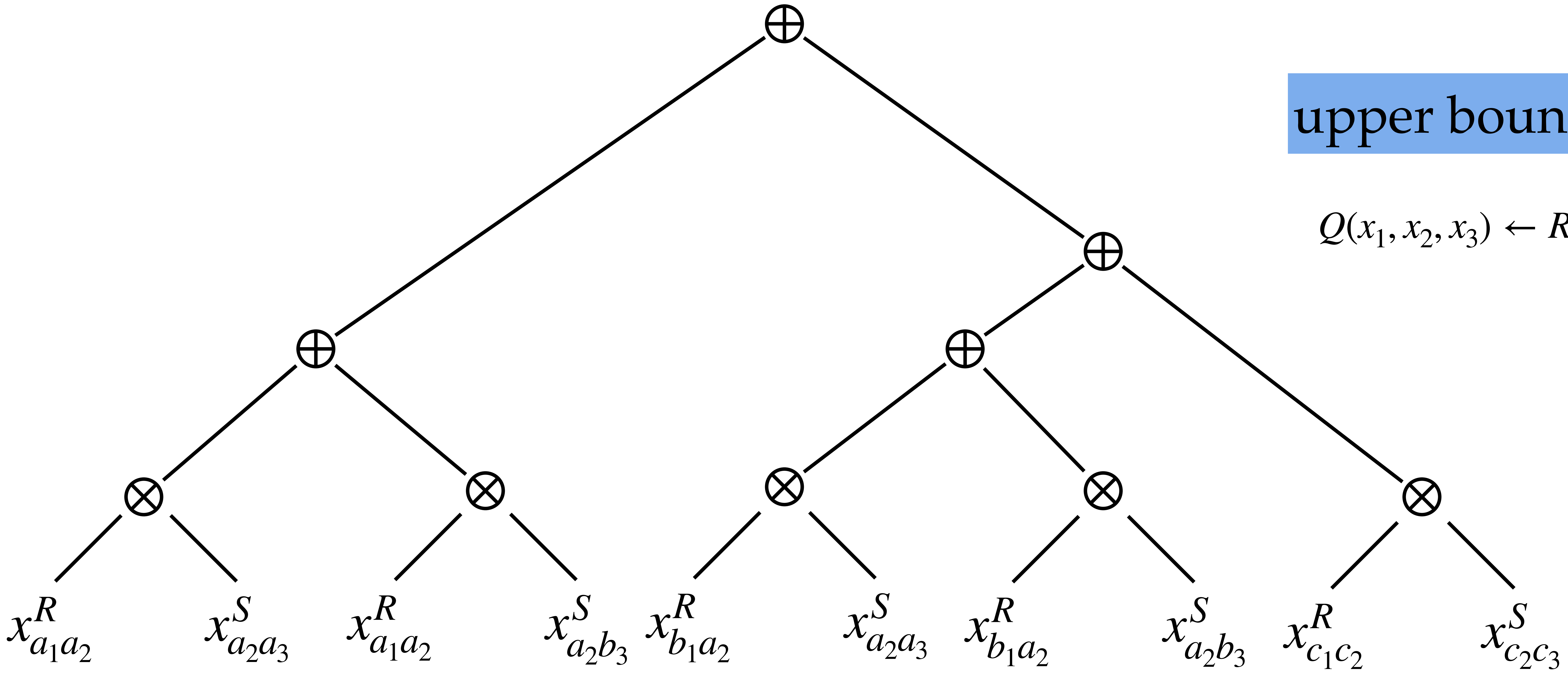
Why Circuits?

1. Circuits are computational models that capture algorithms that *solely* exploit the algebraic structure of the problem
2. Circuits are concise representations of the sum-product polynomial interpreted over the given semiring (captures the *provenance*)

Circuit Construction: Attempt #1

We can always construct a circuit of size linear to the output size $|Q(I)|$

$$p_I^Q = x^R_{a_1a_2} \cdot x^S_{a_2a_3} + x^R_{a_1a_2} \cdot x^S_{a_2b_3} + x^R_{b_1a_2} \cdot x^S_{a_2a_3} + x^R_{b_1a_2} \cdot x^S_{a_2b_3} + x^R_{c_1c_2} \cdot x^S_{c_2c_3}$$



