

FAQ: Questions Asked Frequently

Mahmoud Abo Khamis Hung Q. Ngo Atri Rudra



Table of Contents

The FAQ problem

Summary of Results and Approach

The OutsideIn algorithm

The InsideOut algorithm

Choosing a variable ordering

Table of Contents

The FAQ problem

Summary of Results and Approach

The OutsideIn algorithm

The InsideOut algorithm

Choosing a variable ordering

Notations

- ▶ X_i denote variables, $i \in [n] = \{1, \dots, n\}$,

Notations

- ▶ X_i denote variables, $i \in [n] = \{1, \dots, n\}$,
- ▶ x_i are values in discrete domain $\text{Dom}(X_i)$

Notations

- ▶ X_i denote variables, $i \in [n] = \{1, \dots, n\}$,
- ▶ x_i are values in discrete domain $\text{Dom}(X_i)$
- ▶ $\mathbf{x} = (x_1, \dots, x_n) \in \text{Dom}(X_1) \times \dots \times \text{Dom}(X_n)$

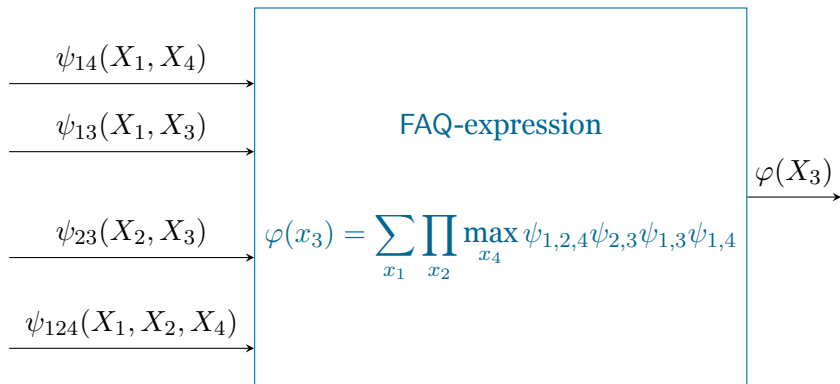
Notations

- ▶ X_i denote variables, $i \in [n] = \{1, \dots, n\}$,
- ▶ x_i are values in discrete domain $\text{Dom}(X_i)$
- ▶ $\mathbf{x} = (x_1, \dots, x_n) \in \text{Dom}(X_1) \times \dots \times \text{Dom}(X_n)$
- ▶ For any $S \subseteq [n]$,

$$\mathbf{x}_S = (x_i)_{i \in S} \in \prod_{i \in S} \text{Dom}(X_i)$$

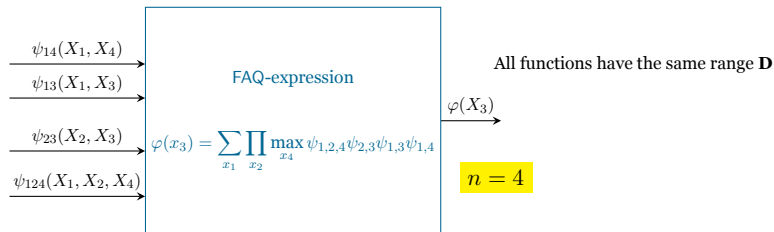
$$\text{e.g. } \mathbf{x}_{\{2,5,8\}} = (x_2, x_5, x_8) \in \text{Dom}(X_2) \times \text{Dom}(X_5) \times \text{Dom}(X_8)$$

Function Aggregate Query: the Problem



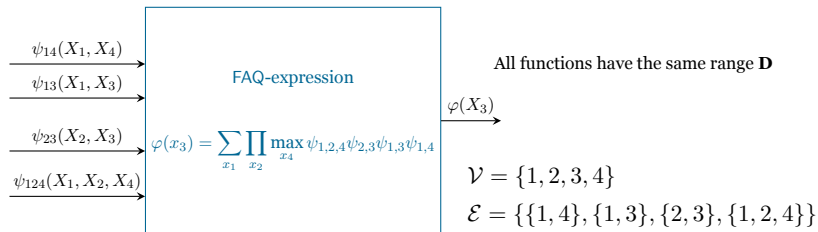
All functions have the same range **D**

Function Aggregate Query: the Input



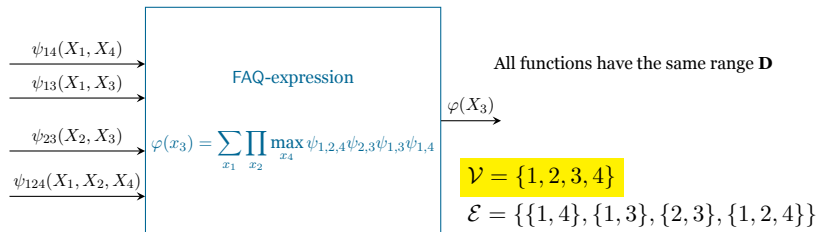
- ▶ n variables X_1, \dots, X_n

Function Aggregate Query: the Input



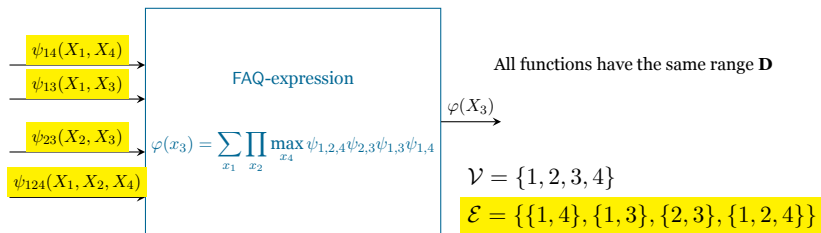
- ▶ n variables X_1, \dots, X_n
- ▶ a multi-hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$

Function Aggregate Query: the Input



- ▶ n variables X_1, \dots, X_n
- ▶ a multi-hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$
 - ▶ Each vertex is a variable (notation overload: $\mathcal{V} = [n]$)

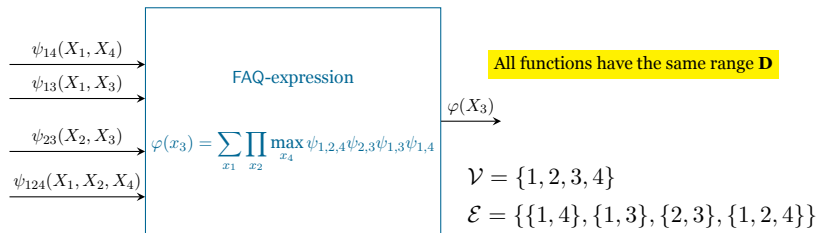
Function Aggregate Query: the Input



- ▶ n variables X_1, \dots, X_n
- ▶ a multi-hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$
 - ▶ Each vertex is a variable (notation overload: $\mathcal{V} = [n]$)
 - ▶ To each hyperedge $S \in \mathcal{E}$ there corresponds a factor ψ_S

$$\psi_S : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \mathbf{D}$$

Function Aggregate Query: the Input

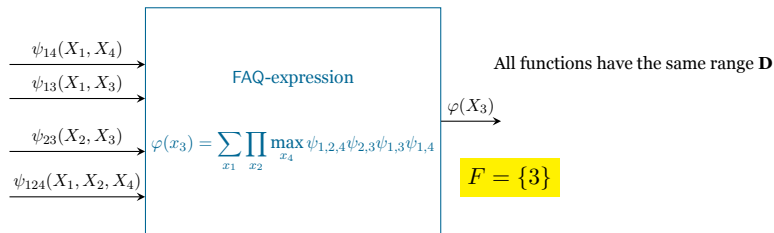


- ▶ n variables X_1, \dots, X_n
- ▶ a multi-hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$
 - ▶ Each vertex is a variable (notation overload: $\mathcal{V} = [n]$)
 - ▶ To each hyperedge $S \in \mathcal{E}$ there corresponds a factor ψ_S

$$\psi_S : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \mathbf{D}$$

$\mathbb{R}_+, \{\text{true}, \text{false}\}, \{0, 1\}, 2^{\mathcal{U}}, \text{etc.}$

Function Aggregate Query: the Input



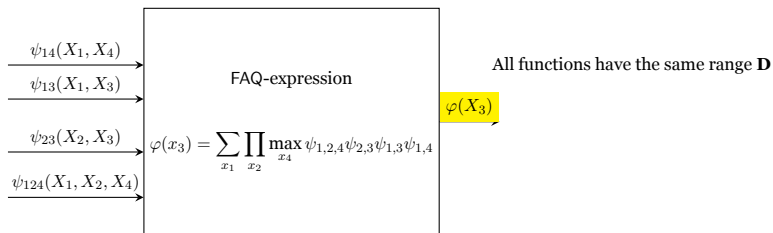
- ▶ n variables X_1, \dots, X_n
- ▶ a multi-hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$
 - ▶ Each vertex is a variable (notation overload: $\mathcal{V} = [n]$)
 - ▶ To each hyperedge $S \in \mathcal{E}$ there corresponds a factor ψ_S

$$\psi_S : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \mathbf{D}$$

\uparrow
 $\mathbb{R}_+, \{\text{true}, \text{false}\}, \{0, 1\}, 2^{\mathcal{U}}, \text{etc.}$

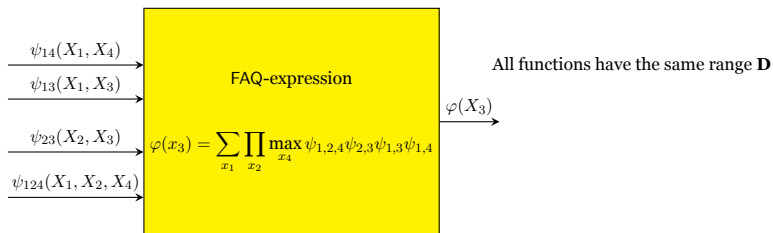
- ▶ a set $F \subseteq \mathcal{V}$ of free variables (wlog, $F = [f] = \{1, \dots, f\}$)

Function Aggregate Query: the Output



- Compute the function $\varphi : \prod_{i \in F} \text{Dom}(X_i) \rightarrow \mathbf{D}$.

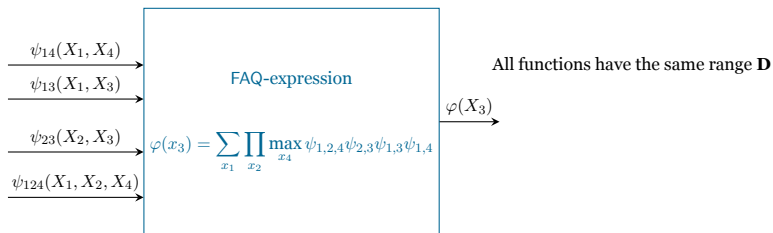
Function Aggregate Query: the Output



- ▶ Compute the function $\varphi : \prod_{i \in F} \text{Dom}(X_i) \rightarrow \mathbf{D}$.
- ▶ φ defined by the *FAQ-expression*

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1} \in \text{Dom}(X_{f+1})}^{(f+1)} \dots \bigoplus_{x_{n-1} \in \text{Dom}(X_{n-1})}^{(n-1)} \bigoplus_{x_n \in \text{Dom}(X_n)}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

Function Aggregate Query: the Output

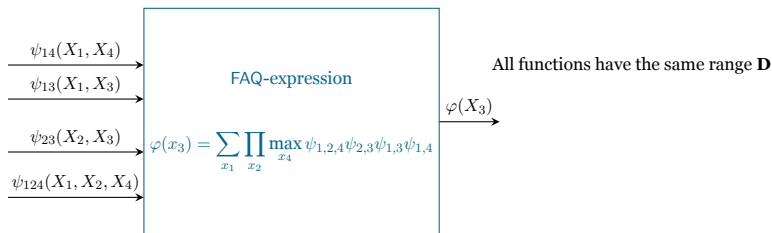


- ▶ Compute the function $\varphi : \prod_{i \in F} \text{Dom}(X_i) \rightarrow \mathbf{D}$.
- ▶ φ defined by the *FAQ-expression*

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1} \in \text{Dom}(X_{f+1})}^{(f+1)} \dots \bigoplus_{x_{n-1} \in \text{Dom}(X_{n-1})}^{(n-1)} \bigoplus_{x_n \in \text{Dom}(X_n)}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ For each $\bigoplus^{(i)}$

Function Aggregate Query: the Output

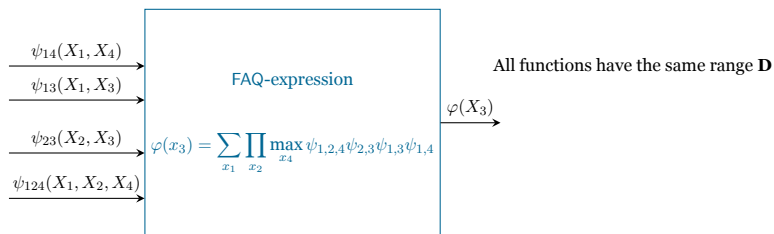


- ▶ Compute the function $\varphi : \prod_{i \in F} \text{Dom}(X_i) \rightarrow \mathbf{D}$.
- ▶ φ defined by the FAQ-expression

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1} \in \text{Dom}(X_{f+1})}^{(f+1)} \dots \bigoplus_{x_{n-1} \in \text{Dom}(X_{n-1})}^{(n-1)} \bigoplus_{x_n \in \text{Dom}(X_n)}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ For each $\bigoplus^{(i)}$
 - ▶ Either $(\mathbf{D}, \bigoplus^{(i)}, \bigotimes)$ is a commutative semiring

Function Aggregate Query: the Output



- ▶ Compute the function $\varphi : \prod_{i \in F} \text{Dom}(X_i) \rightarrow \mathbf{D}$.
- ▶ φ defined by the FAQ-expression

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1} \in \text{Dom}(X_{f+1})}^{(f+1)} \dots \bigoplus_{x_{n-1} \in \text{Dom}(X_{n-1})}^{(n-1)} \bigoplus_{x_n \in \text{Dom}(X_n)}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ For each $\bigoplus^{(i)}$
 - ▶ Either $(\mathbf{D}, \bigoplus^{(i)}, \bigotimes)$ is a *commutative semiring*
 - ▶ Or $\bigoplus^{(i)} = \bigotimes$

SumProd = FAQ-SS *without* free variables

- ▶ SumProd: compute the *constant*

$$\varphi = \bigoplus_{x_1} \bigoplus_{x_2} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

SumProd = FAQ-SS *without* free variables

- ▶ SumProd: compute the *constant*

$$\varphi = \bigoplus_{x_1} \bigoplus_{x_2} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ Where $(\mathbf{D}, \oplus, \otimes)$ is a **commutative semiring**

Additive identity $\mathbf{0} \in \mathbf{D} : \mathbf{0} \oplus e = e \oplus \mathbf{0} = e$

Multiplicative identity $\mathbf{1} \in \mathbf{D} : \mathbf{1} \otimes e = e \otimes \mathbf{1} = e$

Annihilation by 0 $\mathbf{0} \otimes e = e \otimes \mathbf{0} = \mathbf{0}$

Distributive law $a \otimes b \oplus a \otimes c = a \otimes (b \oplus c)$

SumProd = FAQ-SS *without* free variables

- ▶ SumProd: compute the *constant*

$$\varphi = \bigoplus_{x_1} \bigoplus_{x_2} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ Where $(\mathbf{D}, \oplus, \otimes)$ is a **commutative semiring**

Additive identity $\mathbf{0} \in \mathbf{D} : \mathbf{0} \oplus e = e \oplus \mathbf{0} = e$

Multiplicative identity $\mathbf{1} \in \mathbf{D} : \mathbf{1} \otimes e = e \otimes \mathbf{1} = e$

Annihilation by 0 $\mathbf{0} \otimes e = e \otimes \mathbf{0} = \mathbf{0}$

Distributive law $a \otimes b \oplus a \otimes c = a \otimes (b \oplus c)$

- ▶ Common examples (there are many more!)

Boolean $(\{\text{true}, \text{false}\}, \vee, \wedge)$

sum-product $(\mathbb{R}, +, \times)$

max-product $(\mathbb{R}_+, \max, \times)$

set $(2^U, \cup, \cap)$

Problem (CSP = Boolean Join Query)

is there a “truth assignment” \mathbf{x} satisfying all constraints?

Problem (CSP = Boolean Join Query)

is there a “truth assignment” \mathbf{x} satisfying all constraints?

- ▶ Boolean semiring $(\{\text{true}, \text{false}\}, \vee, \wedge)$

$$\bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \bigvee_{\mathbf{x}} \bigwedge_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

Problem (CSP = Boolean Join Query)

is there a “truth assignment” \mathbf{x} satisfying all constraints?

- ▶ Boolean semiring $(\{\text{true}, \text{false}\}, \vee, \wedge)$

$$\bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \bigvee_{\mathbf{x}} \bigwedge_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ constraint $\psi_S : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \{\text{true}, \text{false}\}$ with support S

Problem (CSP = Boolean Join Query)

is there a “truth assignment” \mathbf{x} satisfying all constraints?

- ▶ Boolean semiring $(\{\text{true}, \text{false}\}, \vee, \wedge)$

$$\bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \bigvee_{\mathbf{x}} \bigwedge_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ constraint $\psi_S : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \{\text{true}, \text{false}\}$ with *support* S
- ▶ truth assignment satisfying **all** constraints

Problem (CSP = Boolean Join Query)

is there a “truth assignment” \mathbf{x} satisfying all constraints?

- ▶ Boolean semiring $(\{\text{true}, \text{false}\}, \vee, \wedge)$

$$\bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \bigvee_{\mathbf{x}} \bigwedge_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ constraint $\psi_S : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \{\text{true}, \text{false}\}$ with *support* S
- ▶ truth assignment satisfying **all** constraints
- ▶ over all possible truth assignments

Problem (#CSP = Join cardinality query)

count number of satisfying assignments \mathbf{x}

Problem (#CSP = Join cardinality query)

count number of satisfying assignments \mathbf{x}

- ▶ Sum-product semiring $(\{0, 1\}, +, \times)$

$$\bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \sum_{\mathbf{x}} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

Problem (#CSP = Join cardinality query)

count number of satisfying assignments \mathbf{x}

- ▶ Sum-product semiring $(\{0, 1\}, +, \times)$

$$\bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \sum_{\mathbf{x}} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ constraint $\psi_S : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \{0, 1\}$, where

$$\psi_S(\mathbf{x}) = \begin{cases} 1 & \mathbf{x}_S \text{ satisfies constraint} \\ 0 & \text{otherwise} \end{cases}$$

Problem (#CSP = Join cardinality query)

count number of satisfying assignments \mathbf{x}

- ▶ Sum-product semiring $(\{0, 1\}, +, \times)$

$$\bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \sum_{\mathbf{x}} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ constraint $\psi_S : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \{0, 1\}$, where

$$\psi_S(\mathbf{x}) = \begin{cases} 1 & \mathbf{x}_S \text{ satisfies constraint} \\ 0 & \text{otherwise} \end{cases}$$

- ▶ count 1 only if \mathbf{x} satisfies all constraints

Problem (#CSP = Join cardinality query)

count number of satisfying assignments \mathbf{x}

- ▶ Sum-product semiring $(\{0, 1\}, +, \times)$

$$\bigoplus_{\mathbf{x}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \sum_{\mathbf{x}} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ constraint $\psi_S : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \{0, 1\}$, where

$$\psi_S(\mathbf{x}) = \begin{cases} 1 & \mathbf{x}_S \text{ satisfies constraint} \\ 0 & \text{otherwise} \end{cases}$$

- ▶ count 1 only if \mathbf{x} satisfies **all** constraints
- ▶ sum over all possible truth assignments

Many examples for FAQ-SS w/o free variables

- ▶ Boolean conjunctive query evaluation (Boolean semiring)
- ▶ SAT (Boolean semiring)
- ▶ Quantifier-free conjunctive query evaluation (set semiring)
- ▶ k -colorability (Boolean)
- ▶ Permanent (Sum-Product semiring)
- ▶ Partition function (Sum-Product semiring)
- ▶ etc.

- ▶ Sum-Prod = Marginalize a Product Function problem
 - ▶ Dechter (Artificial Intelligence 1999)
 - ▶ Aji and McEliece (IEEE Trans. Inform. Theory 2000)

Adding free variables

Problem (FAQ-SS with free variables)

Compute the function

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \bigoplus_{\mathbf{x}_{[n]-[f]}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

Adding free variables

Problem (FAQ-SS with free variables)

Compute the function

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) = \bigoplus_{\mathbf{x}_{[n]-[f]}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ In DB jargons: we have Group By!

Problem (Conjunctive Query Evaluation – CQE)

$$\Phi(\mathbf{x}_{[f]}) = \exists_{x_{f+1}} \cdots \exists_{x_n} \bigwedge_{R \in \text{atoms}(\Phi)} R(\text{vars}(R))$$

- ▶ Boolean Semiring ($\{\text{true}, \text{false}\}$), \vee, \wedge)
- ▶ FAQ-SS instance:

$$\varphi(\mathbf{x}_{[f]}) = \bigvee_{x_{f+1}} \cdots \bigvee_{x_n} \bigwedge_{S \in \mathcal{E}} \varphi_S(\mathbf{x}_S).$$

Problem (Matrix Chain Multiplication – MCM)

Let $\mathbf{A}_i = (a_{x,y}^{(i)})$, compute

$$\underbrace{\mathbf{A}}_{p_0 \times p_k} = \underbrace{\mathbf{A}_1}_{p_0 \times p_1} \times \underbrace{\mathbf{A}_2}_{p_1 \times p_2} \times \cdots \times \underbrace{\mathbf{A}_k}_{p_{k-1} \times p_k} .$$

Problem (Matrix Chain Multiplication – MCM)

Let $\mathbf{A}_i = (a_{x,y}^{(i)})$, compute

$$\underbrace{\mathbf{A}}_{p_0 \times p_k} = \underbrace{\mathbf{A}_1}_{p_0 \times p_1} \times \underbrace{\mathbf{A}_2}_{p_1 \times p_2} \times \cdots \times \underbrace{\mathbf{A}_k}_{p_{k-1} \times p_k} .$$

- ▶ Sum-product semiring $(\mathbb{R}, +, \times)$, $(\mathbb{C}, +, \times)$, $(\mathbb{Z}, +, \times)$

Problem (Matrix Chain Multiplication – MCM)

Let $\mathbf{A}_i = (a_{x,y}^{(i)})$, compute

$$\underbrace{\mathbf{A}}_{p_0 \times p_k} = \underbrace{\mathbf{A}_1}_{p_0 \times p_1} \times \underbrace{\mathbf{A}_2}_{p_1 \times p_2} \times \cdots \times \underbrace{\mathbf{A}_k}_{p_{k-1} \times p_k} .$$

- ▶ Sum-product semiring $(\mathbb{R}, +, \times)$, $(\mathbb{C}, +, \times)$, $(\mathbb{Z}, +, \times)$
- ▶ FAQ-SS instance:

Problem (Matrix Chain Multiplication – MCM)

Let $\mathbf{A}_i = (a_{x,y}^{(i)})$, compute

$$\underbrace{\mathbf{A}}_{p_0 \times p_k} = \underbrace{\mathbf{A}_1}_{p_0 \times p_1} \times \underbrace{\mathbf{A}_2}_{p_1 \times p_2} \times \cdots \times \underbrace{\mathbf{A}_k}_{p_{k-1} \times p_k} .$$

- ▶ Sum-product semiring $(\mathbb{R}, +, \times)$, $(\mathbb{C}, +, \times)$, $(\mathbb{Z}, +, \times)$
- ▶ FAQ-SS instance:
 - ▶ Variables: $\text{Dom}(X_i) = [p_i], i \in \{0, \dots, k\}$

Problem (Matrix Chain Multiplication – MCM)

Let $\mathbf{A}_i = (a_{x,y}^{(i)})$, compute

$$\underbrace{\mathbf{A}}_{p_0 \times p_k} = \underbrace{\mathbf{A}_1}_{p_0 \times p_1} \times \underbrace{\mathbf{A}_2}_{p_1 \times p_2} \times \cdots \times \underbrace{\mathbf{A}_k}_{p_{k-1} \times p_k} .$$

- ▶ Sum-product semiring $(\mathbb{R}, +, \times)$, $(\mathbb{C}, +, \times)$, $(\mathbb{Z}, +, \times)$
- ▶ FAQ-SS instance:
 - ▶ Variables: $\text{Dom}(X_i) = [p_i]$, $i \in \{0, \dots, k\}$
 - ▶ Factors: $\psi_{i,i+1}(x_i, x_{i+1}) = a_{x_i, x_{i+1}}^{(i)}$

Problem (Matrix Chain Multiplication – MCM)

Let $\mathbf{A}_i = (a_{x,y}^{(i)})$, compute

$$\underbrace{\mathbf{A}}_{p_0 \times p_k} = \underbrace{\mathbf{A}_1}_{p_0 \times p_1} \times \underbrace{\mathbf{A}_2}_{p_1 \times p_2} \times \cdots \times \underbrace{\mathbf{A}_k}_{p_{k-1} \times p_k} .$$

- ▶ Sum-product semiring $(\mathbb{R}, +, \times)$, $(\mathbb{C}, +, \times)$, $(\mathbb{Z}, +, \times)$
- ▶ FAQ-SS instance:
 - ▶ Variables: $\text{Dom}(X_i) = [p_i]$, $i \in \{0, \dots, k\}$
 - ▶ Factors: $\psi_{i,i+1}(x_i, x_{i+1}) = a_{x_i, x_{i+1}}^{(i)}$
 - ▶ Compute new function $\varphi : [p_0] \times [p_k] \rightarrow \mathbf{D}$

$$\varphi(x_0, x_k) = \sum_{x_1} \cdots \sum_{x_{k-1}} \prod_{i=0}^{k-1} \psi_{i,i+1}(x_i, x_{i+1}).$$

Probabilistic Graphical Models

- ▶ In PGMs, factors are also called *potential functions*

Probabilistic Graphical Models

- ▶ In PGMs, factors are also called *potential functions*
- ▶ Three typical **inference/learning** tasks on PGMs
 - ▶ Compute some marginal distribution
 - ▶ Compute $p(\mathbf{x}_A \mid \mathbf{x}_B)$
 - ▶ Compute $\operatorname{argmax}_{\mathbf{x}_A} p(\mathbf{x}_A \mid \mathbf{x}_B)$ (e.g. MAP queries)

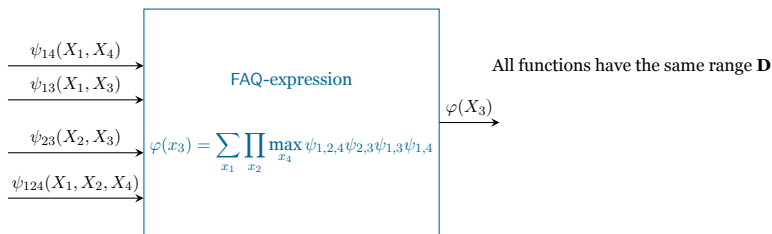
Probabilistic Graphical Models

- ▶ In PGMs, factors are also called *potential functions*
- ▶ Three typical **inference/learning** tasks on PGMs
 - ▶ Compute some marginal distribution
 - ▶ Compute $p(\mathbf{x}_A \mid \mathbf{x}_B)$
 - ▶ Compute $\operatorname{argmax}_{\mathbf{x}_A} p(\mathbf{x}_A \mid \mathbf{x}_B)$ (e.g. MAP queries)
- ▶ These problems are *precisely* FAQ-SS (i.e. SumProd) on
 - ▶ $(\mathbb{R}_+, +, \times)$
 - ▶ $(\mathbb{R}_+, \max, \times)$

Many more examples

- ▶ Discrete Fourier Transform
- ▶ Hollant Problem (as in Holographic algorithms)
- ▶ Graph Homomorphism Problem
- ▶ Weighted CSP
- ▶ List recoverable codes
- ▶ LDPC codes
- ▶ With a squint: also called **Aggregate over Factorized DB**
 - ▶ Bakibayev et al. (VLDB 2014), Olteanu-Zavodny (TODS 2015)
- ▶ etc.

Why the generality of FAQ again?



- ▶ Compute the function $\varphi : \prod_{i \in F} \text{Dom}(X_i) \rightarrow \mathbf{D}$.
- ▶ φ defined by the FAQ-expression

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1} \in \text{Dom}(X_{f+1})}^{(f+1)} \dots \bigoplus_{x_{n-1} \in \text{Dom}(X_{n-1})}^{(n-1)} \bigoplus_{x_n \in \text{Dom}(X_n)}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ For each $\bigoplus^{(i)}$
 - ▶ Either $(\mathbf{D}, \bigoplus^{(i)}, \bigotimes)$ is a *commutative semiring*
 - ▶ Or $\bigoplus^{(i)} = \bigotimes$

Quantified Conjunctive Queries

Problem (QCQ with free variables)

Given $Q_i \in \{\exists, \forall\}$, for $i > f$.

$$\Phi(X_1, \dots, X_f) = Q_{f+1} X_{f+1} \cdots Q_n X_n \left(\bigwedge_{R \in \text{atoms}(\Phi)} R \right),$$

Quantified Conjunctive Queries

Problem (QCQ with free variables)

Given $Q_i \in \{\exists, \forall\}$, for $i > f$.

$$\Phi(X_1, \dots, X_f) = Q_{f+1} X_{f+1} \cdots Q_n X_n \left(\bigwedge_{R \in \text{atoms}(\Phi)} R \right),$$

► FAQ instance:

- $(\{0, 1\}, \{\text{max}, \times\}, \times)$
- Compute the function

$$\varphi(x_1, \dots, x_f) = \bigoplus_{x_{f+1} \in \{0,1\}}^{(f+1)} \cdots \bigoplus_{x_n \in \{0,1\}}^{(n)} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

where

$$\bigoplus^{(i)} = \begin{cases} \text{max} & \text{if } Q_i = \exists \\ \times & \text{if } Q_i = \forall \end{cases}$$

Quantified Conjunctive Queries

Problem (#QCQ)

Given $Q_i \in \{\exists, \forall\}$, for $i > f$, and expression

$$\Phi(X_1, \dots, X_f) = Q_{f+1} X_{f+1} \cdots Q_n X_n \left(\bigwedge_{R \in \text{atoms}(\Phi)} R \right),$$

Count the number of $\mathbf{x}_{[f]}$ for which $\Phi(\mathbf{x}_{[f]}) = \text{true}$

Quantified Conjunctive Queries

Problem (#QCQ)

Given $Q_i \in \{\exists, \forall\}$, for $i > f$, and expression

$$\Phi(X_1, \dots, X_f) = Q_{f+1} X_{f+1} \cdots Q_n X_n \left(\bigwedge_{R \in \text{atoms}(\Phi)} R \right),$$

Count the number of $\mathbf{x}_{[f]}$ for which $\Phi(\mathbf{x}_{[f]}) = \text{true}$

► FAQ instance:

- $(\{0, 1\} \cup \mathbb{R}_+, \{\max, \times, +\}, \times)$
- Compute the **constant**

$$\varphi = \sum_{x_1} \cdots \sum_{x_f} \bigoplus_{x_{f+1} \in \{0,1\}}^{(f+1)} \cdots \bigoplus_{x_n \in \{0,1\}}^{(n)} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

where

$$\bigoplus^{(i)} = \begin{cases} \max & \text{if } Q_i = \exists \\ \times & \text{if } Q_i = \forall \end{cases}$$

Table of Contents

The FAQ problem

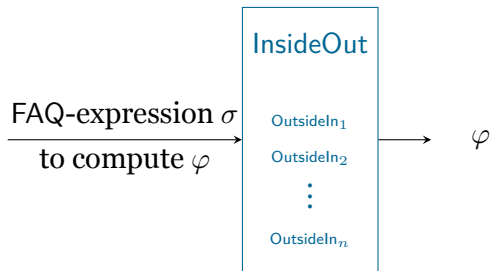
Summary of Results and Approach

The OutsideIn algorithm

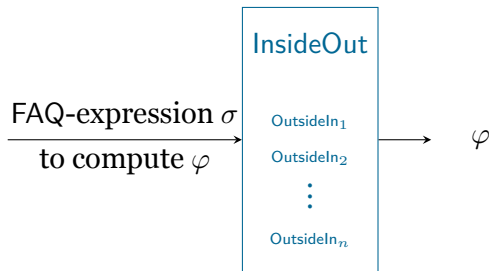
The InsideOut algorithm

Choosing a variable ordering

The Algorithm

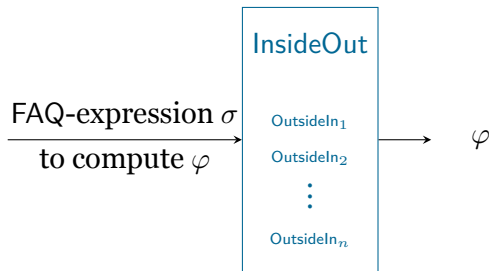


The Algorithm



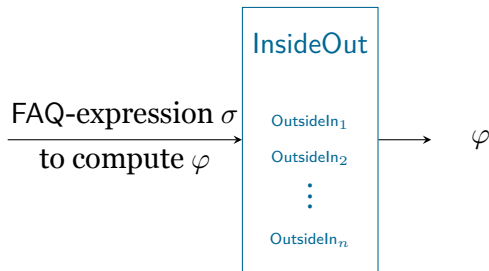
- ▶ InsideOut represents **dynamic programming**

The Algorithm



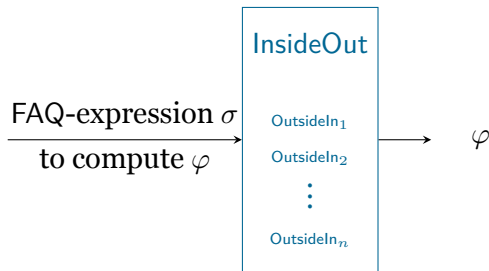
- ▶ InsideOut represents **dynamic programming**
 - ▶ is variable elimination with new twists
 - ▶ Variable Elimination (Zhang-Poole, JAIR 1996)
 - ▶ Bucket Elimination (Dechter, Artif. Intell. 1999)

The Algorithm



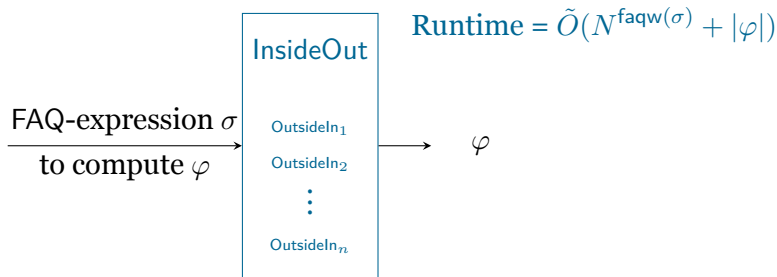
- ▶ InsideOut represents **dynamic programming**
 - ▶ is variable elimination with new twists
 - ▶ Variable Elimination (Zhang-Poole, JAIR 1996)
 - ▶ Bucket Elimination (Dechter, Artif. Intell. 1999)
- ▶ Outsideln represents **backtracking search**

The Algorithm



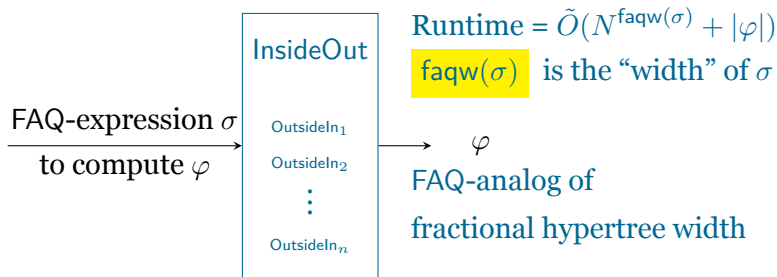
- ▶ InsideOut represents **dynamic programming**
 - ▶ is variable elimination with new twists
 - ▶ Variable Elimination (Zhang-Poole, JAIR 1996)
 - ▶ Bucket Elimination (Dechter, Artif. Intell. 1999)
- ▶ Outsideln represents **backtracking search**
 - ▶ any worst-case optimal join algorithm, such as
 - ▶ NPRR (Ngo et al., PODS 2012)
 - ▶ Leapfrog-Triejoin (Veldhuizen, ICDT'2014)

The Algorithm



- ▶ InsideOut represents **dynamic programming**
 - ▶ is variable elimination with new twists
 - ▶ Variable Elimination (Zhang-Poole, JAIR 1996)
 - ▶ Bucket Elimination (Dechter, Artif. Intell. 1999)
- ▶ Outsideln represents **backtracking search**
 - ▶ any worst-case optimal join algorithm, such as
 - ▶ NPRR (Ngo et al., PODS 2012)
 - ▶ Leapfrog-Triejoin (Veldhuizen, ICDT'2014)

The Algorithm



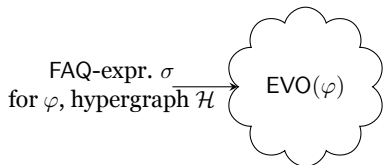
- ▶ InsideOut represents **dynamic programming**
 - ▶ is variable elimination with new twists
 - ▶ Variable Elimination (Zhang-Poole, JAIR 1996)
 - ▶ Bucket Elimination (Dechter, Artif. Intell. 1999)
- ▶ Outsideln represents **backtracking search**
 - ▶ any worst-case optimal join algorithm, such as
 - ▶ NPRR (Ngo et al., PODS 2012)
 - ▶ Leapfrog-Triejoin (Veldhuizen, ICDT'2014)

Technical Contributions

FAQ-expr. σ
for φ , hypergraph \mathcal{H}

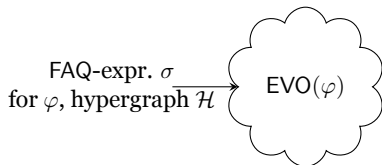
Technical Contributions

$EVO(\varphi)$ = set of expressions
“semantically equivalent” to φ



Technical Contributions

$EVO(\varphi)$ = set of expressions
“semantically equivalent” to φ

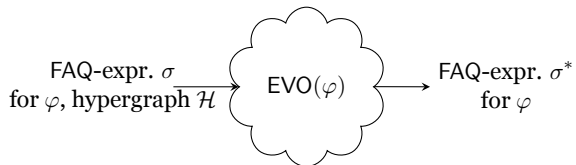


$$\varphi(x_3) = \sum_{x_2} \sum_{x_1} \sum_{x_5} \max_{x_4} \psi_{1,2,4} \psi_{2,3} \psi_{1,3} \psi_{1,4} \psi_{2,5}$$

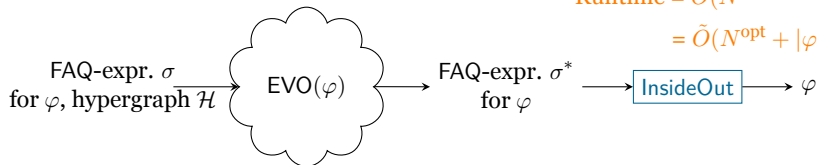
$$\varphi(x_3) = \sum_{x_1} \sum_{x_2} \max_{x_4} \sum_{x_5} \psi_{1,2,4} \psi_{2,3} \psi_{1,3} \psi_{1,4} \psi_{2,5}$$

Technical Contributions

$$\sigma^* = \arg \min_{\tau \in \text{EVO}(\varphi)} \text{faqw}(\tau)$$



Technical Contributions

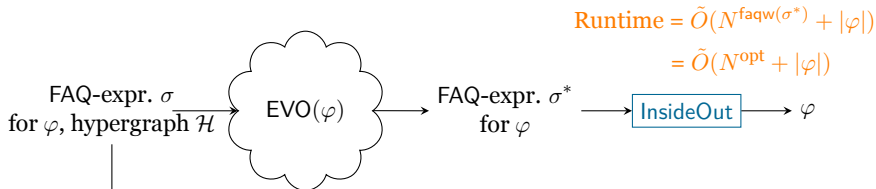


$$\sigma^* = \arg \min_{\tau \in \text{EVO}(\varphi)} \text{faqw}(\tau)$$

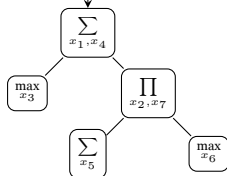
$$\text{Runtime} = \tilde{O}(N^{\text{faqw}(\sigma^*)} + |\varphi|)$$

$$= \tilde{O}(N^{\text{opt}} + |\varphi|)$$

Technical Contributions

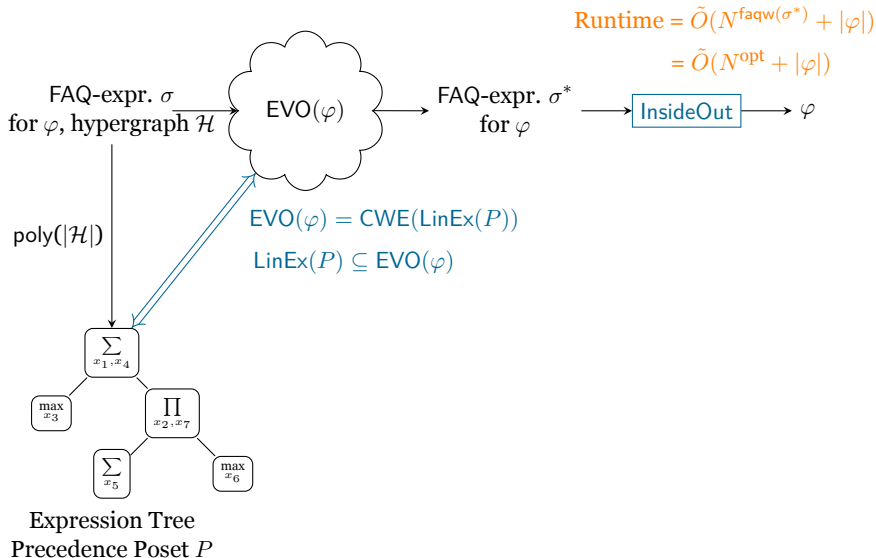


$\text{poly}(|\mathcal{H}|)$

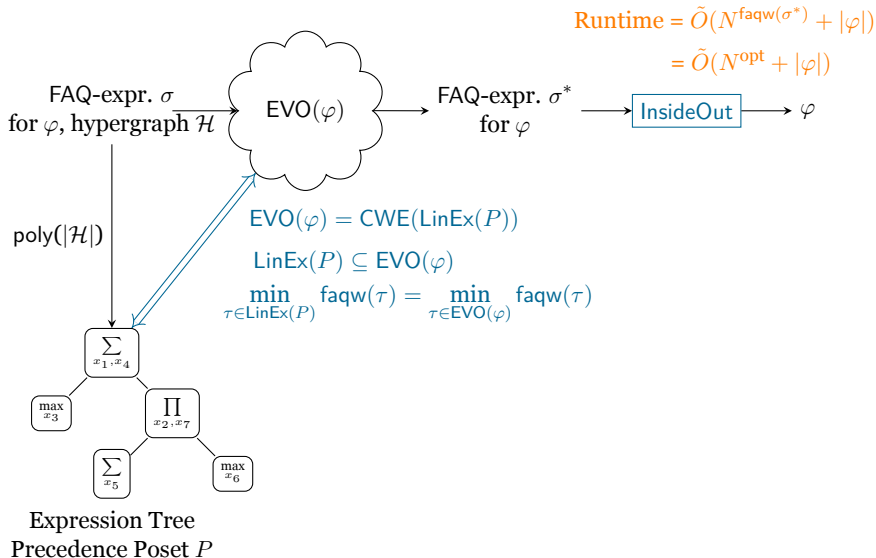


Expression Tree
Precedence Poset P

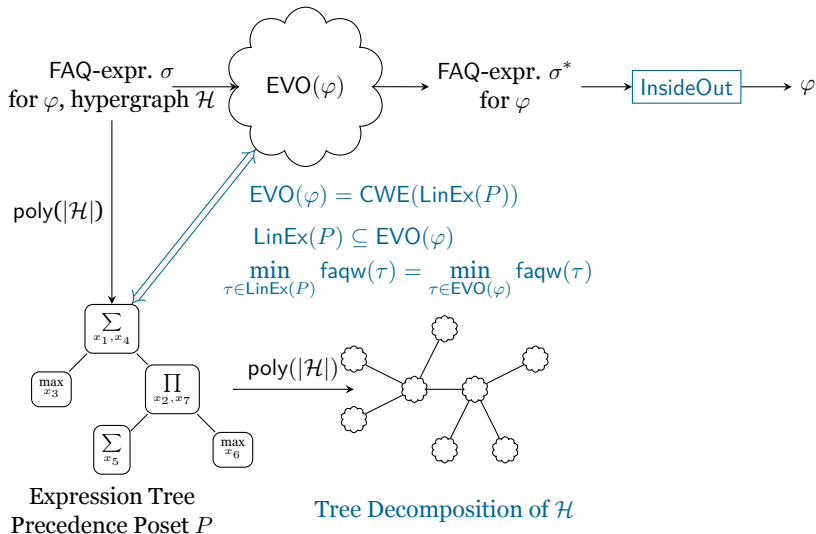
Technical Contributions



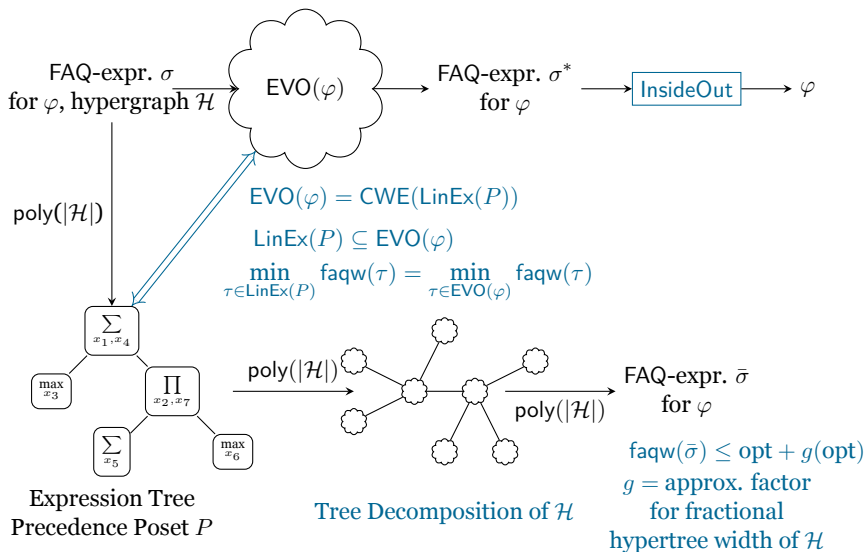
Technical Contributions



Technical Contributions

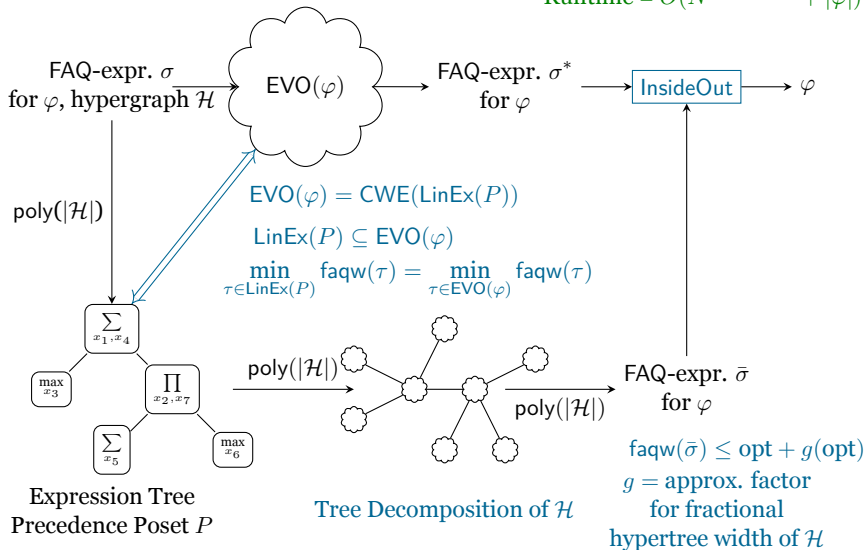


Technical Contributions



Technical Contributions

$$\text{Runtime} = \tilde{O}(N^{\text{opt}+g(\text{opt})} + |\varphi|)$$



Some Corollaries

Problem	Previous Algo.	InsideOut
#QCQ	No non-trivial algo	$\tilde{O}(N^{\text{faqw}(\varphi)} + \varphi)$
QCQ	$\tilde{O}(N^{\text{PW}(\mathcal{H})} + \varphi)$ Chen-Dalmau (LICS 2012)	$\tilde{O}(N^{\text{faqw}(\varphi)} + \varphi)$
#CQ	$\tilde{O}(N^{\text{DM}(\mathcal{H})} + \varphi)$ Durand-Mengel (ICDT 2013)	$\tilde{O}(N^{\text{faqw}(\varphi)} + \varphi)$
Joins	$\tilde{O}(N^{\text{ftw}(\mathcal{H})} + \varphi)$ Grohe-Marx (SODA'06)	$\tilde{O}(N^{\text{ftw}(\mathcal{H})} + \varphi)$
Marginal Distrib.	$\tilde{O}(N^{\text{htw}(\varphi)} + \varphi)$	$\tilde{O}(N^{\text{faqw}(\varphi)} + \varphi)$
MAP query	$\tilde{O}(N^{\text{htw}(\varphi)} + \varphi)$ Kask et al. (Artif. Intel. 2005)	$\tilde{O}(N^{\text{faqw}(\varphi)} + \varphi)$
Matrix Chain Mult.	DP bound	DP Bound
DFT	$O(N \log_p N)$	$O(N \log_p N)$
	Aji-McEliece (IEEE Trans. IT 2000) Dechter (Artif. Intell. 1999) Textbook	

Table of Contents

The FAQ problem

Summary of Results and Approach

The OutsideIn algorithm

The InsideOut algorithm

Choosing a variable ordering

How are factors accessed?