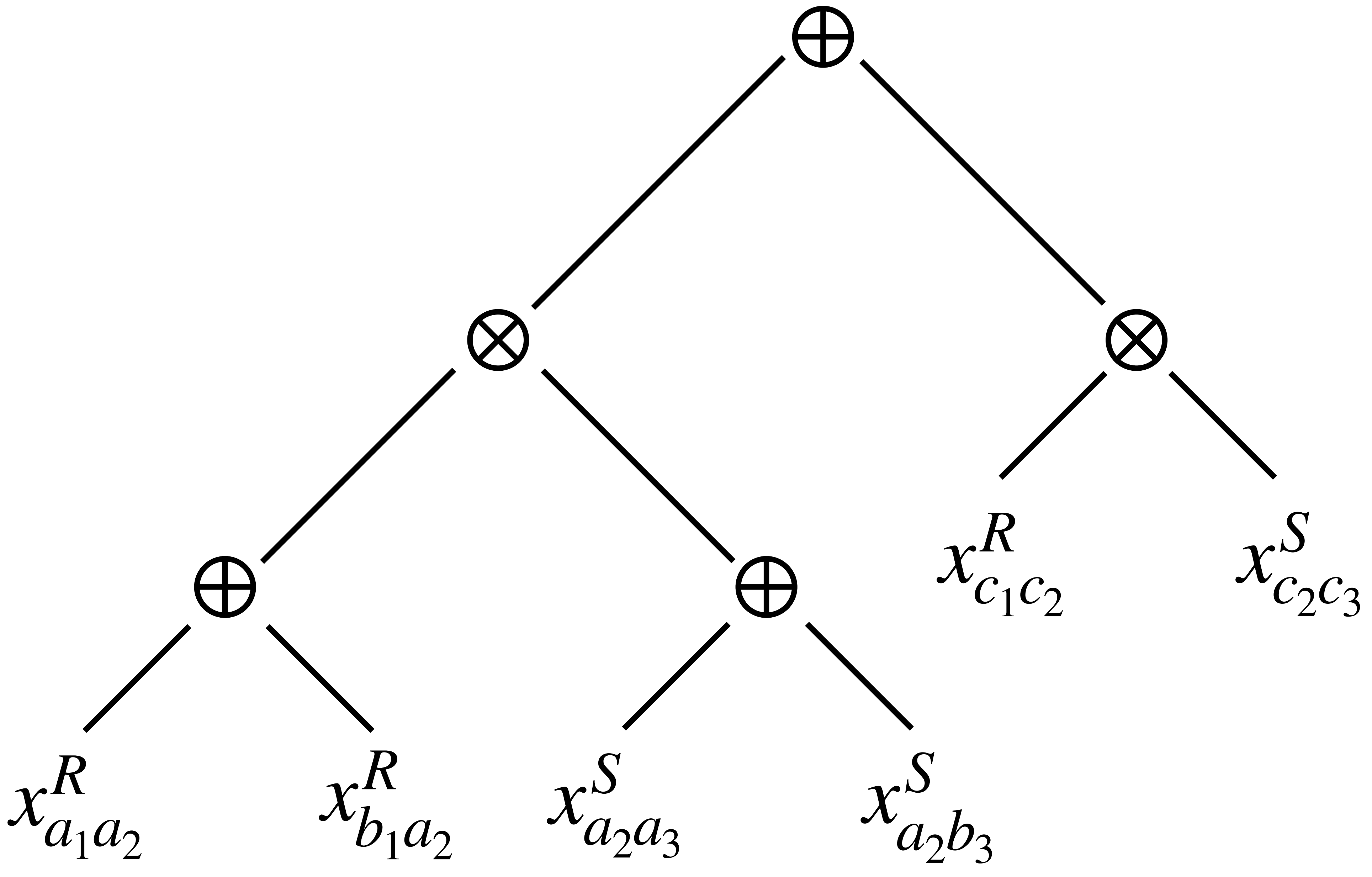
upper bound: $O(N^{\rho^*(Q)})$

$Q(x_1, x_2, x_3) \leftarrow R(x_1, x_2) \wedge S(x_2, x_3)$

Circuit Construction: Attempt #2

We can use the distributive property in a semiring to factorize computation!

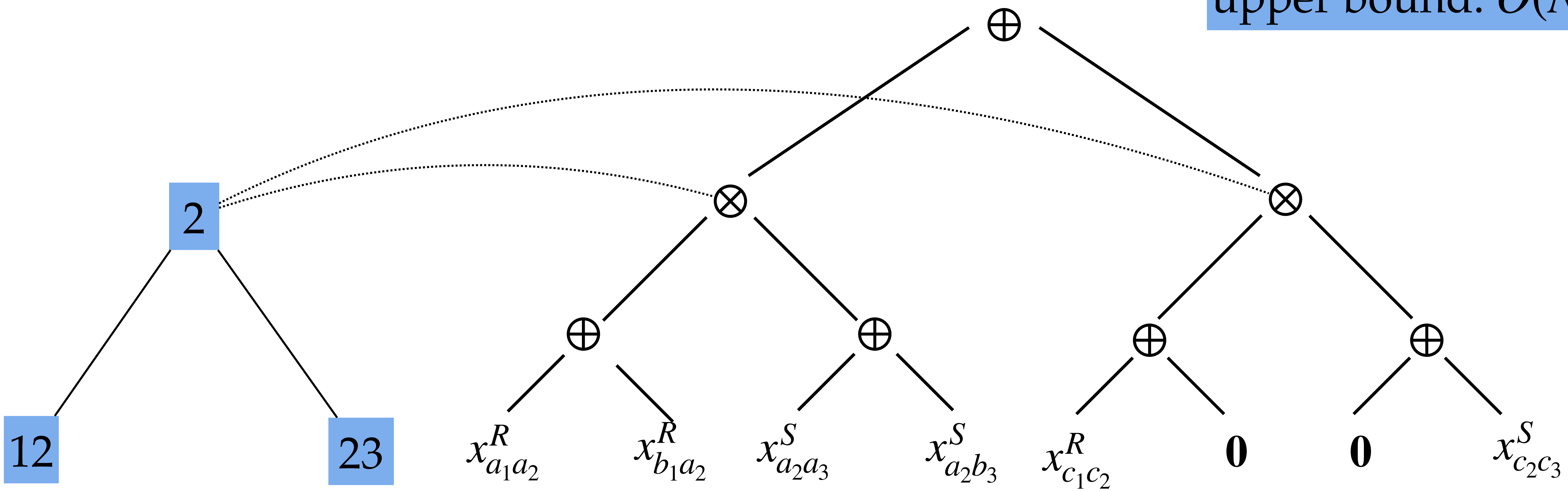
$$p_I^Q = x^R_{a_1a_2} \cdot x^S_{a_2a_3} + x^R_{a_1a_2} \cdot x^S_{a_2b_3} + x^R_{b_1a_2} \cdot x^S_{a_2a_3} + x^R_{b_1a_2} \cdot x^S_{a_2b_3} + x^R_{c_1c_2} \cdot x^S_{c_2c_3}$$



Factorization via Tree Decompositions

In fact, we can guide factorization using any tree decomposition of the query

upper bound: $O(N^{\mathsf{fhw}(Q)})$

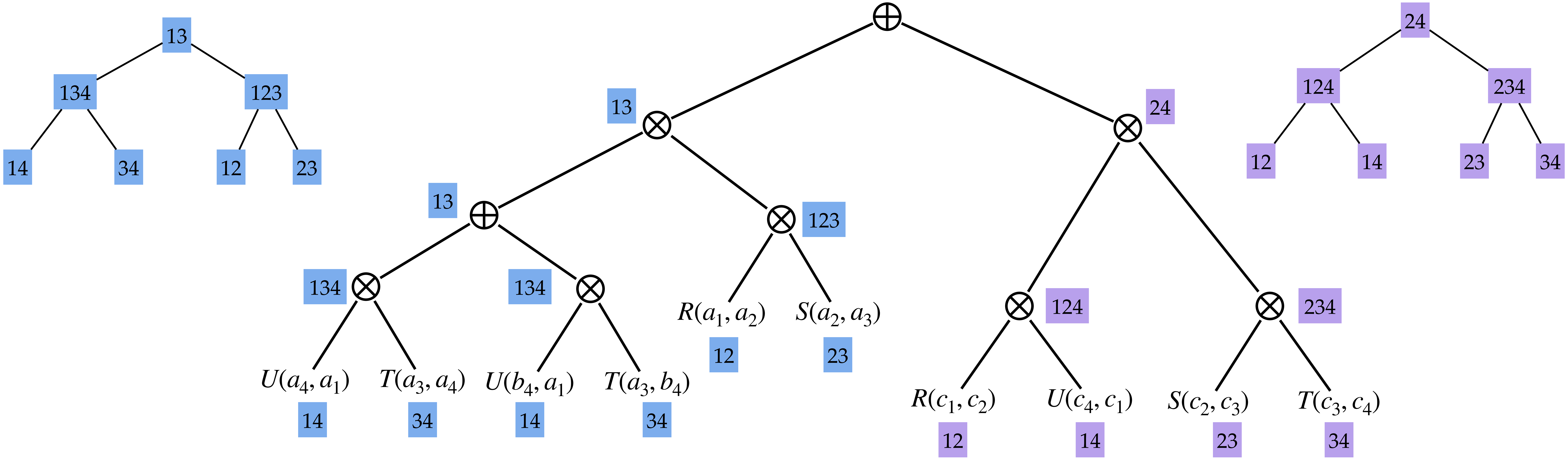


This construction corresponds to a *d-representation* [Olteanu & Zavodny '15]