- ▶ A **factor oracle** for each input factor ψ_S

How are factors accessed?

- ▶ A **factor oracle** for each input factor ψ_S
 - ▶ Oracle = (complexity people's) fancy name for an API

How are factors accessed?

- ▶ A **factor oracle** for each input factor ψ_S
 - ▶ Oracle = (complexity people's) fancy name for an API
- ▶ Output factor: a table of tuples

$$\langle \mathbf{x}_F, \varphi[\mathbf{x}_F] \rangle$$

but only store tuples for which $\varphi[\mathbf{x}_F] \neq \mathbf{0}$

How are factors accessed?

- ▶ A **factor oracle** for each input factor ψ_S
 - ▶ Oracle = (complexity people's) fancy name for an API
- ▶ Output factor: a table of tuples

$$\langle \mathbf{x}_F, \varphi[\mathbf{x}_F] \rangle$$

but only store tuples for which $\varphi[\mathbf{x}_F] \neq \mathbf{0}$

- ▶ Called the **listing representation**

How are factors accessed?

- ▶ A **factor oracle** for each input factor ψ_S
 - ▶ Oracle = (complexity people's) fancy name for an API
- ▶ Output factor: a table of tuples

$$\langle \mathbf{x}_F, \varphi[\mathbf{x}_F] \rangle$$

but only store tuples for which $\varphi[\mathbf{x}_F] \neq \mathbf{0}$

- ▶ Called the **listing representation**

Example (Salary : {ID, Name} $\rightarrow \mathbb{R}_+$)

How are factors accessed?

- ▶ A **factor oracle** for each input factor ψ_S
 - ▶ Oracle = (complexity people's) fancy name for an API
- ▶ Output factor: a table of tuples

$$\langle \mathbf{x}_F, \varphi[\mathbf{x}_F] \rangle$$

but only store tuples for which $\varphi[\mathbf{x}_F] \neq \mathbf{0}$

- ▶ Called the **listing representation**

Example (Salary : {ID, Name} $\rightarrow \mathbb{R}_+$)

ID	Name	Salary
1234	AnHai Doan	\$400,000
5678	Current Hung Ngo	\$100,000
9999	Future Hung Ngo	\$1,000,000

Notation: conditional factor

- ▶ Let ψ_S be a factor
- ▶ Let $K \subseteq [n]$, $\mathbf{y}_K \in \prod_{i \in K} \text{Dom}(X_i)$
- ▶ **Conditional factor**

$$\psi_S(\cdot \mid \mathbf{y}_K) : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \mathbf{D}$$

where

$$\psi_S(\mathbf{x}_S \mid \mathbf{y}_K) = \begin{cases} \mathbf{0} & \text{if } S \cap K \neq \emptyset \text{ and } \mathbf{x}_{S \cap K} \neq \mathbf{y}_{S \cap K} \\ \psi_S(\mathbf{x}_S) & \text{otherwise.} \end{cases}$$

Notation: conditional factor

- ▶ Let ψ_S be a factor
- ▶ Let $K \subseteq [n]$, $\mathbf{y}_K \in \prod_{i \in K} \text{Dom}(X_i)$
- ▶ **Conditional factor**

$$\psi_S(\cdot \mid \mathbf{y}_K) : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \mathbf{D}$$

where

$$\psi_S(\mathbf{x}_S \mid \mathbf{y}_K) = \begin{cases} \mathbf{0} & \text{if } S \cap K \neq \emptyset \text{ and } \mathbf{x}_{S \cap K} \neq \mathbf{y}_{S \cap K} \\ \psi_S(\mathbf{x}_S) & \text{otherwise.} \end{cases}$$

- ▶ Salary(ID=9999 | Name = “Future Hung Ngo”) = \$1,000,000.

Notation: conditional factor

- ▶ Let ψ_S be a factor
- ▶ Let $K \subseteq [n]$, $\mathbf{y}_K \in \prod_{i \in K} \text{Dom}(X_i)$
- ▶ **Conditional factor**

$$\psi_S(\cdot \mid \mathbf{y}_K) : \prod_{i \in S} \text{Dom}(X_i) \rightarrow \mathbf{D}$$

where

$$\psi_S(\mathbf{x}_S \mid \mathbf{y}_K) = \begin{cases} \mathbf{0} & \text{if } S \cap K \neq \emptyset \text{ and } \mathbf{x}_{S \cap K} \neq \mathbf{y}_{S \cap K} \\ \psi_S(\mathbf{x}_S) & \text{otherwise.} \end{cases}$$

- ▶ Salary(ID=9999 | Name = “Future Hung Ngo”) = \$1,000,000.
- ▶ Salary(ID=9999 | Name = “Paris Koutris”) = \$0.

OutsideIn for FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

OutsideIn for FAQ-SS *without* free variables

$$\begin{aligned}\varphi &= \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \\ &= \bigoplus_{x_1 \in \text{Dom}(X_1)} \left(\bigoplus_{\mathbf{x}_{[n]-\{1\}}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S \mid x_1) \right)\end{aligned}$$

OutsideIn for FAQ-SS *without* free variables

Subproblem on $\mathcal{H} - \{1\}$

$$\begin{aligned}\varphi &= \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \\ &= \bigoplus_{x_1 \in \text{Dom}(X_1)} \left(\bigoplus_{\mathbf{x}_{[n]-\{1\}}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S \mid x_1) \right)\end{aligned}$$

OutsiderIn for FAQ-SS *without* free variables

Subproblem on $\mathcal{H} - \{1\}$

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1 \in \text{Dom}(X_1)} \left(\bigoplus_{\mathbf{x}_{[n]-\{1\}}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S \mid x_1) \right)$$

Runtime $O(D^n)$ ← $x_1 \in \text{Dom}(X_1)$

OutsiderIn for FAQ-SS *without* free variables

Subproblem on $\mathcal{H} - \{1\}$

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1 \in \text{Dom}(X_1)} \left(\bigoplus_{\mathbf{x}_{[n]-\{1\}}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S \mid x_1) \right)$$

Runtime $O(D^n)$

$$= \bigoplus_{x_1 \in I_1} \left(\bigoplus_{\mathbf{x}_{[n]-\{1\}}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S \mid x_1) \right).$$

OutsideIn for FAQ-SS *without* free variables

Subproblem on $\mathcal{H} - \{1\}$

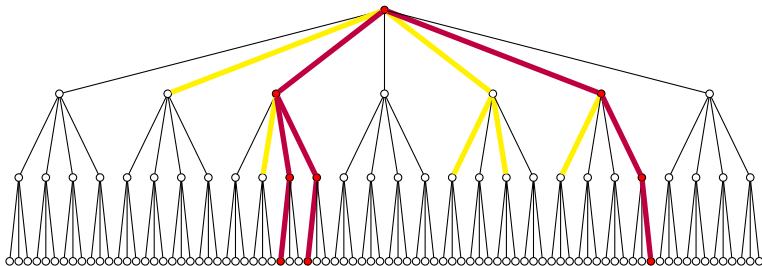
$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1 \in \text{Dom}(X_1)} \left(\bigoplus_{\mathbf{x}_{[n]-\{1\}}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S | x_1) \right)$$

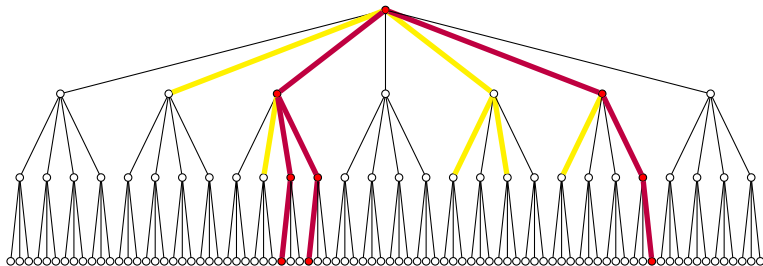
$$= \bigoplus_{x_1 \in I_1} \left(\bigoplus_{\mathbf{x}_{[n]-\{1\}}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S | x_1) \right).$$

$$I_1 = \{x_1 \mid \psi_S(\cdot | x_1) \not\equiv \mathbf{0}, \forall S \in \mathcal{E}\}$$

OutsideIn = Backtracking Search

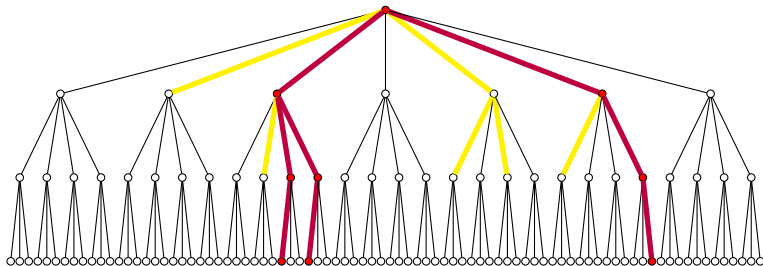


OutsideIn = Backtracking Search



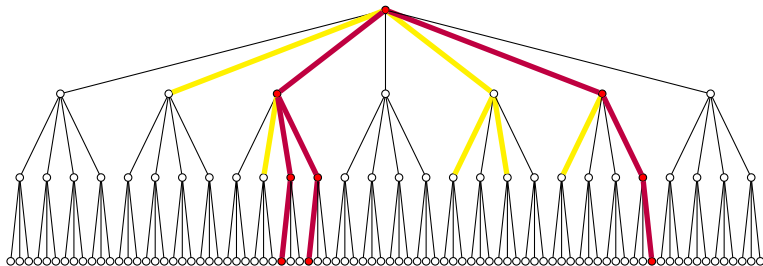
- **Question:** How good is OutsideIn?

Outsideln = Backtracking Search



- ▶ **Question:** How good is Outsideln?
- ▶ **Answer:** Worst-case optimal for joins!

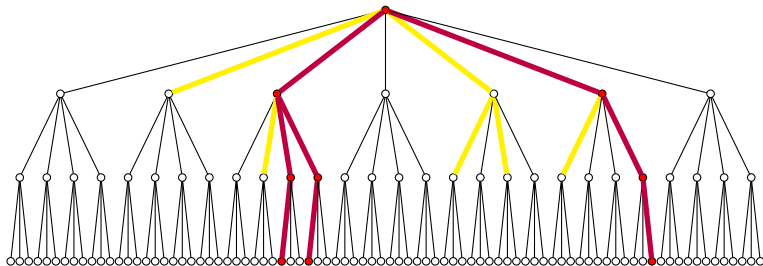
Outsideln = Backtracking Search



- ▶ **Question:** How good is Outsideln?
- ▶ **Answer:** Worst-case optimal for joins!
 - ▶ Grohe, Marx (SODA'06)
 - ▶ Atserias, Grohe, Marx (FOCS'08)

AGM-bound

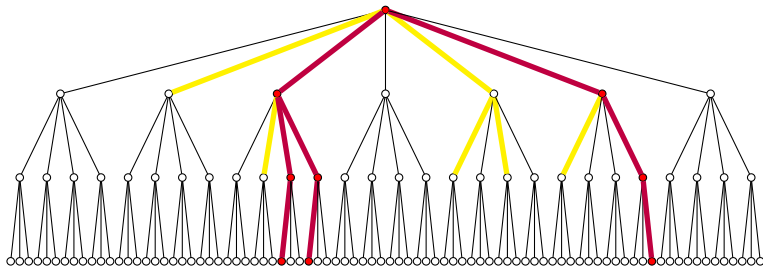
OutsideIn = Backtracking Search



- ▶ **Question:** How good is OutsideIn?
- ▶ **Answer:** Worst-case optimal for joins!
 - ▶ Grohe, Marx (SODA'06)
 - ▶ Atserias, Grohe, Marx (FOCS'08)
 - ▶ Ngo, Ré, Porat, Rudra (PODS'12)

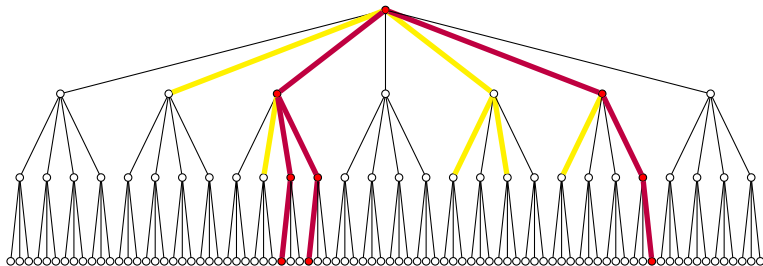
AGM-bound
NPRR algorithm

Outsideln = Backtracking Search



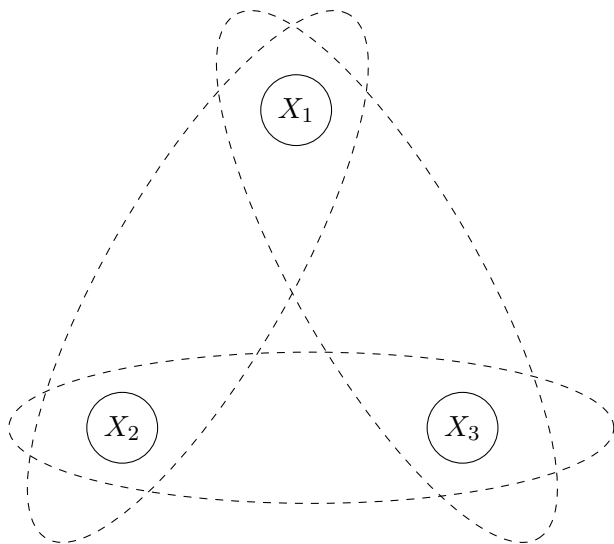
- ▶ **Question:** How good is Outsiden?
- ▶ **Answer:** Worst-case optimal for joins!
 - ▶ Grohe, Marx (SODA'06)
 - ▶ Atserias, Grohe, Marx (FOCS'08) AGM-bound
 - ▶ Ngo, Ré, Porat, Rudra (PODS'12) NPRR algorithm
 - ▶ Veldhuizen (ICDT'14) Leapfrog-Triejoin algorithm

Outsideln = Backtracking Search

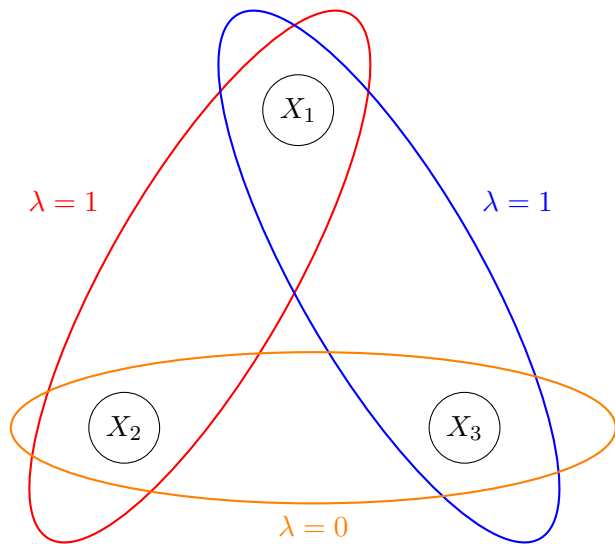


- ▶ **Question:** How good is Outsideln?
- ▶ **Answer:** Worst-case optimal for joins!
 - ▶ Grohe, Marx (SODA'06)
 - ▶ Atserias, Grohe, Marx (FOCS'08) AGM-bound
 - ▶ Ngo, Ré, Porat, Rudra (PODS'12) NPRR algorithm
 - ▶ Veldhuizen (ICDT'14) Leapfrog-Triejoin algorithm
 - ▶ Ngo, Ré, Rudra (SIGMOD Records'13) Generic-Join

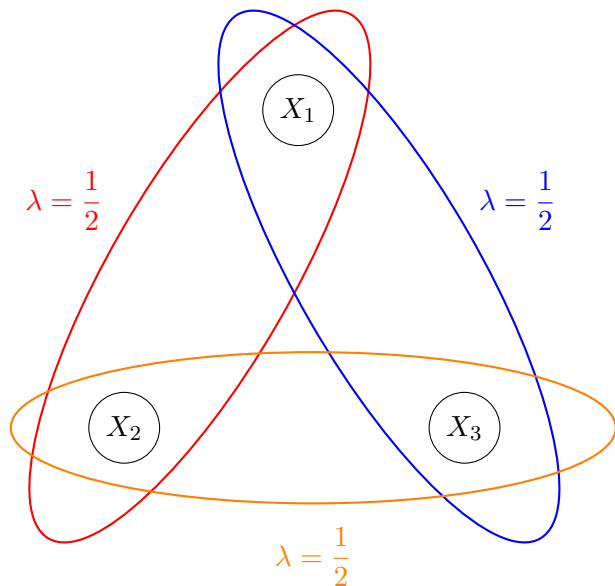
Background: fractional edge cover of \mathcal{H}



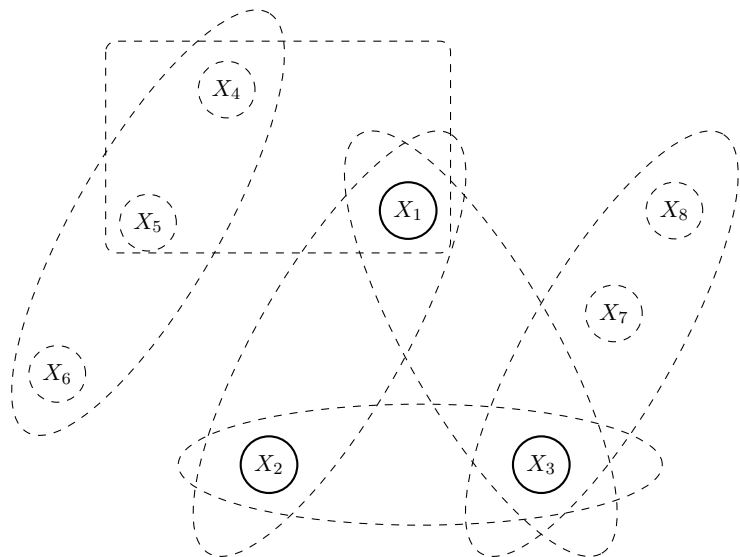
Background: fractional edge cover of \mathcal{H}



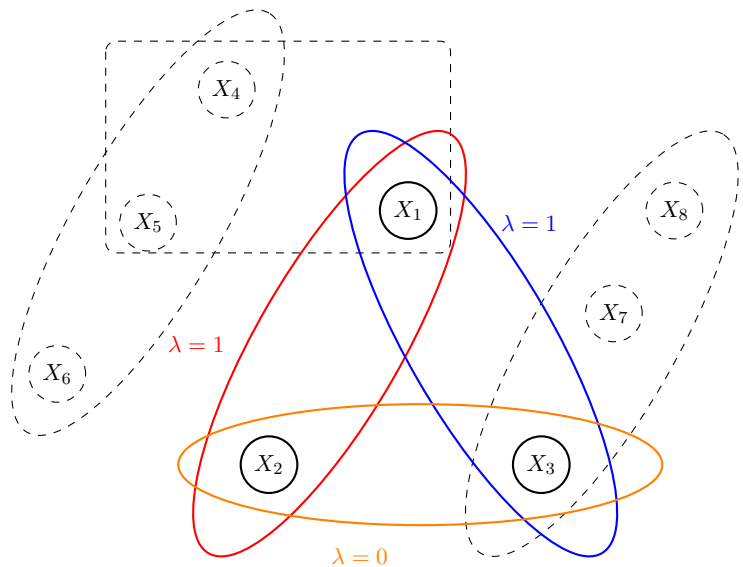
Background: fractional edge cover of \mathcal{H}



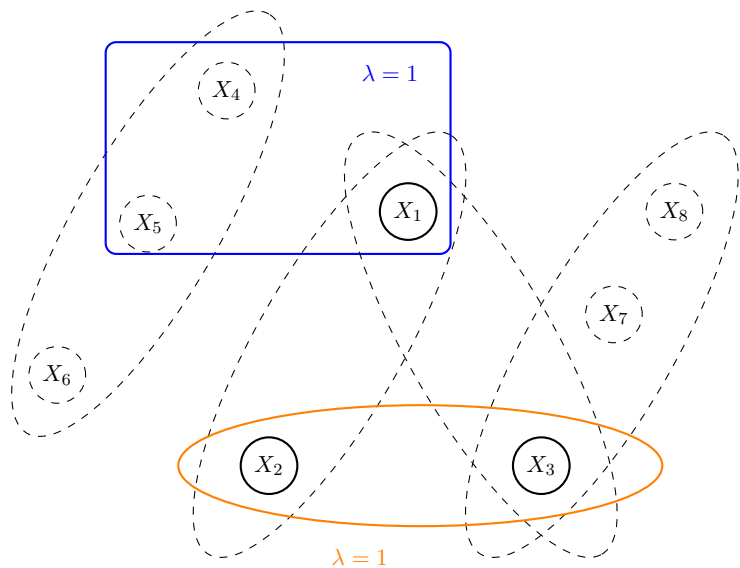
Background: fractional edge cover of $B = \{1, 2, 3\}$



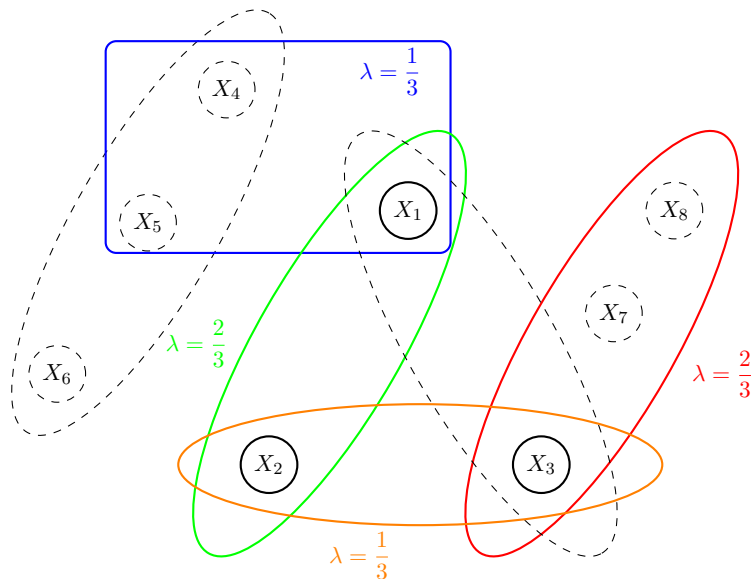
Background: fractional edge cover of $B = \{1, 2, 3\}$



Background: fractional edge cover of $B = \{1, 2, 3\}$



Background: fractional edge cover of $B = \{1, 2, 3\}$



Background: fractional edge cover

- ▶ $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ a multi hypergraph
- ▶ $B \subseteq \mathcal{V}$
- ▶ a **fractional edge cover of B** is a vector $(\lambda_S)_{S \in \mathcal{E}}$, s.t.

$$\sum_{S: v \in S} \lambda_S \geq 1, \quad \forall v \in B$$
$$\lambda_S \geq 0, \quad \forall S \in \mathcal{E}.$$

- ▶ called **fractional edge cover of \mathcal{H}** when $B = \mathcal{V}$

Pros of Outsideln

Theorem (Runtime of Outsideln)

Let $(\lambda_S)_{S \in \mathcal{E}}$ be **any** fractional edge cover of \mathcal{H} , then Outsideln runs in time

$$\tilde{O} \left(mn \prod_{S \in \mathcal{E}} |\psi_S|^{\lambda_S} \right)$$

Pros of Outsideln

Theorem (Runtime of Outsideln)

Let $(\lambda_S)_{S \in \mathcal{E}}$ be **any** fractional edge cover of \mathcal{H} , then Outsideln runs in time

$$\tilde{O} \left(mn \prod_{S \in \mathcal{E}} |\psi_S|^{\lambda_S} \right) \text{ AGM-bound}$$

Pros of Outsideln

Theorem (Runtime of Outsideln)

Let $(\lambda_S)_{S \in \mathcal{E}}$ be **any** fractional edge cover of \mathcal{H} , then Outsideln runs in time

$$\tilde{O} \left(mn \prod_{S \in \mathcal{E}} |\psi_S|^{\lambda_S} \right) \text{ AGM-bound}$$

- ▶ Outsideln is worst-case optimal for joins
- ▶ Very simple to implement

Pros of Outsideln

Theorem (Runtime of Outsideln)

Let $(\lambda_S)_{S \in \mathcal{E}}$ be **any** fractional edge cover of \mathcal{H} , then Outsideln runs in time

$$\tilde{O} \left(mn \prod_{S \in \mathcal{E}} |\psi_S|^{\lambda_S} \right) \text{ AGM-bound}$$

- ▶ Outsideln is worst-case optimal for joins
- ▶ Very simple to implement
- ▶ Works for FAQ-SS (i.e. Sum-Prod) *with* free variables!

Pros of Outsideln

Theorem (Runtime of Outsideln)

Let $(\lambda_S)_{S \in \mathcal{E}}$ be **any** fractional edge cover of \mathcal{H} , then Outsideln runs in time

$$\tilde{O} \left(mn \prod_{S \in \mathcal{E}} |\psi_S|^{\lambda_S} \right) \text{ AGM-bound}$$

- ▶ Outsideln is worst-case optimal for joins
- ▶ Very simple to implement
- ▶ Works for FAQ-SS (i.e. Sum-Prod) *with* free variables!
- ▶ Works well for dense queries (such as cliques)

Pros of Outsideln

Theorem (Runtime of Outsideln)

Let $(\lambda_S)_{S \in \mathcal{E}}$ be **any** fractional edge cover of \mathcal{H} , then Outsideln runs in time

$$\tilde{O} \left(mn \prod_{S \in \mathcal{E}} |\psi_S|^{\lambda_S} \right) \text{ AGM-bound}$$

- ▶ Outsideln is worst-case optimal for joins
- ▶ Very simple to implement
- ▶ Works for FAQ-SS (i.e. Sum-Prod) *with* free variables!
- ▶ Works well for dense queries (such as cliques)
- ▶ Works with any variable ordering

Pros of Outsideln

Theorem (Runtime of Outsideln)

Let $(\lambda_S)_{S \in \mathcal{E}}$ be **any** fractional edge cover of \mathcal{H} , then Outsideln runs in time

$$\tilde{O} \left(mn \prod_{S \in \mathcal{E}} |\psi_S|^{\lambda_S} \right) \text{ AGM-bound}$$

- ▶ Outsideln is worst-case optimal for joins
- ▶ Very simple to implement
- ▶ Works for FAQ-SS (i.e. Sum-Prod) *with* free variables!
- ▶ Works well for dense queries (such as cliques)
- ▶ Works with any variable ordering
- ▶ **Virtually zero memory footprint (backtracking search!).**

Examples of AGM-bound

- ▶ $Q = R(A, B) \bowtie S(B, C) \bowtie T(A, C), |R| = |S| = |T| = N$

—

Examples of AGM-bound

- ▶ $Q = R(A, B) \bowtie S(B, C) \bowtie T(A, C)$, $|R| = |S| = |T| = N$
- ▶ OutsideIn runs in time $\tilde{O}(N^{3/2})$ for any input

—

Examples of AGM-bound

- ▶ $Q = R(A, B) \bowtie S(B, C) \bowtie T(A, C)$, $|R| = |S| = |T| = N$
- ▶ OutsideIn runs in time $\tilde{O}(N^{3/2})$ for any input
 - ▶ $N^{3/2}$ is the AGM-bound $\text{AGM}(\mathcal{H})$

—

Examples of AGM-bound

- ▶ $Q = R(A, B) \bowtie S(B, C) \bowtie T(A, C)$, $|R| = |S| = |T| = N$
- ▶ OutsideIn runs in time $\tilde{O}(N^{3/2})$ for any input
 - ▶ $N^{3/2}$ is the AGM-bound $\text{AGM}(\mathcal{H})$
 - ▶ $3/2$ is the *fractional edge cover number* $\rho_{\mathcal{H}}^*$

—

Examples of AGM-bound

- ▶ $Q = R(A, B) \bowtie S(B, C) \bowtie T(A, C)$, $|R| = |S| = |T| = N$
- ▶ OutsideIn runs in time $\tilde{O}(N^{3/2})$ for any input
 - ▶ $N^{3/2}$ is the AGM-bound $\text{AGM}(\mathcal{H})$
 - ▶ $3/2$ is the *fractional edge cover number* $\rho_{\mathcal{H}}^*$
- ▶ **Any** join-project plan requires $\Omega(N^2)$ for some input

—

Examples of AGM-bound

- ▶ $Q = R(A, B) \bowtie S(B, C) \bowtie T(A, C)$, $|R| = |S| = |T| = N$
- ▶ OutsideIn runs in time $\tilde{O}(N^{3/2})$ for any input
 - ▶ $N^{3/2}$ is the AGM-bound $\text{AGM}(\mathcal{H})$
 - ▶ $3/2$ is the *fractional edge cover number* $\rho_{\mathcal{H}}^*$
- ▶ **Any** join-project plan requires $\Omega(N^2)$ for some input

—

- ▶ $Q = R(A, B, C) \bowtie S(A, B, D) \bowtie T(A, C, D) \bowtie W(B, C, D)$

Examples of AGM-bound

- ▶ $Q = R(A, B) \bowtie S(B, C) \bowtie T(A, C)$, $|R| = |S| = |T| = N$
- ▶ OutsideIn runs in time $\tilde{O}(N^{3/2})$ for any input
 - ▶ $N^{3/2}$ is the AGM-bound $\text{AGM}(\mathcal{H})$
 - ▶ $3/2$ is the *fractional edge cover number* $\rho_{\mathcal{H}}^*$
- ▶ **Any** join-project plan requires $\Omega(N^2)$ for some input

—

- ▶ $Q = R(A, B, C) \bowtie S(A, B, D) \bowtie T(A, C, D) \bowtie W(B, C, D)$
- ▶ OutsideIn runs in time $\tilde{O}(N^{4/3})$ for any input

Examples of AGM-bound

- ▶ $Q = R(A, B) \bowtie S(B, C) \bowtie T(A, C)$, $|R| = |S| = |T| = N$
- ▶ OutsideIn runs in time $\tilde{O}(N^{3/2})$ for any input
 - ▶ $N^{3/2}$ is the AGM-bound $\text{AGM}(\mathcal{H})$
 - ▶ $3/2$ is the *fractional edge cover number* $\rho_{\mathcal{H}}^*$
- ▶ **Any** join-project plan requires $\Omega(N^2)$ for some input

—

- ▶ $Q = R(A, B, C) \bowtie S(A, B, D) \bowtie T(A, C, D) \bowtie W(B, C, D)$
- ▶ OutsideIn runs in time $\tilde{O}(N^{4/3})$ for any input
 - ▶ $N^{4/3}$ is the AGM-bound $\text{AGM}(\mathcal{H})$

Examples of AGM-bound

- ▶ $Q = R(A, B) \bowtie S(B, C) \bowtie T(A, C)$, $|R| = |S| = |T| = N$
- ▶ OutsideIn runs in time $\tilde{O}(N^{3/2})$ for any input
 - ▶ $N^{3/2}$ is the AGM-bound $\text{AGM}(\mathcal{H})$
 - ▶ $3/2$ is the *fractional edge cover number* $\rho_{\mathcal{H}}^*$
- ▶ **Any** join-project plan requires $\Omega(N^2)$ for some input

—

- ▶ $Q = R(A, B, C) \bowtie S(A, B, D) \bowtie T(A, C, D) \bowtie W(B, C, D)$
- ▶ OutsideIn runs in time $\tilde{O}(N^{4/3})$ for any input
 - ▶ $N^{4/3}$ is the AGM-bound $\text{AGM}(\mathcal{H})$
 - ▶ $4/3$ is the *fractional edge cover number* $\rho_{\mathcal{H}}^*$

Examples of AGM-bound

- ▶ $Q = R(A, B) \bowtie S(B, C) \bowtie T(A, C)$, $|R| = |S| = |T| = N$
- ▶ OutsideIn runs in time $\tilde{O}(N^{3/2})$ for any input
 - ▶ $N^{3/2}$ is the AGM-bound $\text{AGM}(\mathcal{H})$
 - ▶ $3/2$ is the *fractional edge cover number* $\rho_{\mathcal{H}}^*$
- ▶ **Any** join-project plan requires $\Omega(N^2)$ for some input

—

- ▶ $Q = R(A, B, C) \bowtie S(A, B, D) \bowtie T(A, C, D) \bowtie W(B, C, D)$
- ▶ OutsideIn runs in time $\tilde{O}(N^{4/3})$ for any input
 - ▶ $N^{4/3}$ is the AGM-bound $\text{AGM}(\mathcal{H})$
 - ▶ $4/3$ is the *fractional edge cover number* $\rho_{\mathcal{H}}^*$
- ▶ **Any** join-project plan requires $\Omega(N^2)$ for some input

Cons of OutsideIn

- ▶ Horrible for large acyclic queries

Cons of OutsideIn

- ▶ Horrible for large acyclic queries
- ▶ Yannakakis algorithm runs in time

$$\tilde{O}(N + |\text{output}|)$$

on any (α -)acyclic query

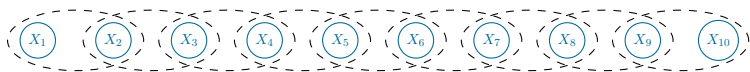
Cons of OutsideIn

- ▶ Horrible for large acyclic queries
- ▶ Yannakakis algorithm runs in time

$$\tilde{O}(N + |\text{output}|)$$

on any (α -)acyclic query

- ▶ On a $2k$ -path query, AGM-bound is $O(N^k)$.



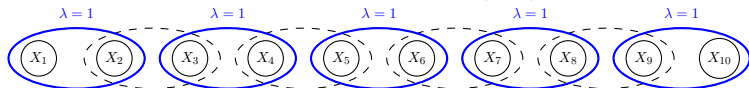
Cons of Outsiden

- ▶ Horrible for large acyclic queries
- ▶ Yannakakis algorithm runs in time

$$\tilde{O}(N + |\text{output}|)$$

on any (α -)acyclic query

- ▶ On a $2k$ -path query, AGM-bound is $O(N^k)$.



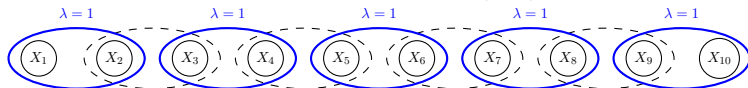
Cons of Outsiden

- ▶ Horrible for large acyclic queries
- ▶ Yannakakis algorithm runs in time

$$\tilde{O}(N + |\text{output}|)$$

on any (α -)acyclic query

- ▶ On a $2k$ -path query, AGM-bound is $O(N^k)$.



- ▶ More generally, horrible whenever $\text{fhtw} < \text{AGM-bound}$.

Table of Contents

The FAQ problem

Summary of Results and Approach

The OutsideIn algorithm

The InsideOut algorithm

Choosing a variable ordering

FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

FAQ-SS *without* free variables

$$\begin{aligned}\varphi &= \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \\ &= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)\end{aligned}$$

FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{n \in S} \psi_S(\mathbf{x}_S) \right)$$

FAQ-SS *without* free variables

$$\begin{aligned}\varphi &= \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \\ &= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right) \\ &= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \right) \left(\bigoplus_{x_n} \bigotimes_{n \in S} \psi_S(\mathbf{x}_S) \right)\end{aligned}$$

\uparrow
 \otimes distributes over \oplus

FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{n \in S} \psi_S(\mathbf{x}_S) \right)$$

FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

$$U_n = \bigcup_{S: n \in S} S$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \left(\bigoplus_{x_n} \bigotimes_{n \in S} \psi_S(\mathbf{x}_S) \right)$$

FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

$$U_n = \bigcup_{S: n \in S} S$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{n \in S} \psi_S(\mathbf{x}_S) \right)$$

$$\psi_{U_n - \{n\}}(\mathbf{x}_{U_n - \{n\}})$$

FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

$$U_n = \bigcup_{S: n \in S} S$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{n \in S} \psi_S(\mathbf{x}_S) \right)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \otimes \psi_{U_n - \{n\}}(\mathbf{x}_{U_n - \{n\}})$$

$$\psi_{U_n - \{n\}}(\mathbf{x}_{U_n - \{n\}})$$

FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

$$U_n = \bigcup_{S: n \in S} S$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{n \in S} \psi_S(\mathbf{x}_S) \right)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \otimes \psi_{U_n - \{n\}}(\mathbf{x}_{U_n - \{n\}})$$

FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

$$U_n = \bigcup_{S: n \in S} S$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{n \in S} \psi_S(\mathbf{x}_S) \right)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \otimes \psi_{U_n - \{n\}}(\mathbf{x}_{U_n - \{n\}})$$

FAQ-SS *without* free variables

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \left(\bigoplus_{x_n} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

$$U_n = \bigcup_{S: n \in S} S$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{n \in S} \psi_S(\mathbf{x}_S) \right)$$

$$= \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \otimes \psi_{U_n - \{n\}}(\mathbf{x}_{U_n - \{n\}})$$

New instance of FAQ-SS

$$\mathcal{H}_{n-1} = ([n-1], \mathcal{E}_{n-1})$$

$$\mathcal{E}_{n-1} = \mathcal{E} + (U_n - \{n\}) - \{S \mid n \in S\}$$

Naïve InsideOut = Variable Elimination

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

Naïve InsideOut = Variable Elimination

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

Naïve InsideOut = Variable Elimination

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

\uparrow
 $\psi_{U_n - \{n\}}(\mathbf{x}_{U_n - \{n\}})$

Naïve InsideOut = Variable Elimination

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

But how do we compute this?

$$\uparrow$$
$$\psi_{U_n - \{n\}}(\mathbf{x}_{U_n - \{n\}})$$

Naïve InsideOut = Variable Elimination

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

Run the OutsideIn algorithm!

$$\uparrow$$
$$\psi_{U_n - \{n\}}(\mathbf{x}_{U_n - \{n\}})$$

Naïve InsideOut = Variable Elimination

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

Run the OutsideIn algorithm!

$$\psi_{U_n - \{n\}}(\mathbf{x}_{U_n - \{n\}})$$

$$\text{Runtime} \approx |U_n| \cdot |\partial(n)| \cdot \prod_{S \in \partial(n)} |\psi_S|^{\lambda_S^{(n)}}$$

Naïve InsideOut = Variable Elimination

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

Theorem (Runtime of Naïve InsideOut)

For a given variable ordering v_1, \dots, v_n , the runtime of Naïve InsideOut makes is \tilde{O} of

$$\sum_{k=1}^n |U_k| \cdot |\partial(v_k)| \cdot \prod_{S \in \partial(v_k)} |\psi_S|^{\lambda_S^{(k)}} \leq mn \sum_{k=1}^n \prod_{S \in \partial(v_k)} |\psi_S|^{\lambda_S^{(k)}}$$

Naïve InsideOut = Variable Elimination

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

Theorem (Runtime of Naïve InsideOut)

For a given variable ordering v_1, \dots, v_n , the runtime of Naïve InsideOut makes is \tilde{O} of

$$\sum_{k=1}^n |U_k| \cdot |\partial(v_k)| \cdot \prod_{S \in \partial(v_k)} |\psi_S|^{\lambda_S^{(k)}} \leq mn \sum_{k=1}^n \prod_{S \in \partial(v_k)} |\psi_S|^{\lambda_S^{(k)}}$$

Corollary (Textbook result – Zhang-Poole’95, Dechter’96)

For a “good” variable ordering, PGM inference can be done in time $O(mn^2 D^{w+1})$ where $w = \text{tree-width}(\mathcal{H}) = \text{tw}(\mathcal{H})$.

Pros and Cons of Naïve InsideOut

- ▶ **The good:** $O(kN)$ -time for k -path query!

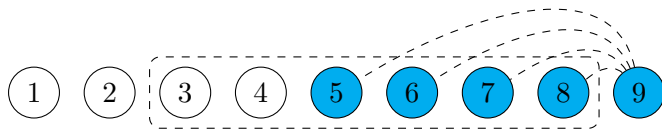
Pros and Cons of Naïve InsideOut

- ▶ **The good:** $O(kN)$ -time for k -path query!
- ▶ **The bad:** consider the following (part of a) graph



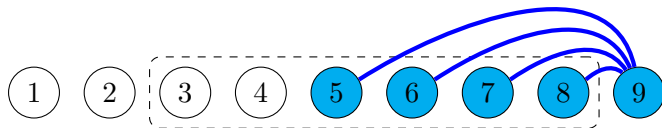
Pros and Cons of Naïve InsideOut

- ▶ **The good:** $O(kN)$ -time for k -path query!
- ▶ **The bad:** consider the following (part of a) graph



Pros and Cons of Naïve InsideOut

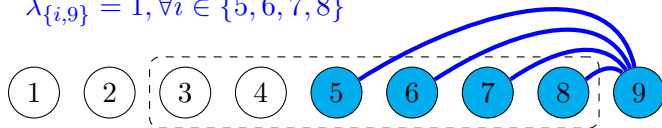
- ▶ **The good:** $O(kN)$ -time for k -path query!
- ▶ **The bad:** consider the following (part of a) graph



Pros and Cons of Naïve InsideOut

- ▶ **The good:** $O(kN)$ -time for k -path query!
- ▶ **The bad:** consider the following (part of a) graph

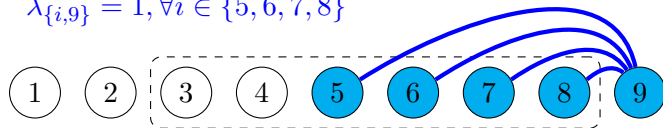
$$\lambda_{\{i,9\}} = 1, \forall i \in \{5, 6, 7, 8\}$$



Pros and Cons of Naïve InsideOut

- ▶ **The good:** $O(kN)$ -time for k -path query!
- ▶ **The bad:** consider the following (part of a) graph

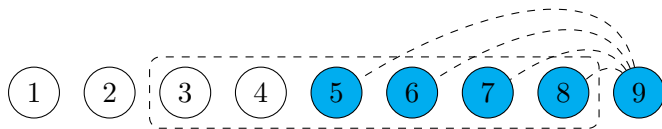
$$\lambda_{\{i,9\}} = 1, \forall i \in \{5, 6, 7, 8\}$$



- ▶ Say $|\psi_S| = N$, then runtime $O(N^4)$.

Pros and Cons of Naïve InsideOut

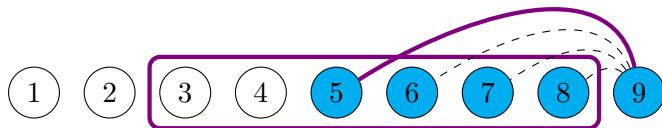
- ▶ **The good:** $O(kN)$ -time for k -path query!
- ▶ **The bad:** consider the following (part of a) graph



- ▶ Say $|\psi_S| = N$, then runtime $O(N^4)$.

Pros and Cons of Naïve InsideOut

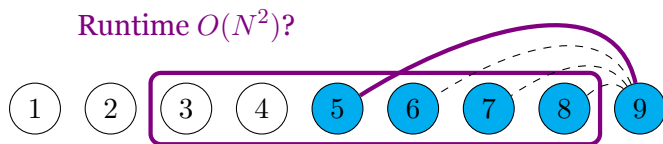
- ▶ **The good:** $O(kN)$ -time for k -path query!
- ▶ **The bad:** consider the following (part of a) graph



- ▶ Say $|\psi_S| = N$, then runtime $O(N^4)$.

Pros and Cons of Naïve InsideOut

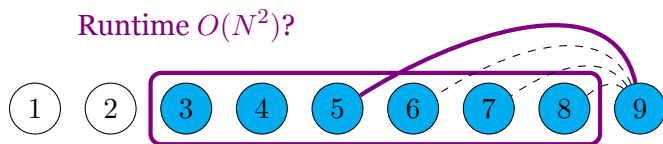
- ▶ **The good:** $O(kN)$ -time for k -path query!
- ▶ **The bad:** consider the following (part of a) graph



- ▶ Say $|\psi_S| = N$, then runtime $O(N^4)$.

Pros and Cons of Naïve InsideOut

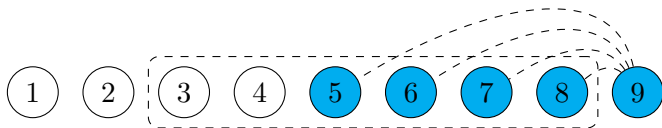
- ▶ **The good:** $O(kN)$ -time for k -path query!
- ▶ **The bad:** consider the following (part of a) graph



- ▶ Say $|\psi_S| = N$, then runtime $O(N^4)$.
- ▶ Good idea, but naïve application increases U_g

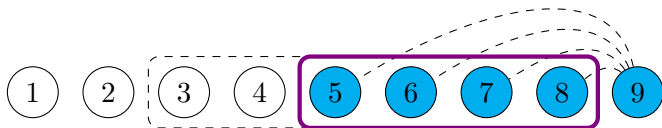
01-projection

- Projection onto U_9 :



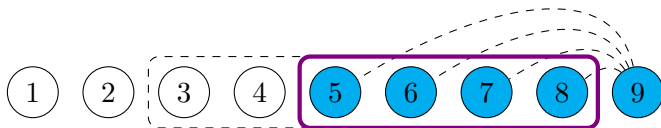
01-projection

- Projection onto U_9 :



01-projection

- ▶ Projection onto U_9 :



- ▶ Formally, $S \in \mathcal{E}, U \subseteq \mathcal{V}$ s.t. $S \cap U \neq \emptyset$,

$$\psi_{S|U} : \prod_{i \in S \cap U} \text{Dom}(X_i) \rightarrow \{\mathbf{0}, \mathbf{1}\},$$

where

$$\psi_{S|U}(\mathbf{x}_{S \cap U}) = \begin{cases} \mathbf{1} & \text{if } \exists \mathbf{y}_S \text{ s.t. } \psi_S(\mathbf{y}_S) \neq \mathbf{0} \text{ and } \mathbf{y}_{S \cap U} = \mathbf{x}_{S \cap U} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

InsideOut = VE + 01-projections

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

InsideOut = VE + 01-projections

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \left(\bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

InsideOut = VE + 01-projections

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \otimes \bigotimes_{\substack{S \notin \partial(n) \\ S \cap U_n \neq \emptyset}} \psi_{S|U_n}(\mathbf{x}_{S \cap U_n})$$

InsideOut = VE + 01-projections

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \bigotimes_{\substack{S \notin \partial(n) \\ S \cap U_n \neq \emptyset}} \psi_{S|U_n}(\mathbf{x}_{S \cap U_n})$$

$$\text{Time} \approx m|U_n| \cdot \prod_{S \cap U_n \neq \emptyset} |\psi_S|^{\lambda_S^{(k)}} = \text{AGM}(U_n)$$

InsideOut = VE + 01-projections

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \otimes \bigotimes_{\substack{S \notin \partial(n) \\ S \cap U_n \neq \emptyset}} \psi_{S|U_n}(\mathbf{x}_{S \cap U_n})$$

Theorem (Runtime of InsideOut)

For FAQ-SS, InsideOut runs in time \tilde{O} of

$$\sum_{k=1}^n |\{S \mid S \cap U_k \neq \emptyset\}| \cdot |U_k| \cdot \text{AGM}(U_k).$$

InsideOut = VE + 01-projections

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \bigotimes_{\substack{S \notin \partial(n) \\ S \cap U_n \neq \emptyset}} \psi_{S|U_n}(\mathbf{x}_S \cap U_n)$$

Theorem (Runtime of InsideOut)

For FAQ-SS, InsideOut runs in time \tilde{O} of

$$\sum_{k=1}^n |\{S \mid S \cap U_k \neq \emptyset\}| \cdot |U_k| \cdot \text{AGM}(U_k).$$

Corollary (New (textbook?) result)

For a “good” variable ordering v_1, \dots, v_n , PGM inference can be done in time $O(mn^2N^w)$ where $w = \text{fhtw}(\mathcal{H})$.