Circuit Construction: Attempt #3

We can use *multiple* tree decompositions to guide the circuit construction for different parts of the input data

$$Q(x_1, x_2, x_3, x_4) \leftarrow R(x_1, x_2) \wedge S(x_2, x_3) \wedge T(x_3, x_4) \wedge U(x_4, x_1)$$



The Upper Bound

For any *idempotent* semiring, we can construct a circuit that computes the sum-product polynomial with size $O(N^{\mathsf{entw}(Q)})$

$$\mathsf{entw}(Q) = \max_{h \in \overline{\Gamma}^*_n \cap \mathsf{ED}} \; \min_{(\mathcal{T},\chi) \in \mathsf{TD}} \; \max_{v \in V(\mathcal{T})} h(\chi(v))$$

[Khamis et al. '16]

over all entropic functions

best tree decomposition

largest bag

- Idempotency is necessary (the same output can occur in many decompositions)
- The circuit can be constructed in time linear to the output size!

The Lower Bound

For the *tropical* $(\mathbb{Z}, min, +, +\infty, 0)$ and arithmetic semiring, any circuit that computes the sum-product polynomial must have size $\Omega(N^{\mathsf{entw}(Q)})$

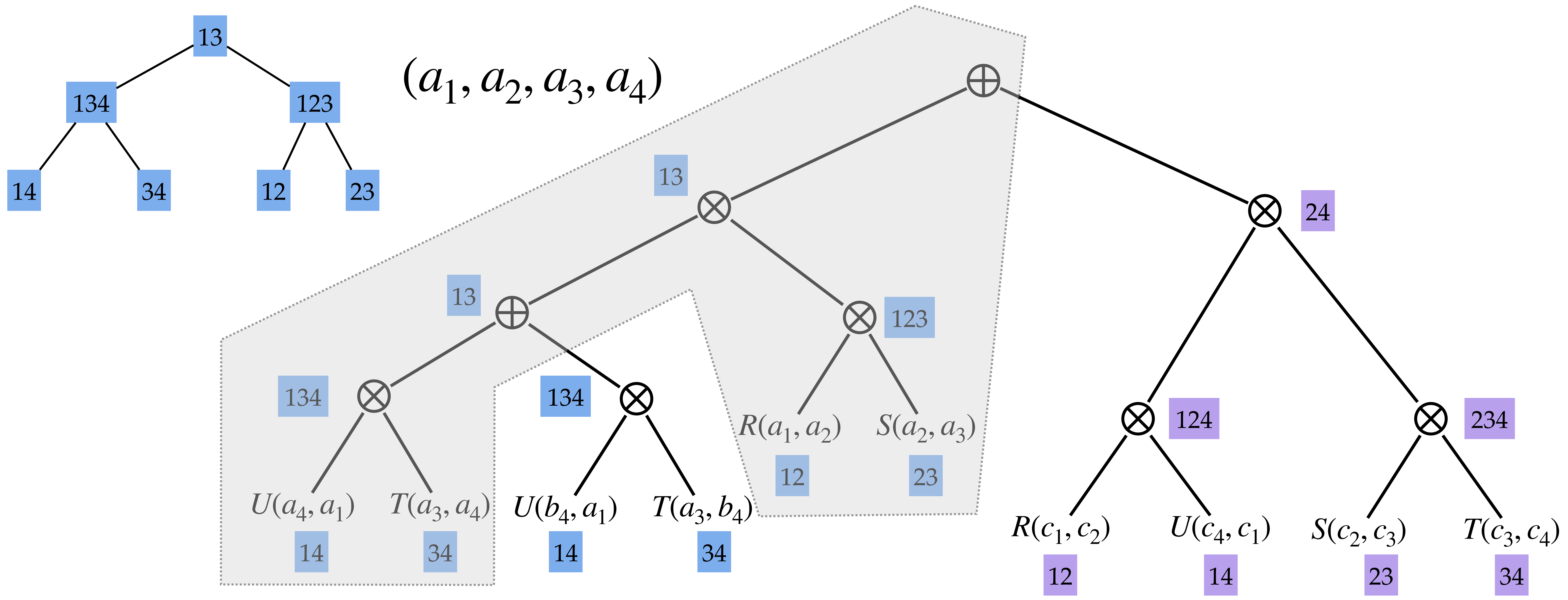
Lower Bound: Key Ideas

$$p_I^Q = x^R_{a_1a_2} \cdot x^S_{a_2a_3} + x^R_{a_1a_2} \cdot x^S_{a_2b_3} + x^R_{b_1a_2} \cdot x^S_{a_2a_3} + x^R_{b_1a_2} \cdot x^S_{a_2b_3} + x^R_{c_1c_2} \cdot x^S_{c_2c_3}$$