InsideOut = VE + 01-projections

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \otimes \bigotimes_{\substack{S \notin \partial(n) \\ S \cap U_n \neq \emptyset}} \psi_{S|U_n}(\mathbf{x}_{S \cap U_n})$$

Theorem (Runtime of InsideOut)

For FAQ-SS, InsideOut runs in time \tilde{O} of

$$\sum_{k=1}^n |\{S \mid S \cap U_k \neq \emptyset\}| \cdot |U_k| \cdot \text{AGM}(U_k).$$

Corollary (Grohe-Marx (SODA'06))

Join can be computed in time $\tilde{O}(N^{\text{fhtw}(\mathcal{H})} + |\text{output}|)$.

InsideOut = VE + 01-projections

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \bigotimes_{\substack{S \notin \partial(n) \\ S \cap U_n \neq \emptyset}} \psi_{S|U_n}(\mathbf{x}_{S \cap U_n})$$

Theorem (Runtime of InsideOut)

For FAQ-SS, InsideOut runs in time \tilde{O} of

$$\sum_{k=1}^n |\{S \mid S \cap U_k \neq \emptyset\}| \cdot |U_k| \cdot \text{AGM}(U_k).$$

Corollary (Grohe-Marx (SODA'06))

CSP on instances with bounded fhtw are fixed-parameter tractable.

InsideOut = VE + 01-projections

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \otimes \bigotimes_{\substack{S \notin \partial(n) \\ S \cap U_n \neq \emptyset}} \psi_{S|U_n}(\mathbf{x}_{S \cap U_n})$$

Theorem (Runtime of InsideOut)

For FAQ-SS, InsideOut runs in time \tilde{O} of

$$\sum_{k=1}^n |\{S \mid S \cap U_k \neq \emptyset\}| \cdot |U_k| \cdot \text{AGM}(U_k).$$

Corollary (New?)

Join cardinality can be computed in time $\tilde{O}(N^{\text{fhtw}(\mathcal{H})})$

InsideOut = VE + 01-projections

$$\varphi = \bigoplus_{x_1} \cdots \bigoplus_{x_{n-1}} \bigotimes_{n \notin S} \psi_S(\mathbf{x}_S) \bigoplus_{x_n} \bigotimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \otimes \bigotimes_{\substack{S \notin \partial(n) \\ S \cap U_n \neq \emptyset}} \psi_{S|U_n}(\mathbf{x}_{S \cap U_n})$$

Theorem (Runtime of InsideOut)

For FAQ-SS, InsideOut runs in time \tilde{O} of

$$\sum_{k=1}^n |\{S \mid S \cap U_k \neq \emptyset\}| \cdot |U_k| \cdot \text{AGM}(U_k).$$

Corollary (Pichler-Skritek 2011, Durand-Mengel (ICDT'2013))

For query graphs \mathcal{H} with bounded fhtw, quantifier-free #CQ is polynomial-time solvable.

InsideOut for general FAQ

$$\varphi = \cdots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

InsideOut for general FAQ

$$\begin{aligned}\varphi &= \cdots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \\ &= \cdots \bigoplus_{x_{n-1}}^{(n-1)} \left(\bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)\end{aligned}$$

InsideOut for general FAQ

$$\varphi = \dots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \dots \bigoplus_{x_{n-1}}^{(n-1)} \left(\bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

InsideOut for general FAQ

$$\varphi = \dots \oplus_{x_n}^{(n)} \otimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \dots \oplus_{x_{n-1}}^{(n-1)} \left(\oplus_{x_n}^{(n)} \otimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

If $(\mathbf{D}, \oplus^{(n)}, \otimes)$ is a semiring

InsideOut for general FAQ

$$\varphi = \dots \oplus_{x_n}^{(n)} \otimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \dots \oplus_{x_{n-1}}^{(n-1)} \left(\oplus_{x_n}^{(n)} \otimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

If $(\mathbf{D}, \oplus^{(n)}, \otimes)$ is a semiring

$$\varphi = \dots \oplus_{x_{n-1}}^{(n-1)} \left(\otimes_{S \notin \partial(n)} \psi_S(\mathbf{x}_S) \right) \left(\oplus_{x_n}^{(n)} \otimes_{S \in \partial(n)} \psi_S(\mathbf{x}_S) \right)$$

InsideOut for general FAQ

$$\varphi = \cdots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \cdots \bigoplus_{x_{n-1}}^{(n-1)} \left(\bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

If $\bigoplus^{(n)} = \bigotimes$

InsideOut for general FAQ

$$\varphi = \cdots \oplus_{x_n}^{(n)} \otimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \cdots \oplus_{x_{n-1}}^{(n-1)} \left(\oplus_{x_n}^{(n)} \otimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

If $\oplus^{(n)} = \otimes$

$$\varphi = \cdots \oplus_{x_{n-1}}^{(n-1)} \left(\otimes_{S \in \mathcal{E}} \oplus_{x_n}^{(n)} \psi_S(\mathbf{x}_S) \right)$$

InsideOut for general FAQ

$$\varphi = \cdots \oplus_{x_n}^{(n)} \otimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \cdots \oplus_{x_{n-1}}^{(n-1)} \left(\oplus_{x_n}^{(n)} \otimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

If $\oplus^{(n)} = \otimes$

$$\varphi = \cdots \oplus_{x_{n-1}}^{(n-1)} \left(\otimes_{S \in \mathcal{E}} \oplus_{x_n}^{(n)} \psi_S(\mathbf{x}_S) \right)$$

$$= \cdots \oplus_{x_{n-1}}^{(n-1)} \left(\otimes_{S \notin \partial(n)} [\psi_S(\mathbf{x}_S)]^{|\text{Dom}(X_n)|} \otimes \left(\otimes_{S \in \partial(n)} \oplus_{x_n}^{(n)} \psi_S(\mathbf{x}_S) \right) \right)$$

InsideOut for general FAQ

$$\varphi = \cdots \oplus_{x_n}^{(n)} \otimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

$$= \cdots \oplus_{x_{n-1}}^{(n-1)} \left(\oplus_{x_n}^{(n)} \otimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S) \right)$$

If $\oplus^{(n)} = \otimes$ and $\oplus^{(n)}$ is not-idempotent

$$\varphi = \cdots \oplus_{x_{n-1}}^{(n-1)} \left(\otimes_{S \in \mathcal{E}} \oplus_{x_n}^{(n)} \psi_S(\mathbf{x}_S) \right)$$

$$= \cdots \oplus_{x_{n-1}}^{(n-1)} \left(\otimes_{S \notin \partial(n)} [\psi_S(\mathbf{x}_S)]^{|\text{Dom}(X_n)|} \otimes \left(\otimes_{S \in \partial(n)} \oplus_{x_n}^{(n)} \psi_S(\mathbf{x}_S) \right) \right)$$

Table of Contents

The FAQ problem

Summary of Results and Approach

The OutsideIn algorithm

The InsideOut algorithm

Choosing a variable ordering

FAQ-width of a variable ordering

Time \approx incarnations of **OutsideIn** + **output size** .

FAQ-width of a variable ordering

Time \approx incarnations of **OutsideIn** + **output size** .

$$\leq \tilde{O}(mn) \left[\sum_{k \in K} \text{AGM}(U_k) + |\varphi| \right]$$

FAQ-width of a variable ordering

Time \approx incarnations of **OutsideIn** + **output size** .

$$\leq \tilde{O}(mn) \left[\sum_{k \in K} \text{AGM}(U_k) + |\varphi| \right]$$

Dynamic Programming for
for Matrix Chain Multiplication
and The FFT

FAQ-width of a variable ordering

Time \approx incarnations of **OutsideIn** + **output size** .

$$\begin{aligned} &\leq \tilde{O}(mn) \left[\sum_{k \in K} \text{AGM}(U_k) + |\varphi| \right] \\ &\leq \tilde{O}(mn) \left[\sum_{k \in K} N^{\rho_{\mathcal{H}}^*}(U_k) + |\varphi| \right] \end{aligned}$$

FAQ-width of a variable ordering

Time \approx incarnations of **Outsideln** + **output size** .

$$\begin{aligned} &\leq \tilde{O}(mn) \left[\sum_{k \in K} \text{AGM}(U_k) + |\varphi| \right] \\ &\leq \tilde{O}(mn) \left[\sum_{k \in K} N^{\rho_{\mathcal{H}}^*(U_k)} + |\varphi| \right] \end{aligned}$$

Definition (FAQ-width of a variable ordering σ)

$$\text{faqw}(\sigma) := \max_{k \in K} \rho_{\mathcal{H}}^*(U_k^\sigma).$$

FAQ-width of a variable ordering

Time \approx incarnations of **Outsideln** + **output size** .

$$\begin{aligned} &\leq \tilde{O}(mn) \left[\sum_{k \in K} \text{AGM}(U_k) + |\varphi| \right] \\ &\leq \tilde{O}(mn) \left[\sum_{k \in K} N^{\rho_{\mathcal{H}}^*(U_k)} + |\varphi| \right] \\ &\leq \tilde{O}(mn) \left[nN^{\text{faqw}(\sigma)} + |\varphi| \right]. \end{aligned}$$

Definition (FAQ-width of a variable ordering σ)

$$\text{faqw}(\sigma) := \max_{k \in K} \rho_{\mathcal{H}}^*(U_k^\sigma).$$

FAQ-width of a query

Definition (φ -equivalent variable ordering)

Let φ be an FAQ-query. A φ -equivalent variable ordering is a vertex ordering $\sigma = (v_1, \dots, v_n)$ that satisfies:

FAQ-width of a query

Definition (φ -equivalent variable ordering)

Let φ be an FAQ-query. A φ -equivalent variable ordering is a vertex ordering $\sigma = (v_1, \dots, v_n)$ that satisfies:

- (a) The set $\{v_1, \dots, v_f\}$ is exactly $F = [f]$.

FAQ-width of a query

Definition (φ -equivalent variable ordering)

Let φ be an FAQ-query. A φ -equivalent variable ordering is a vertex ordering $\sigma = (v_1, \dots, v_n)$ that satisfies:

- (a) The set $\{v_1, \dots, v_f\}$ is exactly $F = [f]$.
- (b) The function φ' defined by

$$\varphi'(\mathbf{x}_F) := \bigoplus_{x_{v_{f+1}}}^{(v_{f+1})} \dots \bigoplus_{x_{v_n}}^{(v_n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

is identical to the function φ no matter what the input factors are.

FAQ-width of a query

Definition (φ -equivalent variable ordering)

Let φ be an FAQ-query. A φ -equivalent variable ordering is a vertex ordering $\sigma = (v_1, \dots, v_n)$ that satisfies:

- (a) The set $\{v_1, \dots, v_f\}$ is exactly $F = [f]$.
- (b) The function φ' defined by

$$\varphi'(\mathbf{x}_F) := \bigoplus_{x_{v_{f+1}}}^{(v_{f+1})} \dots \bigoplus_{x_{v_n}}^{(v_n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

is identical to the function φ no matter what the input factors are.

$\text{EVO}(\varphi)$ denotes the set of all φ -equivalent variable orderings.

FAQ-width of a query

Definition (φ -equivalent variable ordering)

Let φ be an FAQ-query. A φ -equivalent variable ordering is a vertex ordering $\sigma = (v_1, \dots, v_n)$ that satisfies:

- (a) The set $\{v_1, \dots, v_f\}$ is exactly $F = [f]$.
- (b) The function φ' defined by

$$\varphi'(\mathbf{x}_F) := \bigoplus_{x_{v_{f+1}}}^{(v_{f+1})} \dots \bigoplus_{x_{v_n}}^{(v_n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

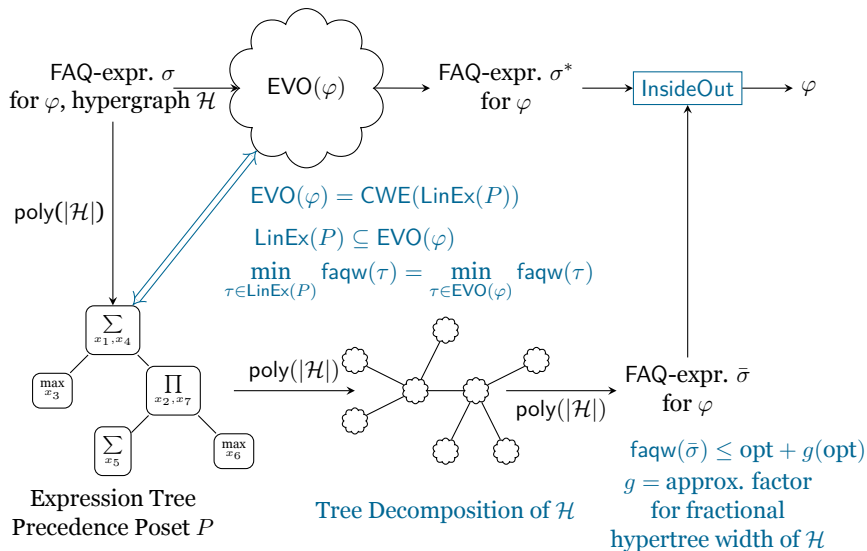
is identical to the function φ no matter what the input factors are.

$\text{EVO}(\varphi)$ denotes the set of all φ -equivalent variable orderings.

Definition (FAQ-width of an FAQ-query φ)

$$\text{faqw}(\varphi) := \min_{\sigma \in \text{EVO}(\varphi)} \text{faqw}(\sigma).$$

Roadmap Reminder



Characterizing EVO(φ)

We define a partially ordered set P on the variables of φ , called the **Precedence Poset**.

Characterizing $EVO(\varphi)$

We define a partially ordered set P on the variables of φ , called the **Precedence Poset**.

Soundness

$$\text{LinEx}(P) \subseteq EVO(\varphi).$$

Characterizing $\text{EVO}(\varphi)$

We define a partially ordered set P on the variables of φ , called the **Precedence Poset**.

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness Every $\sigma \in \text{EVO}(\varphi)$ has the same faqw as some $\sigma' \in \text{LinEx}(P)$.

Characterizing EVO(φ)

We define a partially ordered set P on the variables of φ , called the **Precedence Poset**.

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness Every $\sigma \in \text{EVO}(\varphi)$ has the same faqw as some $\sigma' \in \text{LinEx}(P)$.

$$\text{faqw}(\varphi) := \min_{\sigma \in \text{EVO}(\varphi)} \text{faqw}(\sigma).$$

Characterizing EVO(φ)

We define a partially ordered set P on the variables of φ , called the **Precedence Poset**.

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness Every $\sigma \in \text{EVO}(\varphi)$ has the same faqw as some $\sigma' \in \text{LinEx}(P)$.

$$\text{faqw}(\varphi) := \min_{\sigma \in \text{LinEx}(P)} \text{faqw}(\sigma).$$

FAQ with semiring variable aggregates

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \cdots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

FAQ with semiring variable aggregates

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \cdots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

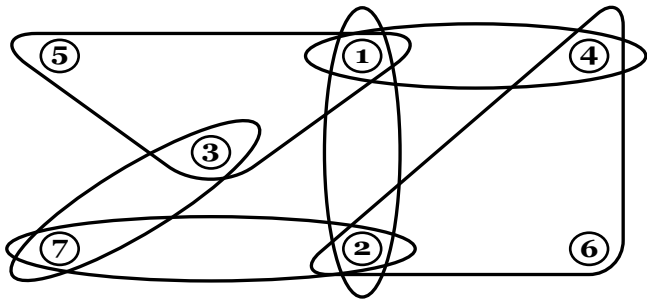
- ▶ **Expression Tree** defines the precedence poset

FAQ with semiring variable aggregates

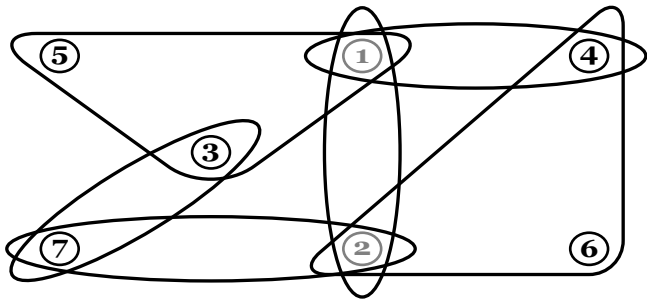
$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \cdots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

- ▶ **Expression Tree** defines the precedence poset
 - ▶ **Compartmentalization Step.**
 - ▶ **Compression Step.**

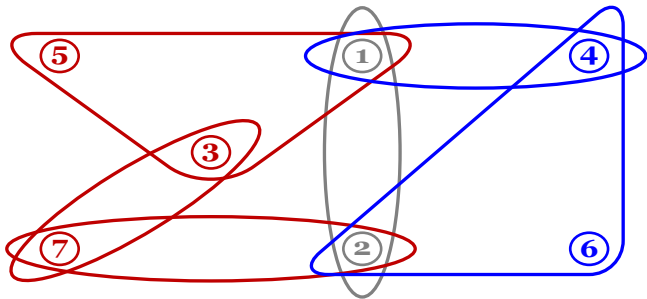
$$\varphi = \sum_{x_1} \sum_{x_2} \max_{x_3} \sum_{x_4} \sum_{x_5} \max_{x_6} \max_{x_7} \psi_{12} \psi_{135} \psi_{14} \psi_{246} \psi_{27} \psi_{37}$$



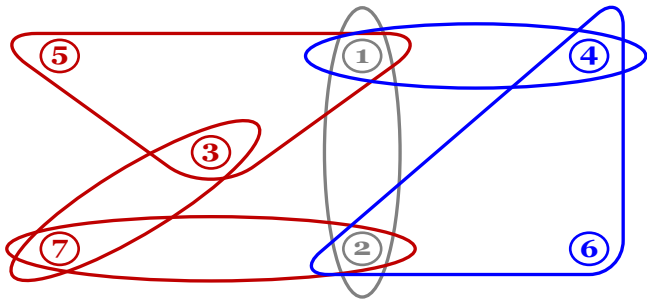
$$\varphi = \sum_{x_1} \sum_{x_2} \max_{x_3} \sum_{x_4} \sum_{x_5} \max_{x_6} \max_{x_7} \psi_{12} \psi_{135} \psi_{14} \psi_{246} \psi_{27} \psi_{37}$$



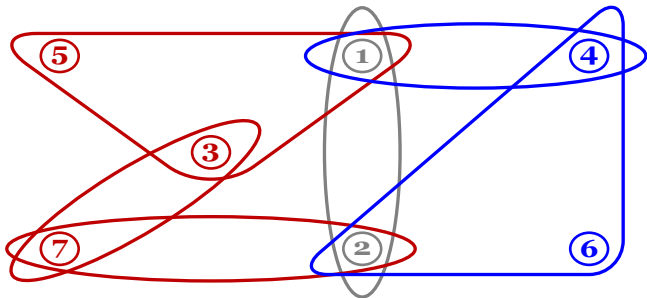
$$\varphi = \sum_{x_1} \sum_{x_2} \max_{x_3} \sum_{x_4} \sum_{x_5} \max_{x_6} \max_{x_7} \psi_{12} \psi_{135} \psi_{14} \psi_{246} \psi_{27} \psi_{37}$$



$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \max_{x_7} \psi_{135} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14} \psi_{246} \right)$$

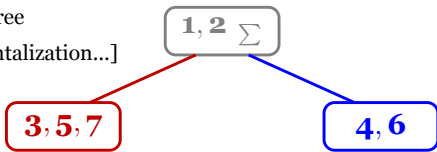


$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \max_{x_7} \psi_{135} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14} \psi_{246} \right)$$

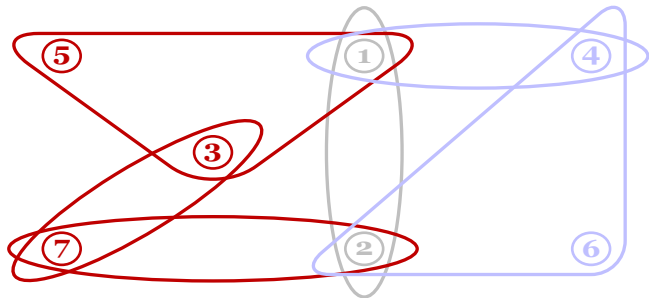


Expression Tree

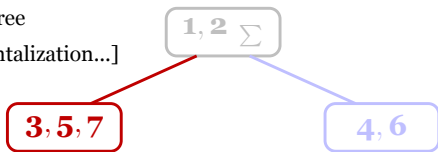
[Compartmentalization...]



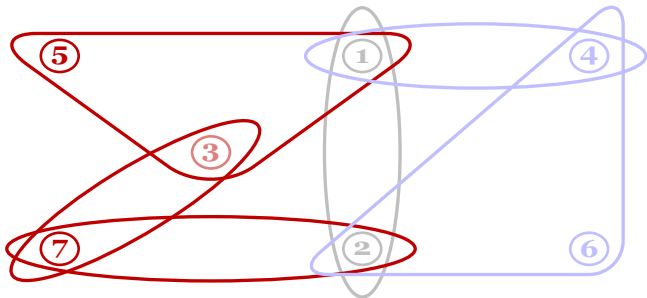
$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \max_{x_7} \psi_{135} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14} \psi_{246} \right)$$



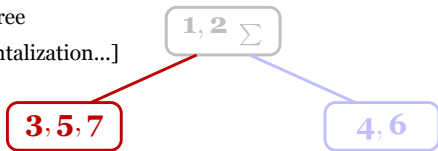
Expression Tree
[Compartmentalization...]



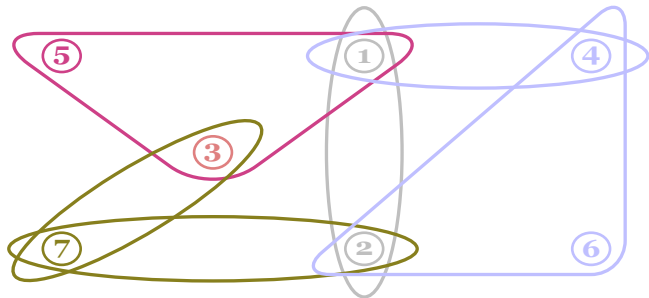
$$\phi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \max_{x_7} \psi_{135} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14} \psi_{246} \right)$$



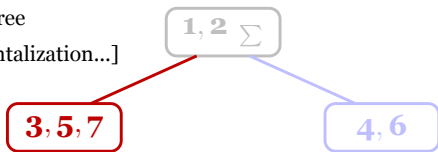
Expression Tree
[Compartmentalization...]



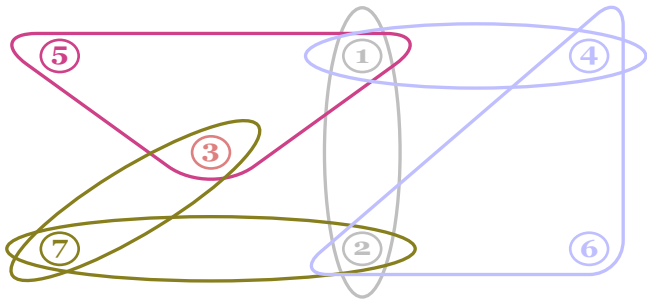
$$\phi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \max_{x_7} \psi_{135} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14} \psi_{246} \right)$$



Expression Tree
[Compartmentalization...]

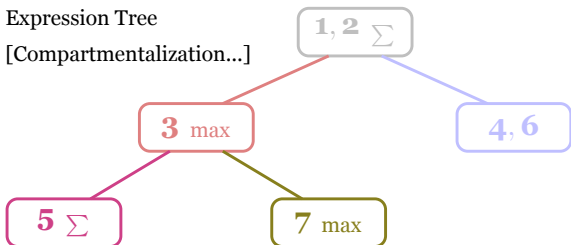


$$\phi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \psi_{135} \max_{x_7} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14} \psi_{246} \right)$$

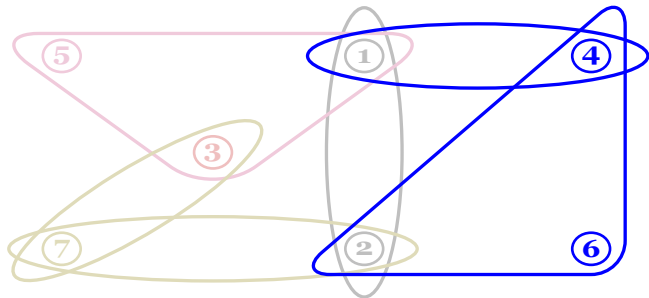


Expression Tree

[Compartmentalization...]

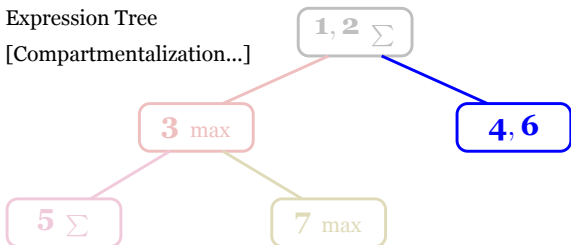


$$\phi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \psi_{135} \max_{x_7} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14} \psi_{246} \right)$$

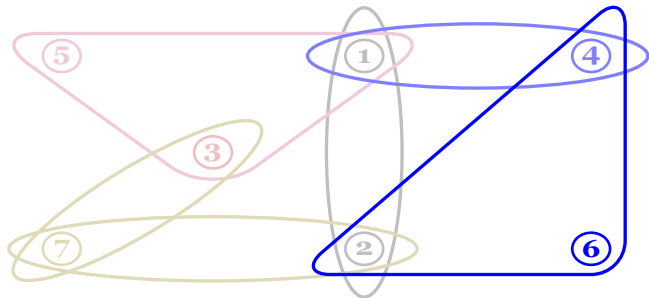


Expression Tree

[Compartmentalization...]

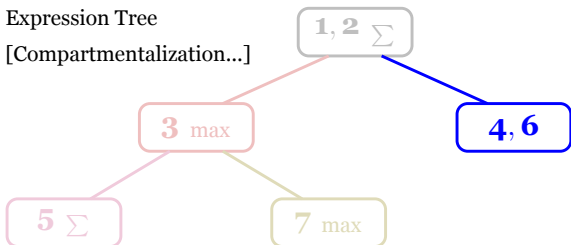


$$\phi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \psi_{135} \max_{x_7} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14} \psi_{246} \right)$$

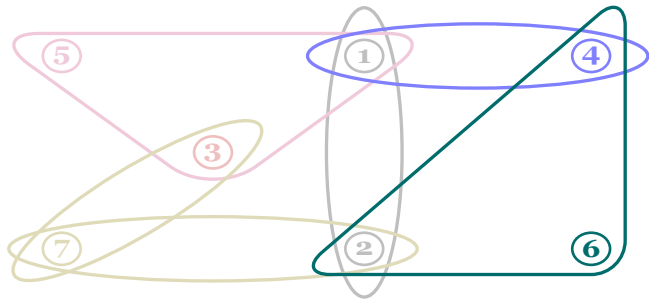


Expression Tree

[Compartmentalization...]

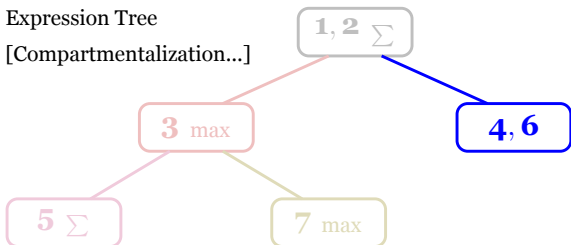


$$\phi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \psi_{135} \max_{x_7} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14} \psi_{246} \right)$$

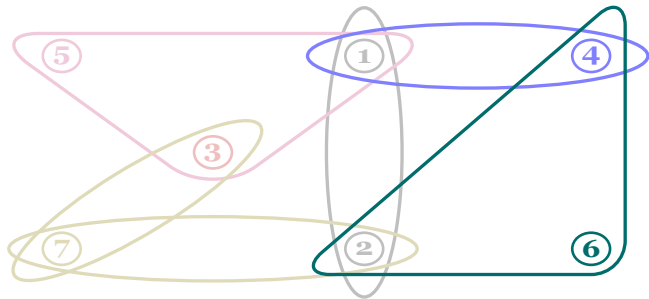


Expression Tree

[Compartmentalization...]

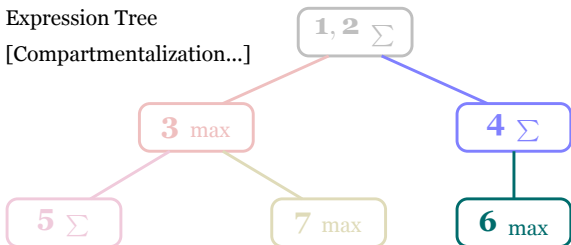


$$\phi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \psi_{135} \max_{x_7} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \psi_{14} \max_{x_6} \psi_{246} \right)$$

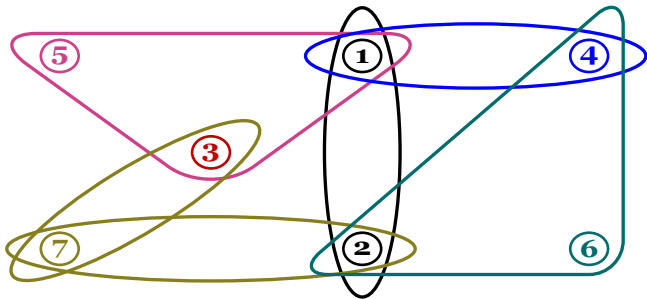


Expression Tree

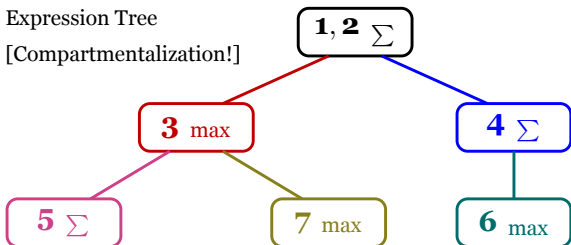
[Compartmentalization...]



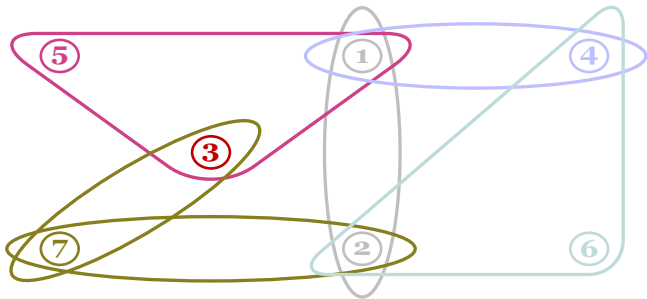
$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \psi_{135} \max_{x_7} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \psi_{14} \max_{x_6} \psi_{246} \right)$$



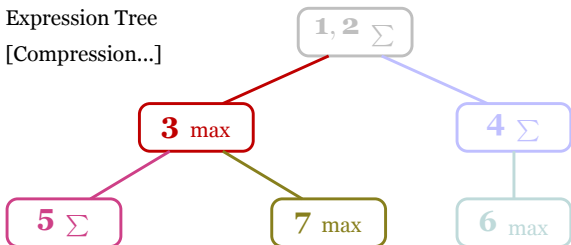
Expression Tree
[Compartmentalization!]



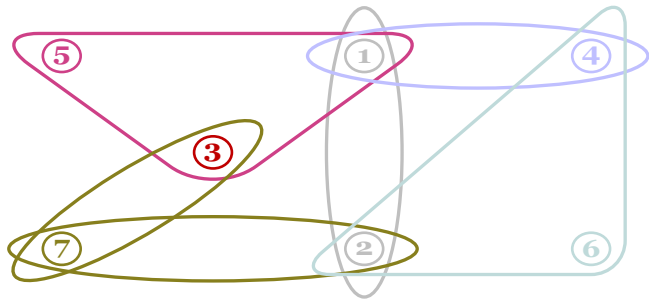
$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \psi_{135} \max_{x_7} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \psi_{14} \max_{x_6} \psi_{246} \right)$$



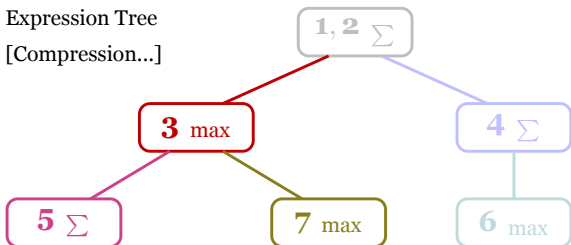
Expression Tree
[Compression...]



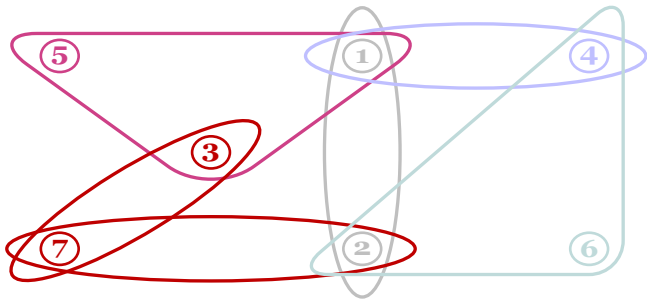
$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \max_{x_7} \psi_{27} \psi_{37} \sum_{x_5} \psi_{135} \right) \left(\sum_{x_4} \psi_{14} \max_{x_6} \psi_{246} \right)$$



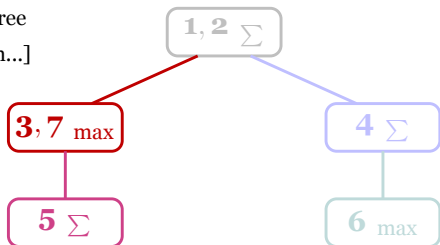
Expression Tree
[Compression...]



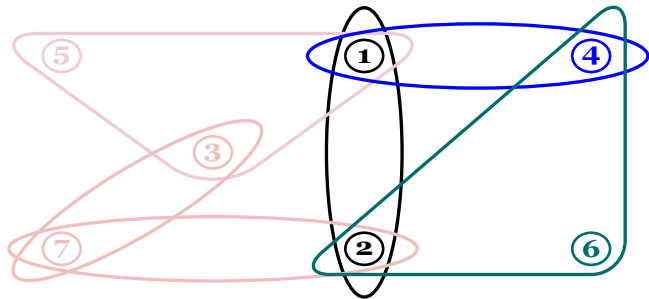
$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \max_{x_7} \psi_{27} \psi_{37} \sum_{x_5} \psi_{135} \right) \left(\sum_{x_4} \psi_{14} \max_{x_6} \psi_{246} \right)$$



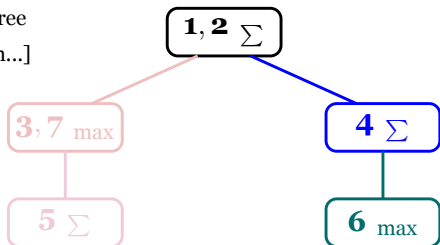
Expression Tree
[Compression...]



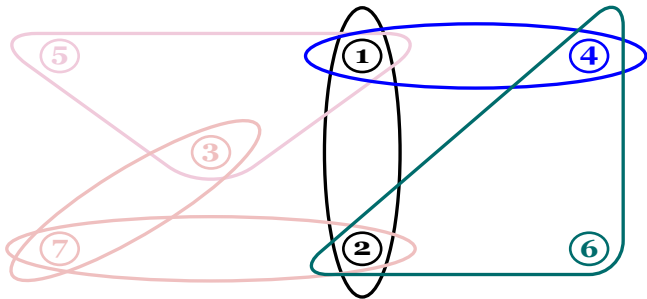
$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \max_{x_7} \psi_{27} \psi_{37} \sum_{x_5} \psi_{135} \right) \left(\sum_{x_4} \psi_{14} \max_{x_6} \psi_{246} \right)$$



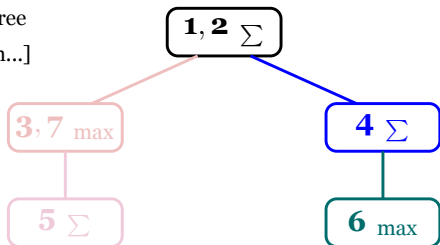
Expression Tree
[Compression...]



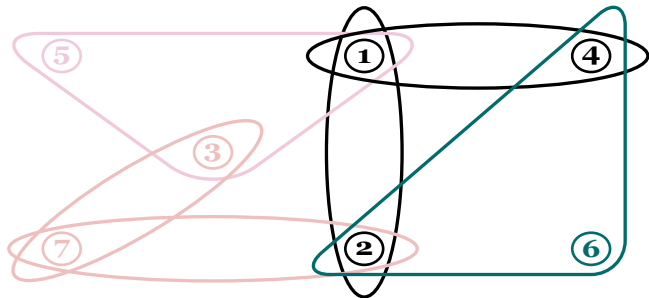
$$\varphi = \sum_{x_1} \sum_{x_2} \sum_{x_4} \psi_{12} \psi_{14} \left(\max_{x_3} \max_{x_7} \psi_{27} \psi_{37} \sum_{x_5} \psi_{135} \right) \left(\max_{x_6} \psi_{246} \right)$$



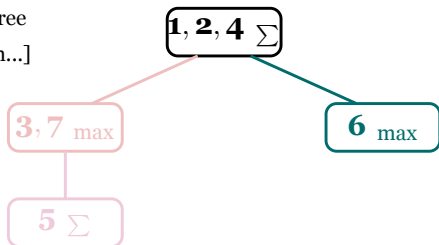
Expression Tree
[Compression...]



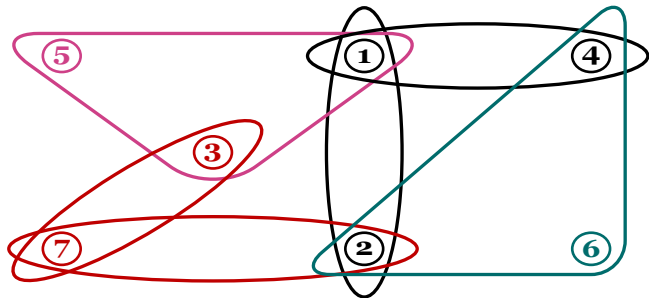
$$\varphi = \sum_{x_1} \sum_{x_2} \sum_{x_4} \psi_{12} \psi_{14} \left(\max_{x_3} \max_{x_7} \psi_{27} \psi_{37} \sum_{x_5} \psi_{135} \right) \left(\max_{x_6} \psi_{246} \right)$$



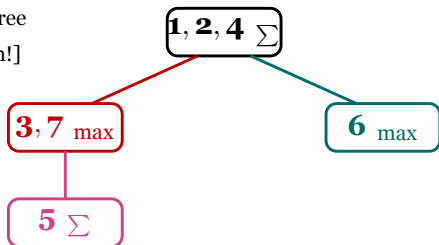
Expression Tree
[Compression...]



$$\varphi = \sum_{x_1} \sum_{x_2} \sum_{x_4} \psi_{12} \psi_{14} \left(\max_{x_3} \max_{x_7} \psi_{27} \psi_{37} \sum_{x_5} \psi_{135} \right) \left(\max_{x_6} \psi_{246} \right)$$



Expression Tree
[Compression!]



Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$g(\text{fhtw}(\mathcal{H})),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{faqw}(\varphi) + g(\text{faqw}(\varphi)).$$

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$g(\text{fhtw}(\mathcal{H})),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{faqw}(\varphi) + g(\text{faqw}(\varphi)).$$

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$g(\text{fhtw}(\mathcal{H})),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{OPT} + g(\text{OPT}).$$

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$g(\text{fhtw}(\mathcal{H})),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{OPT} + g(\text{OPT}).$$

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$g(\text{fhtw}(\mathcal{H})),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{OPT} + g(\text{OPT}).$$

Best known $g(x) = O(x^3)$ [Marx'12]

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$O(\text{fhtw}(\mathcal{H})^3),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{OPT} + O(\text{OPT}^3).$$

Best known $g(x) = O(x^3)$ [Marx'12]

Corollaries

Strictly subsumes a result by Durand-Mengel ([ICDT'13])

Corollary

#CQ is solvable in time $O\left(\text{poly}(m, n) \cdot N^{O(\text{faqw}^3(\varphi))}\right)$.

Strictly subsumes a result by Chen-Dalmau (LICS'12).

Corollary (QCQ tractability)

QCQ is tractable when faqw is bounded.

Answered an **open question** posed by Durand-Mengel (ICDT'13)

Corollary (#QCQ tractability)

#QCQ is tractable when faqw is bounded.

Many Thanks!
Any FAQ?

FAQ with two blocks of semiring aggregates

$$\varphi = \bigoplus_{\mathbf{x}_L} \bar{\bigoplus}_{\mathbf{x}_{[n]-L}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S).$$

FAQ with two blocks of semiring aggregates

$$\varphi = \bigoplus_{\mathbf{x}_L} \bigoplus_{\mathbf{x}_{[n]-L}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S).$$

Example: #CQ

$$\varphi = \sum_{x_1} \cdots \sum_{x_l} \max_{x_{l+1}} \cdots \max_{x_n} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S).$$

FAQ with two blocks of semiring aggregates

$$\varphi = \bigoplus_{\mathbf{x}_L} \bigoplus_{\mathbf{x}_{[n]-L}} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S).$$

Example: #CQ

$$\varphi = \sum_{x_1} \cdots \sum_{x_l} \max_{x_{l+1}} \cdots \max_{x_n} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S).$$

Precedence Poset:

$u \prec v$ whenever $u \in L$ and $v \notin L$.

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$g(\text{fhtw}(\mathcal{H})),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{faqw}(\varphi) + g(\text{faqw}(\varphi)).$$

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$g(\text{fhtw}(\mathcal{H})),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{faqw}(\varphi) + g(\text{faqw}(\varphi)).$$

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$g(\text{fhtw}(\mathcal{H})),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{OPT} + g(\text{OPT}).$$

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$g(\text{fhtw}(\mathcal{H})),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{OPT} + g(\text{OPT}).$$

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

$$g(\text{fhtw}(\mathcal{H})),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

$$\text{OPT} + g(\text{OPT}).$$

Best known $g(x) = O(x^3)$ [Marx'12]

Approximating $\text{faqw}(\varphi)$

Theorem

Given an approximation algorithm for $\text{fhtw}(\mathcal{H})$ within a bound

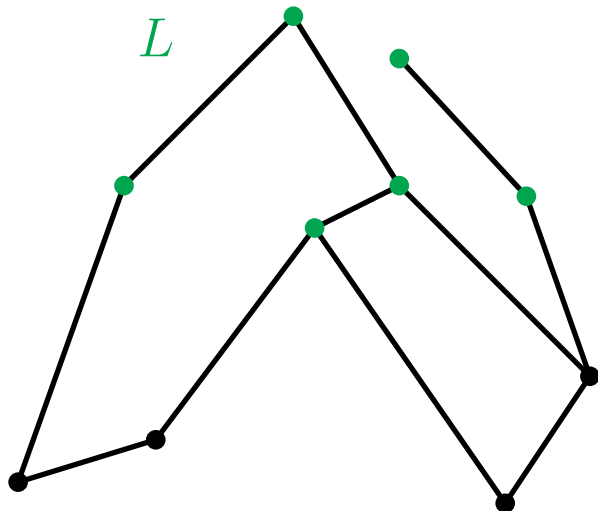
$$O(\text{fhtw}(\mathcal{H})^3),$$

we have an approximation for $\text{faqw}(\varphi)$ within a bound of

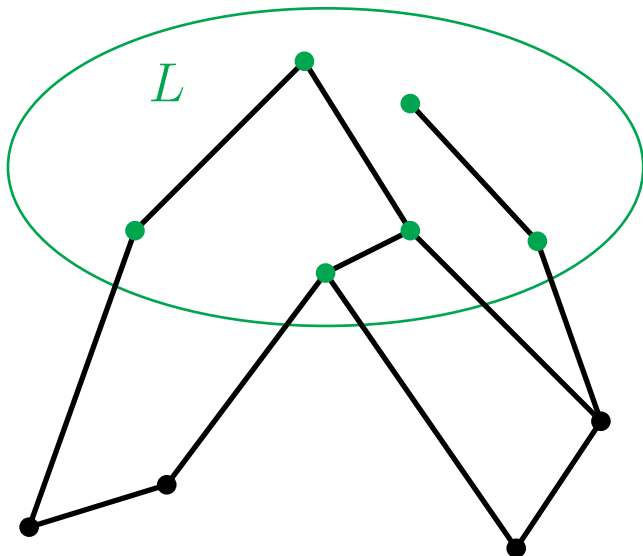
$$\text{OPT} + O(\text{OPT}^3).$$

Best known $g(x) = O(x^3)$ [Marx'12]