1. If the query has no self-joins, the sum-product polynomial is *homogeneous* and *multilinear*
2. Then, the polynomial produced by the circuit is identical to the sum-product polynomial [Jukna '15] (this fails for the Boolean semiring)
3. Hence, we can precisely trace each monomial (output) in the circuit!

Lower Bound: Key Ideas

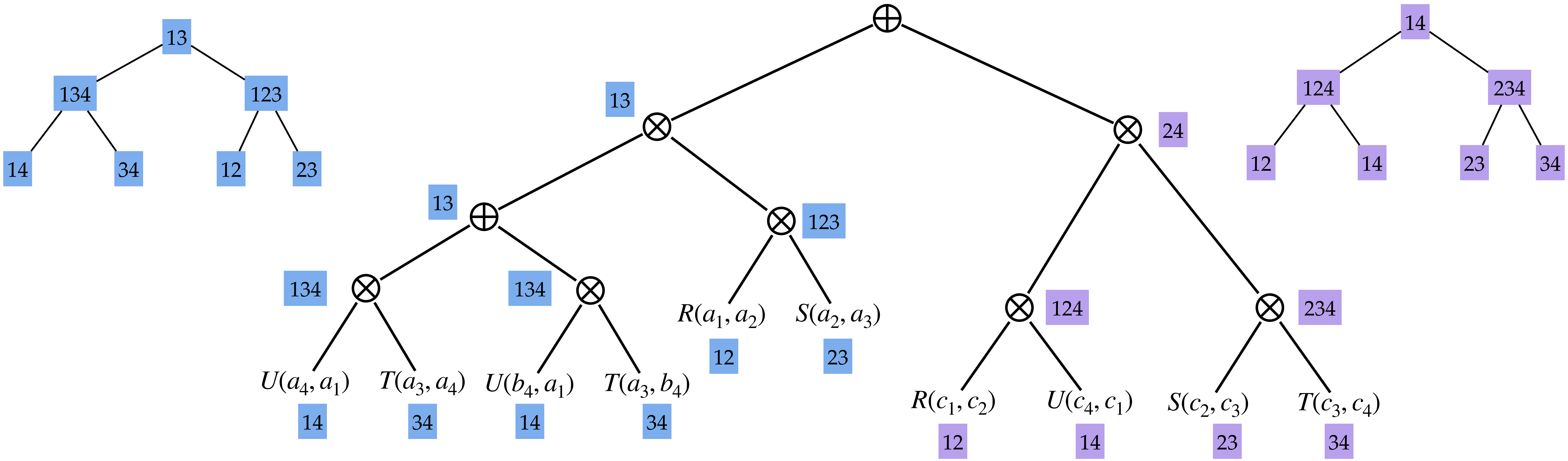
The parse tree of each monomial/output corresponds to a tree decomposition which can be constructed by extracting the "type" of each $\otimes$-gate



$$Q(x_1, x_2, x_3, x_4) \leftarrow R(x_1, x_2) \wedge S(x_2, x_3) \wedge T(x_3, x_4) \wedge U(x_4, x_1)$$

Lower Bound: Key Ideas

- The number of $\otimes$-gates will be bounded by the output of any *disjunctive Datalog rule* that chooses one bag from each decomposition
- Use the worst-case construction for disjunctive rules [Khamis et al. '16]



$$T_{134}(x_1, x_3, x_4) \vee T_{234}(x_2, x_3, x_4) \leftarrow R(x_1, x_2) \wedge S(x_2, x_3) \wedge T(x_3, x_4) \wedge U(x_4, x_1)$$

More in the Paper

1. We show that the same upper & lower bounds hold if we add constraints (*degree-aware entropic width*)
2. We show how to extend the bounds for circuits with multiple outputs (*non-Boolean CQs*)
3. We show how to prove tight upper & lower bounds for circuits that are formulas (*inflationary entropic width*)

Q has semiring *circuits* of linear size $\Leftrightarrow$ Q is *acyclic*

Open Questions

- What is the upper bound for non-idempotent semirings?
- Can we show lower bounds for Boolean semirings?
- What happens when the query has self-joins?

Thank you!