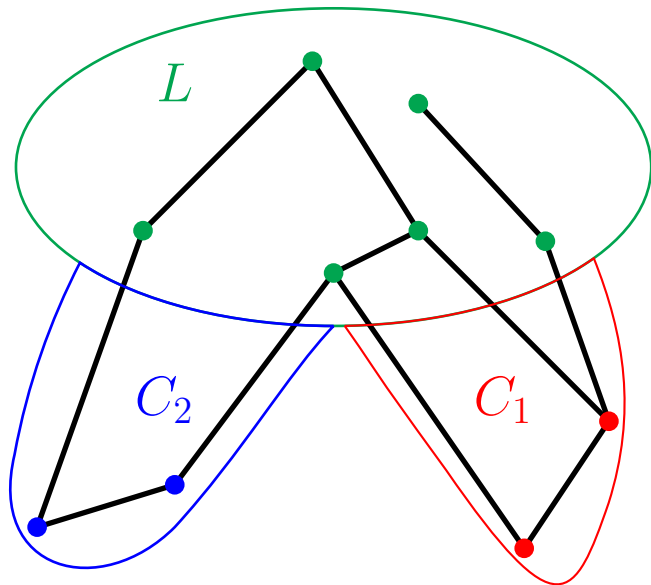
Approximating $\text{faqw}(\varphi)$



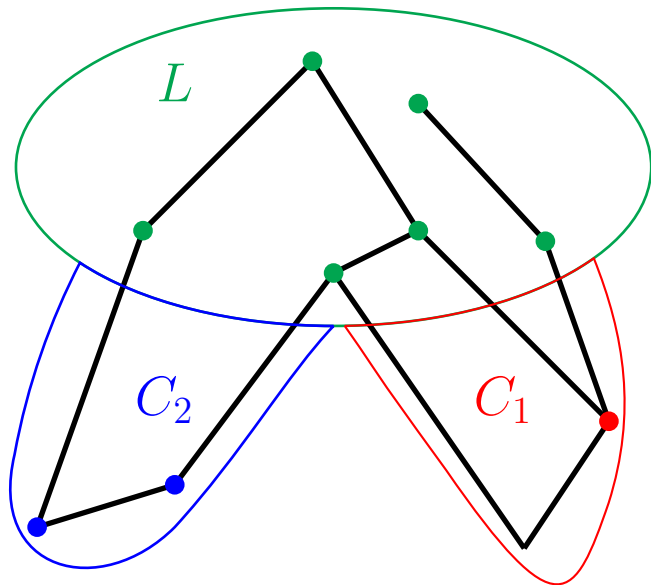
Approximating $\text{faqw}(\varphi)$



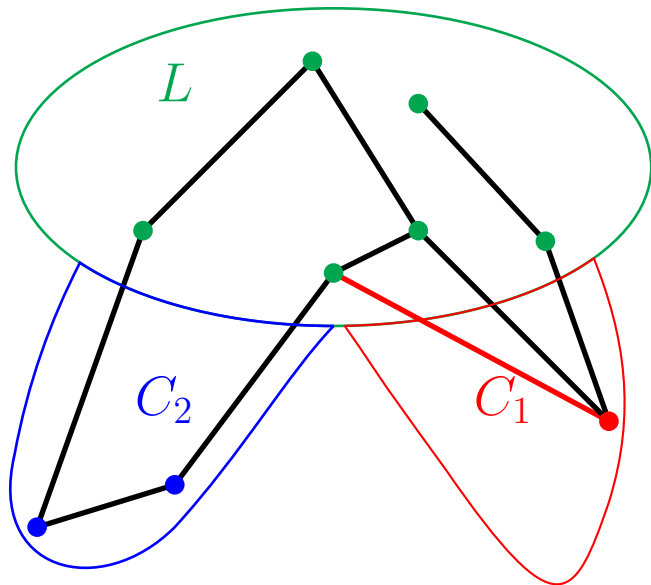
Approximating $\text{faqw}(\varphi)$



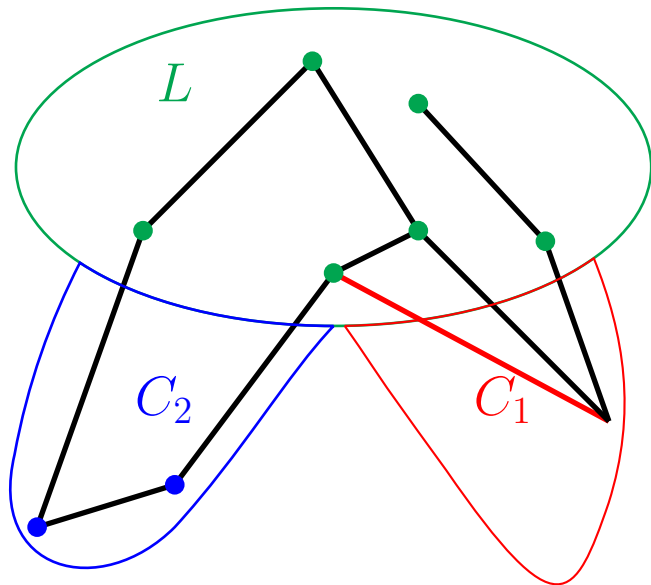
Approximating $\text{faqw}(\varphi)$



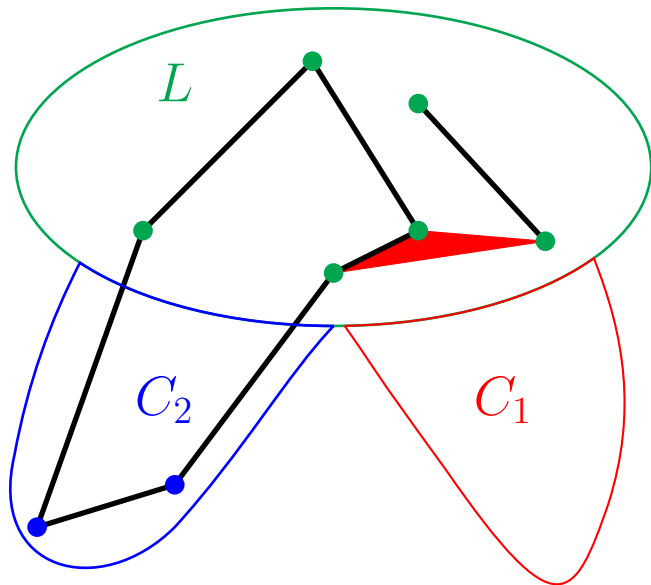
Approximating $\text{faqw}(\varphi)$



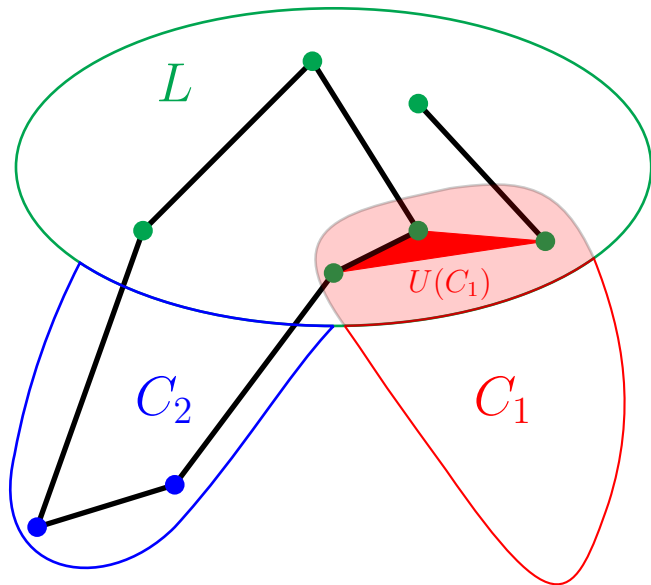
Approximating $\text{faqw}(\varphi)$



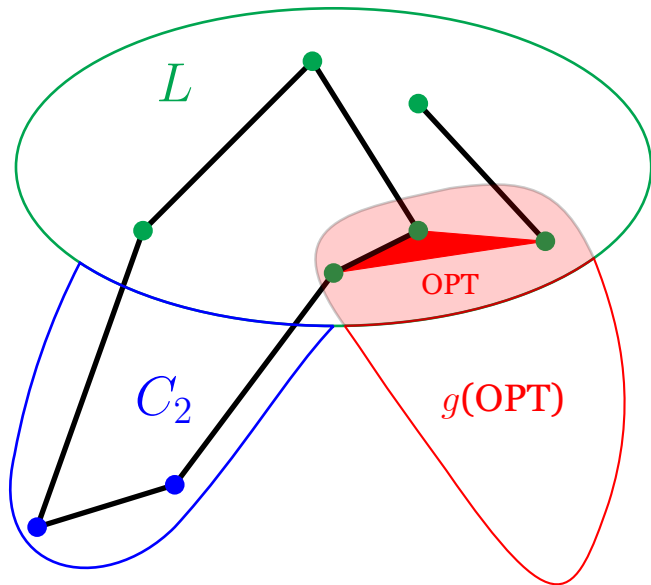
Approximating $\text{faqw}(\varphi)$



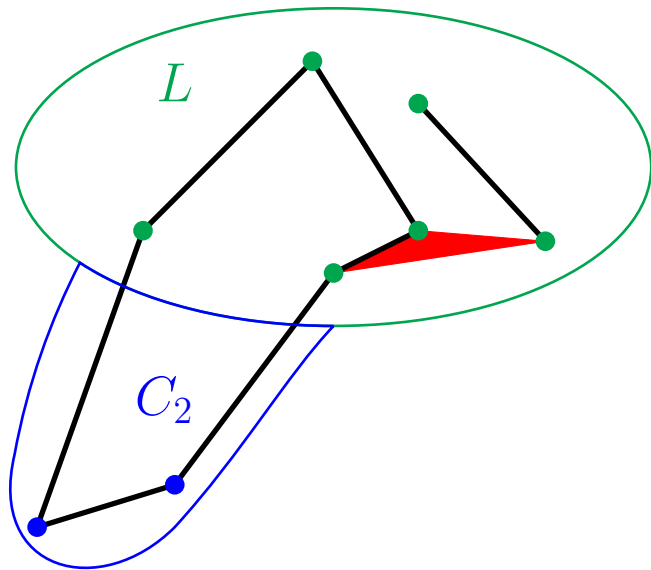
Approximating $\text{faqw}(\varphi)$



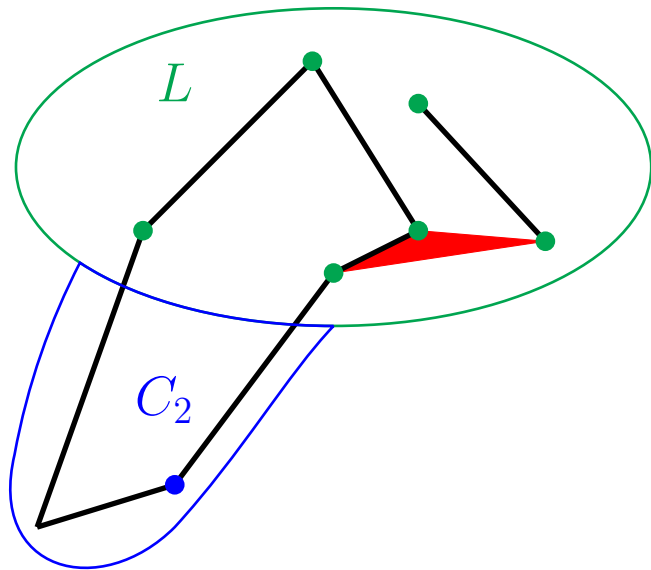
Approximating $\text{faqw}(\varphi)$



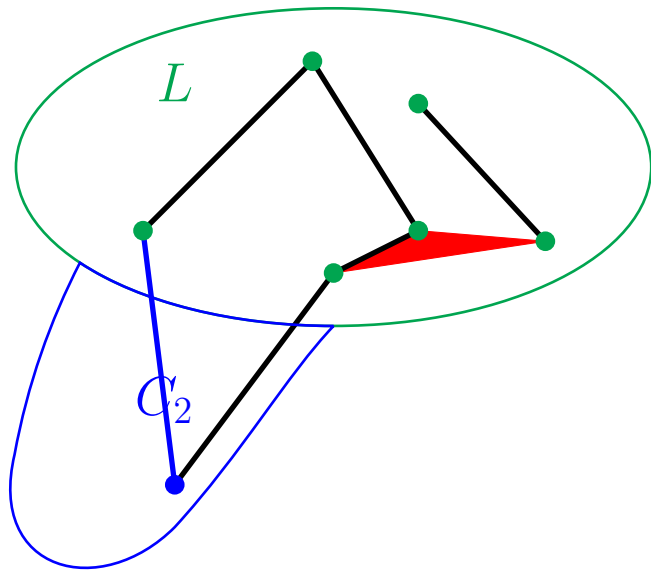
Approximating $\text{faqw}(\varphi)$



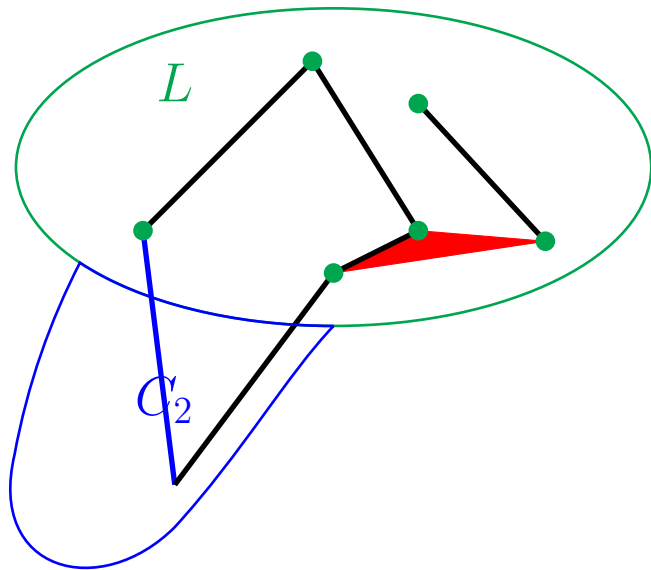
Approximating $\text{faqw}(\varphi)$



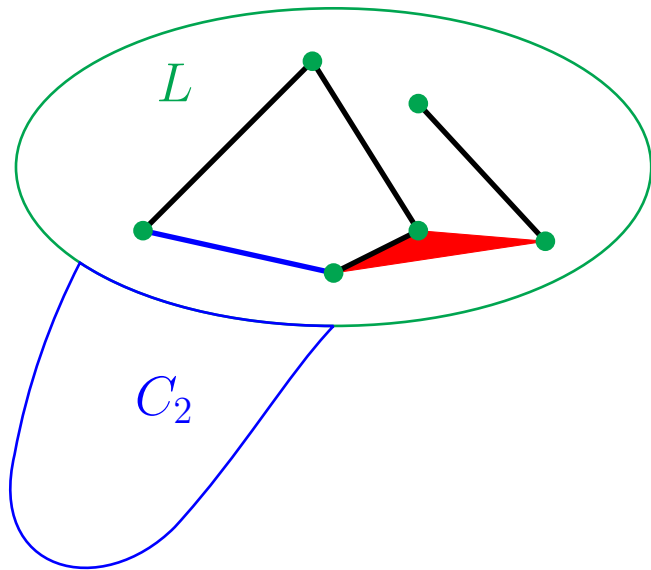
Approximating $\text{faqw}(\varphi)$



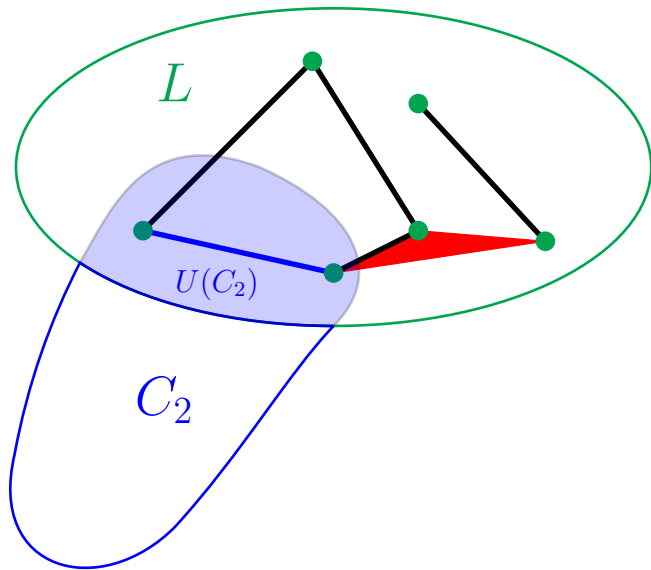
Approximating $\text{faqw}(\varphi)$



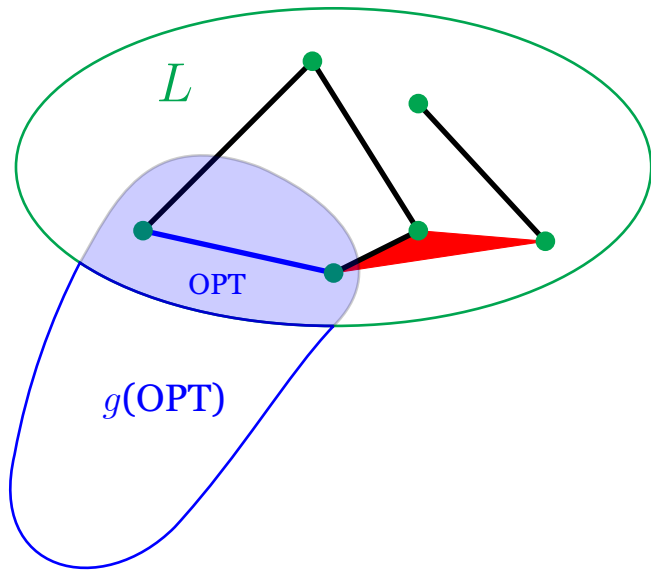
Approximating $\text{faqw}(\varphi)$



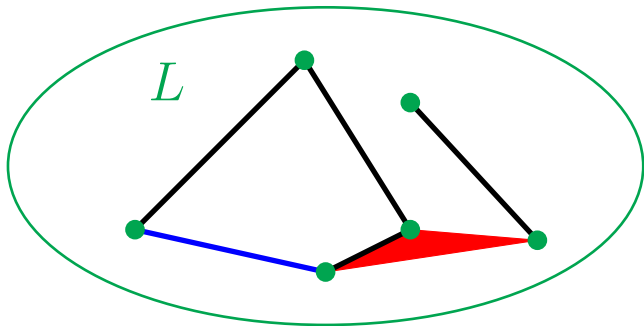
Approximating $\text{faqw}(\varphi)$



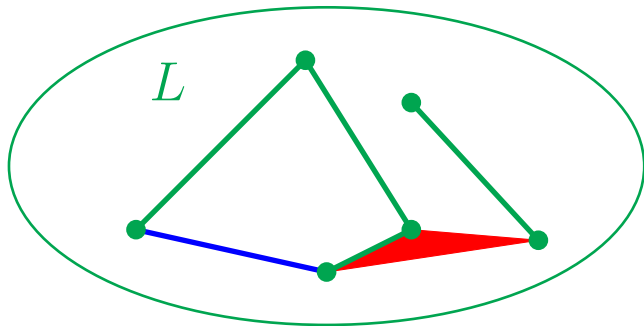
Approximating $\text{faqw}(\varphi)$



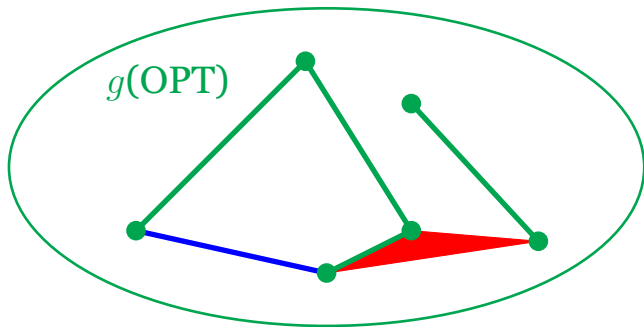
Approximating $\text{faqw}(\varphi)$



Approximating $\text{faqw}(\varphi)$



Approximating $\text{faqw}(\varphi)$



Approximating $\text{faqw}(\varphi)$

Corollary

FAQ with two block of semiring aggregates is solvable in time

$$O\left(\text{poly}(m, n) \cdot N^{O(\text{faqw}^3(\varphi))}\right).$$

Approximating $\text{faqw}(\varphi)$

Corollary

#CQ is solvable in time

$$O\left(\text{poly}(m, n) \cdot N^{O(\text{faqw}^3(\varphi))}\right).$$

Approximating $\text{faqw}(\varphi)$

Corollary

#CQ is solvable in time

$$O\left(\text{poly}(m, n) \cdot N^{O(\text{faqw}^3(\varphi))}\right).$$

Strictly subsumes Durand-Mengel

Theorem (Durand-Mengel [ICDT'13])

#CQ is solvable in time

$$O\left(\text{poly}(|\mathcal{H}|) \cdot N^{\text{poly}(L\text{-ss}(\mathcal{H}), \text{fhtw}(\mathcal{H}))}\right).$$

Completeness Background

Completeness Background

Two aggregates $\oplus, \bar{\oplus}$ are **commutative** iff

Completeness Background

Two aggregates $\oplus, \bar{\oplus}$ are **commutative** iff

$$\begin{array}{ccc} a & \oplus & b \\ & \bar{\oplus} & \\ c & \oplus & d \end{array}$$

Completeness Background

Two aggregates $\oplus, \bar{\oplus}$ are **commutative** iff

$$\begin{array}{c} a \oplus b \\ \bar{\oplus} \\ c \oplus d \end{array} =$$

Completeness Background

Two aggregates $\oplus, \bar{\oplus}$ are **commutative** iff

$$\begin{array}{c} a \oplus b \\ \bar{\oplus} \\ c \oplus d \end{array} = \begin{array}{c} a \\ \bar{\oplus} \\ c \end{array} \oplus \begin{array}{c} b \\ \bar{\oplus} \\ d \end{array}$$

Completeness Background

Two aggregates $\oplus, \bar{\oplus}$ are **commutative** iff

$$\begin{array}{c} a \oplus \mathbf{0} \\ \bar{\oplus} \\ \mathbf{0} \oplus d \end{array} = \begin{array}{c} a \\ \bar{\oplus} \\ \mathbf{0} \end{array} \oplus \begin{array}{c} \mathbf{0} \\ \bar{\oplus} \\ d \end{array}$$

Completeness Background

Two aggregates $\oplus, \bar{\oplus}$ are **commutative** iff

$$\begin{array}{c} a \\ \bar{\oplus} \\ d \end{array} = \begin{array}{c} a \\ \oplus \\ d \end{array}$$

Completeness Background

Two aggregates $\oplus, \bar{\oplus}$ are **commutative** iff

$$a \bar{\oplus} d = a \oplus d$$

Completeness Background

Two aggregates $\oplus, \bar{\oplus}$ are **commutative** iff

$$a \bar{\oplus} d = a \oplus d$$

Hence, \oplus and $\bar{\oplus}$ are identical!

Soundness and Completeness

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Soundness and Completeness

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness

$$\text{LinEx}(P) = \text{EVO}(\varphi) ?$$

Soundness and Completeness

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness

$$\text{LinEx}(P) = \text{EVO}(\varphi) ?$$

$$\varphi = \sum_{x_1} \max_{x_2} \sum_{x_3} \psi_{12} \psi_{13}$$

Soundness and Completeness

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness

$$\text{LinEx}(P) = \text{EVO}(\varphi) ?$$

$$\varphi = \sum_{x_1} \max_{x_2} \sum_{x_3} \psi_{12} \psi_{13}$$

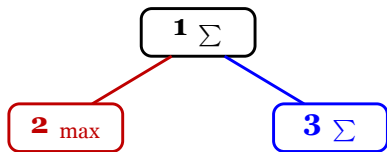
Soundness and Completeness

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness

$$\text{LinEx}(P) = \text{EVO}(\varphi) ?$$



$$\begin{aligned}\varphi &= \sum_{x_1} \max_{x_2} \sum_{x_3} \psi_{12} \psi_{13} \\ &= \sum_{x_1} \sum_{x_3} \psi_{13} \max_{x_2} \psi_{12}\end{aligned}$$

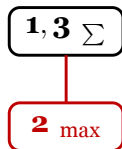
Soundness and Completeness

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness

$$\text{LinEx}(P) = \text{EVO}(\varphi) \text{ ?}$$



$$\begin{aligned}\varphi &= \sum_{x_1} \max_{x_2} \sum_{x_3} \psi_{12} \psi_{13} \\ &= \sum_{x_1} \sum_{x_3} \psi_{13} \max_{x_2} \psi_{12}\end{aligned}$$

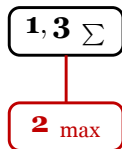
Soundness and Completeness

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness

$$\text{LinEx}(P) \stackrel{?}{=} \text{EVO}(\varphi)$$



$$\begin{aligned}\varphi &= \sum_{x_1} \max_{x_2} \sum_{x_3} \psi_{12} \psi_{13} \\ &= \sum_{x_1} \sum_{x_3} \psi_{13} \max_{x_2} \psi_{12}\end{aligned}$$

$(1, 2, 3) \notin \text{LinEx}(P) !$

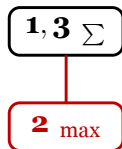
Soundness and Completeness

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness

$$\text{LinEx}(P) \neq \text{EVO}(\varphi).$$



$$\begin{aligned}\varphi &= \sum_{x_1} \max_{x_2} \sum_{x_3} \psi_{12} \psi_{13} \\ &= \sum_{x_1} \sum_{x_3} \psi_{13} \max_{x_2} \psi_{12}\end{aligned}$$

$$(1, 2, 3) \notin \text{LinEx}(P) !$$

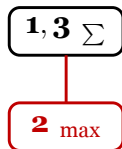
Soundness and Completeness

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness

$$\text{LinEx}(P) \neq \text{EVO}(\varphi).$$



$$\begin{aligned}\varphi &= \sum_{x_1} \max_{x_2} \sum_{x_3} \psi_{12} \psi_{13} \\ &= \sum_{x_1} \sum_{x_3} \psi_{13} \max_{x_2} \psi_{12}\end{aligned}$$

$$(1, 2, 3) \notin \text{LinEx}(P) !$$

$$\text{faqw}(\varphi) := \min_{\sigma \in \text{EVO}(\varphi)} \text{faqw}(\sigma).$$

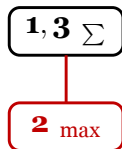
Soundness and Completeness

Soundness

$$\text{LinEx}(P) \subseteq \text{EVO}(\varphi).$$

Completeness

$$\text{LinEx}(P) \neq \text{EVO}(\varphi).$$



$$\begin{aligned}\varphi &= \sum_{x_1} \max_{x_2} \sum_{x_3} \psi_{12} \psi_{13} \\ &= \sum_{x_1} \sum_{x_3} \psi_{13} \max_{x_2} \psi_{12}\end{aligned}$$

$$(1, 2, 3) \notin \text{LinEx}(P) !$$

Theorem $\text{faqw}(\varphi) := \min_{\sigma \in \text{LinEx}(P)} \text{faqw}(\sigma).$

FAQ with *idempotent* product aggregates

FAQ with *idempotent* product aggregates

Let $\mathbf{D}_I \supseteq \{\mathbf{0}, \mathbf{1}\}$ be a set of **idempotent** elements of \otimes .

FAQ with *idempotent* product aggregates

Let $\mathbf{D}_I \supseteq \{\mathbf{0}, \mathbf{1}\}$ be a set of **idempotent** elements of \otimes .

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \dots \bigoplus_{x_{f+\ell}}^{(f+\ell)} \bigoplus_{x_{f+\ell+1}}^{(f+\ell+1)} \dots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S)$$

FAQ with *idempotent* product aggregates

Let $\mathbf{D}_I \supseteq \{\mathbf{0}, \mathbf{1}\}$ be a set of **idempotent** elements of \otimes .

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \dots \bigoplus_{x_{f+\ell}}^{(f+\ell)} \bigoplus_{x_{f+\ell+1}}^{(f+\ell+1)} \dots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \underbrace{\psi_S(\mathbf{x}_S)}_{\in \mathbf{D}_I}$$

FAQ with *idempotent* product aggregates

Let $\mathbf{D}_I \supseteq \{\mathbf{0}, \mathbf{1}\}$ be a set of **idempotent** elements of \otimes .

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \dots \bigoplus_{x_{f+\ell}}^{(f+\ell)} \bigoplus_{x_{f+\ell+1}}^{(f+\ell+1)} \dots \bigoplus_{x_n}^{(n)} \underbrace{\bigotimes_{S \in \mathcal{E}}}_{\substack{\text{Closed} \\ \text{under } \mathbf{D}_I}} \underbrace{\psi_S(\mathbf{x}_S)}_{\in \mathbf{D}_I}$$

FAQ with *idempotent* product aggregates

Let $\mathbf{D}_I \supseteq \{\mathbf{0}, \mathbf{1}\}$ be a set of **idempotent** elements of \otimes .

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \cdots \bigoplus_{x_{f+\ell}}^{(f+\ell)} \underbrace{\bigoplus_{x_{f+\ell+1}}^{(f+\ell+1)} \cdots \bigoplus_{x_n}^{(n)}}_{\substack{\text{Products or semirings} \\ \text{closed under } \mathbf{D}_I}} \underbrace{\bigotimes_{S \in \mathcal{E}}}_{\substack{\text{Closed} \\ \text{under } \mathbf{D}_I}} \underbrace{\psi_S(\mathbf{x}_S)}_{\in \mathbf{D}_I}$$

FAQ with *idempotent* product aggregates

Let $\mathbf{D}_I \supseteq \{\mathbf{0}, \mathbf{1}\}$ be a set of **idempotent** elements of \otimes .

$$\varphi(\mathbf{x}_{[f]}) = \underbrace{\bigoplus_{x_{f+1}}^{(f+1)} \dots \bigoplus_{x_{f+\ell}}^{(f+\ell)}}_{\text{Semiring aggregates}} \underbrace{\bigoplus_{x_{f+\ell+1}}^{(f+\ell+1)} \dots \bigoplus_{x_n}^{(n)}}_{\substack{\text{Products or semirings} \\ \text{closed under } \mathbf{D}_I}} \underbrace{\bigotimes_{S \in \mathcal{E}}}_{\substack{\text{Closed} \\ \text{under } \mathbf{D}_I}} \underbrace{\psi_S(\mathbf{x}_S)}_{\in \mathbf{D}_I}$$

FAQ with *idempotent* product aggregates

Let $\mathbf{D}_I \supseteq \{\mathbf{0}, \mathbf{1}\}$ be a set of **idempotent** elements of \otimes .

$$\varphi(\mathbf{x}_{[f]}) = \underbrace{\bigoplus_{x_{f+1}}^{(f+1)} \dots \bigoplus_{x_{f+\ell}}^{(f+\ell)}}_{\text{Semiring aggregates}} \underbrace{\bigoplus_{x_{f+\ell+1}}^{(f+\ell+1)} \dots \bigoplus_{x_n}^{(n)}}_{\substack{\text{Products or semirings} \\ \text{closed under } \mathbf{D}_I}} \underbrace{\bigotimes_{S \in \mathcal{E}}}_{\substack{\text{Closed} \\ \text{under } \mathbf{D}_I}} \underbrace{\psi_S(\mathbf{x}_S)}_{\in \mathbf{D}_I}$$

Examples: QCQ,

$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \dots \bigoplus_{x_n}^{(n)} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S).$$

FAQ with *idempotent* product aggregates

Let $\mathbf{D}_I \supseteq \{\mathbf{0}, \mathbf{1}\}$ be a set of **idempotent** elements of \otimes .

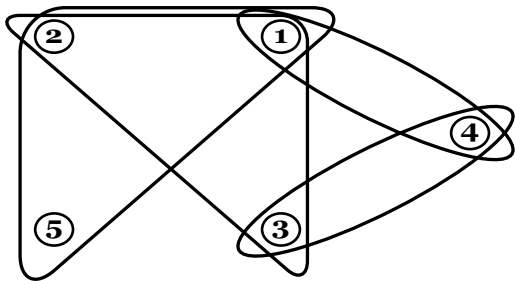
$$\varphi(\mathbf{x}_{[f]}) = \underbrace{\bigoplus_{x_{f+1}}^{(f+1)} \dots \bigoplus_{x_{f+\ell}}^{(f+\ell)}}_{\text{Semiring aggregates}} \underbrace{\bigoplus_{x_{f+\ell+1}}^{(f+\ell+1)} \dots \bigoplus_{x_n}^{(n)}}_{\substack{\text{Products or semirings} \\ \text{closed under } \mathbf{D}_I}} \underbrace{\bigotimes_{S \in \mathcal{E}}}_{\substack{\text{Closed} \\ \text{under } \mathbf{D}_I}} \underbrace{\psi_S(\mathbf{x}_S)}_{\in \mathbf{D}_I}$$

Examples: QCQ, #QCQ

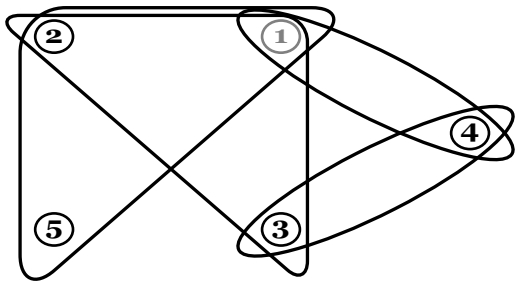
$$\varphi(\mathbf{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \dots \bigoplus_{x_n}^{(n)} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S).$$

$$\varphi = \sum_{x_1} \dots \sum_{x_l} \bigoplus_{x_{l+1}}^{(l+1)} \dots \bigoplus_{x_n}^{(n)} \prod_{S \in \mathcal{E}} \psi_S(\mathbf{x}_S).$$

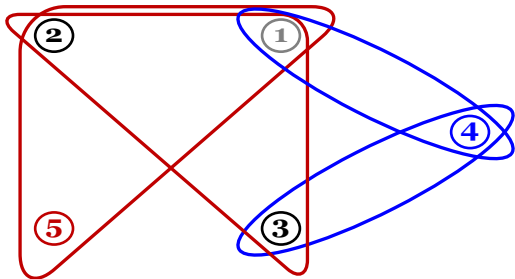
$$\varphi = \sum_{x_1} \max_{x_2} \prod_{x_3} \max_{x_4} \max_{x_5} \psi_{123} \psi_{125} \psi_{14} \psi_{34}$$



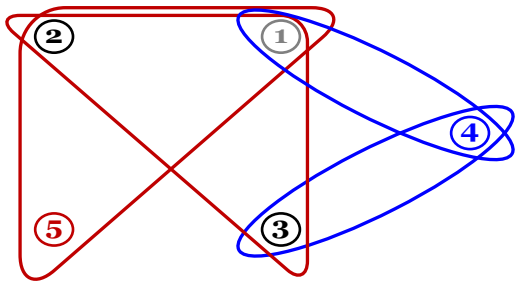
$$\varphi = \sum_{x_1} \max_{x_2} \prod_{x_3} \max_{x_4} \max_{x_5} \psi_{123} \psi_{125} \psi_{14} \psi_{34}$$



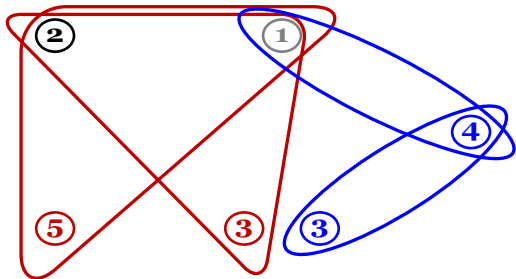
$$\varphi = \sum_{x_1} \max_{x_2} \prod_{x_3} \max_{x_4} \max_{x_5} \psi_{123} \psi_{125} \psi_{14} \psi_{34}$$



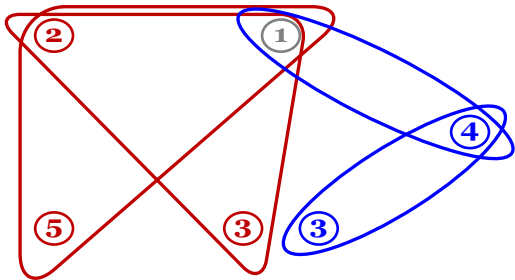
$$\varphi = \sum_{x_1} \max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123} \psi_{125} \max_{x_4} \psi_{14} \psi_{34}$$



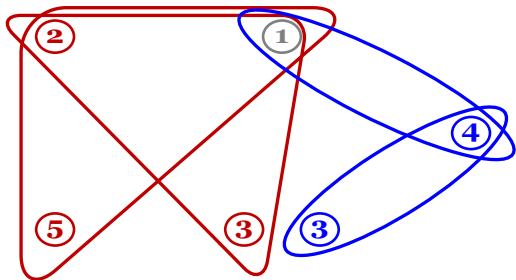
$$\varphi = \sum_{x_1} \max_{x_2} \left(\prod_{x_3} \max_{x_5} \psi_{123} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$



$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

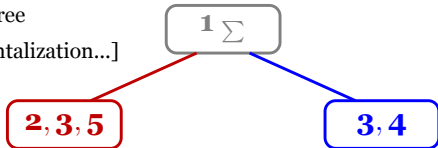


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

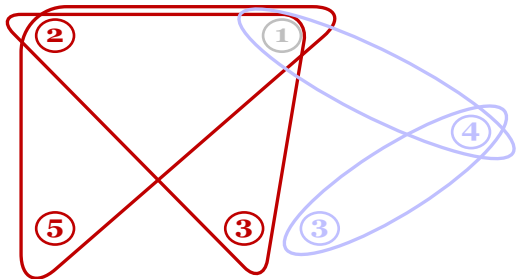


Expression Tree

[Compartmentalization...]

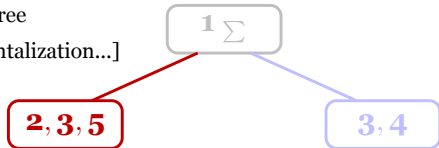


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

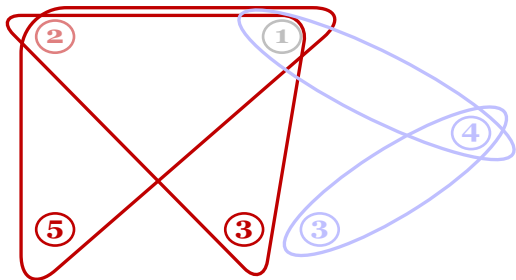


Expression Tree

[Compartmentalization...]

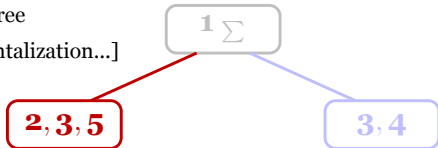


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

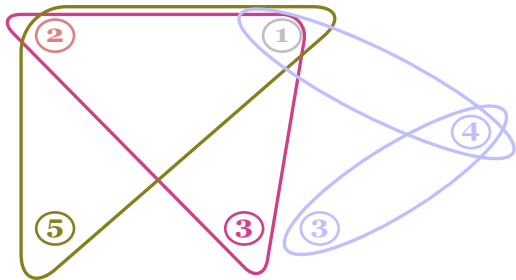


Expression Tree

[Compartmentalization...]

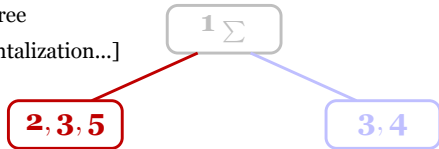


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

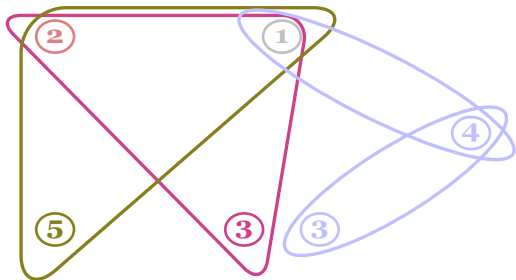


Expression Tree

[Compartmentalization...]

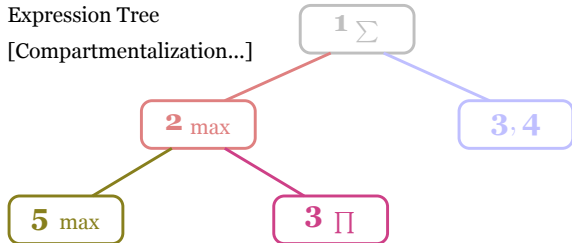


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

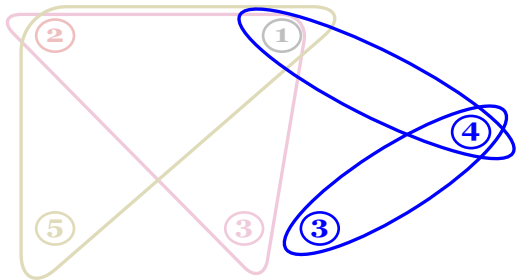


Expression Tree

[Compartmentalization...]

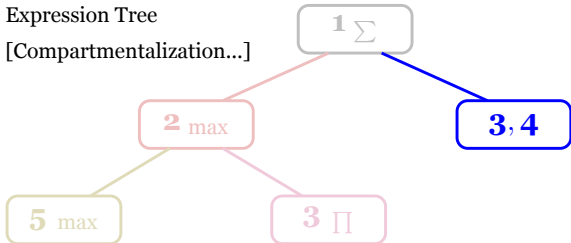


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

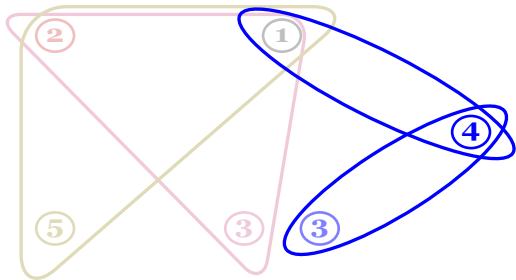


Expression Tree

[Compartmentalization...]

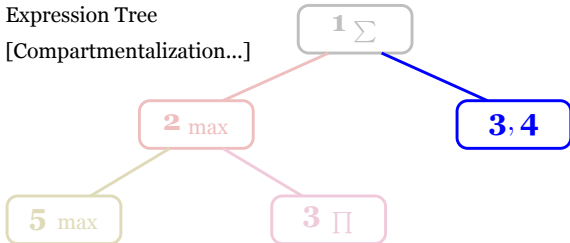


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

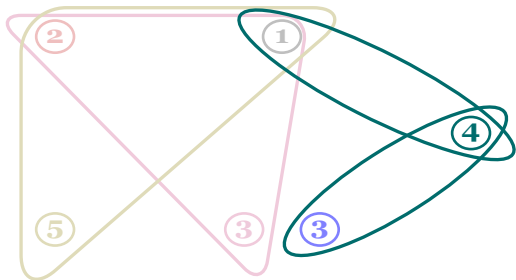


Expression Tree

[Compartmentalization...]

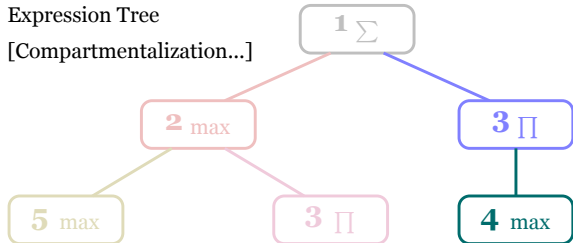


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

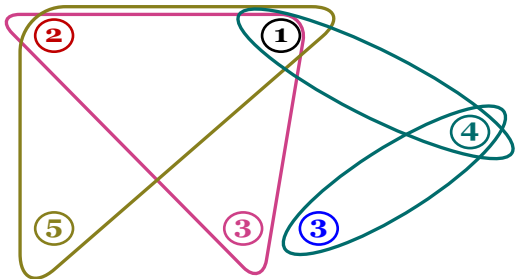


Expression Tree

[Compartmentalization...]

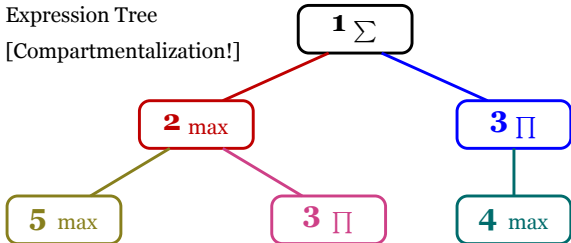


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

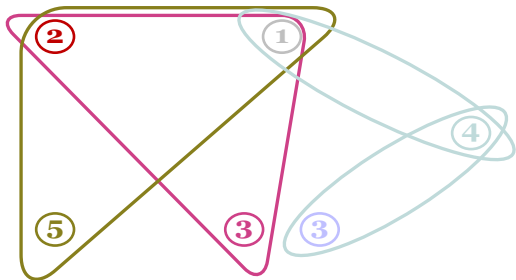


Expression Tree

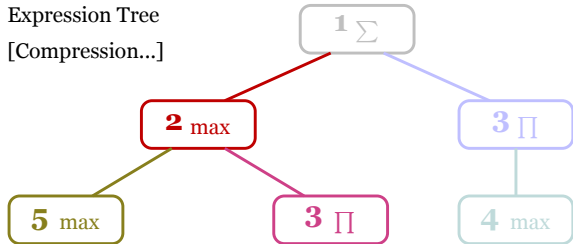
[Compartmentalization!]



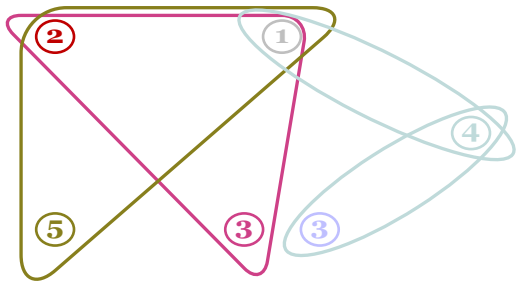
$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$



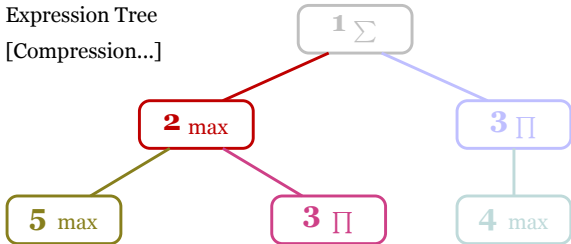
Expression Tree
[Compression...]



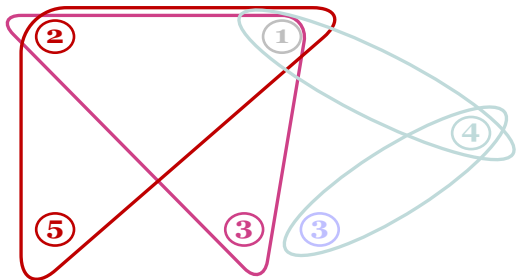
$$\varphi = \sum_{x_1} \left(\max_{x_2} \max_{x_5} \psi_{125} \prod_{x_3} \psi_{123} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$



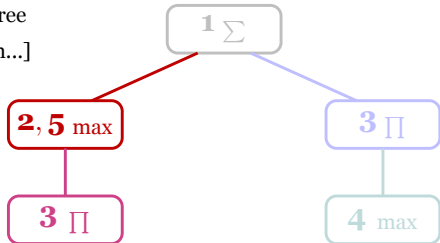
Expression Tree
[Compression...]



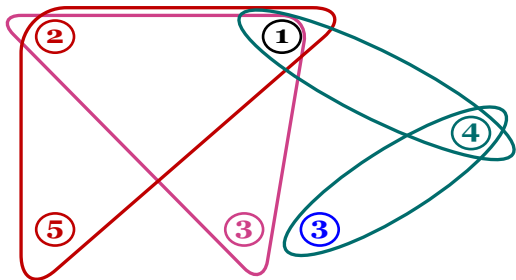
$$\varphi = \sum_{x_1} \left(\max_{x_2} \max_{x_5} \psi_{125} \prod_{x_3} \psi_{123} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$



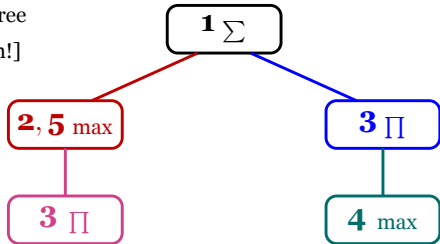
Expression Tree
[Compression...]



$$\varphi = \sum_{x_1} \left(\max_{x_2} \max_{x_5} \psi_{125} \prod_{x_3} \psi_{123} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$



Expression Tree
[Compression!]



Variable ordering and hypergraph sequence

$$\mathcal{V}_8 = \mathcal{V}$$

1 2 3 4 5 6 7 8

1 1 1

$$\mathcal{E}_8 = \mathcal{E}$$

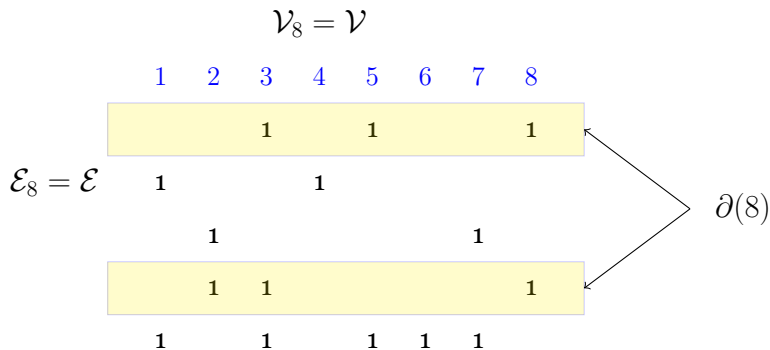
1 1

1 1

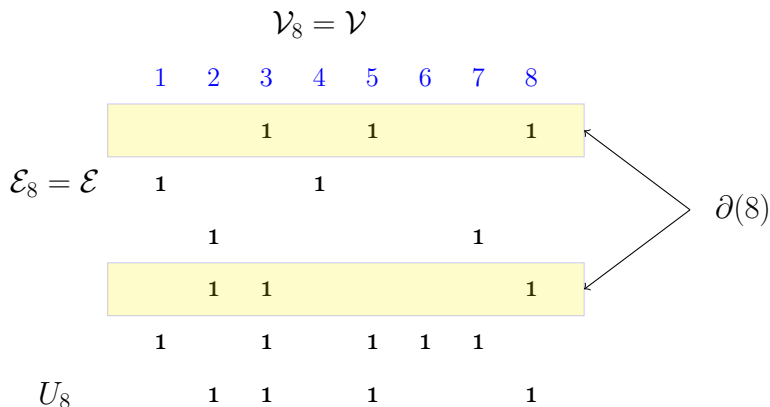
1 1 1

1 1 1 1 1

Variable ordering and hypergraph sequence



Variable ordering and hypergraph sequence



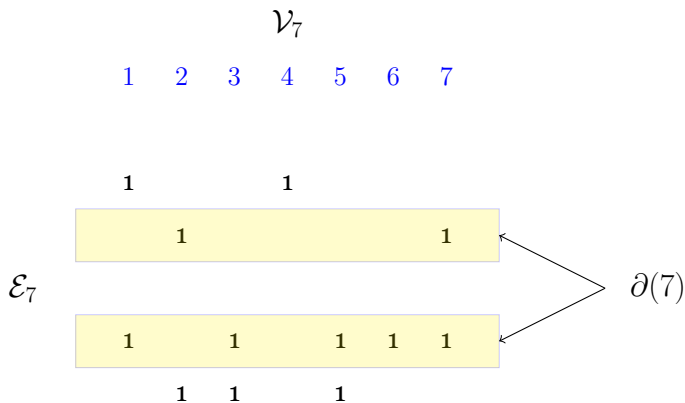
Variable ordering and hypergraph sequence

	\mathcal{V}_7						
	1	2	3	4	5	6	7
\mathcal{E}_7	1			1			
		1					1
	1		1		1	1	1
$U_8 - \{8\}$		1	1		1		

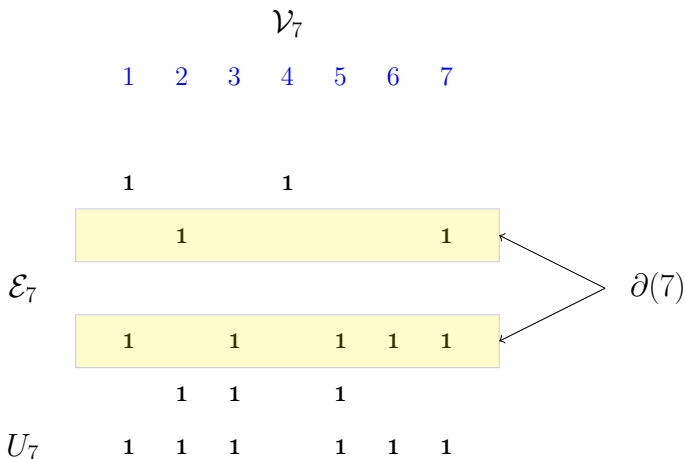
Variable ordering and hypergraph sequence

	\mathcal{V}_7						
	1	2	3	4	5	6	7
\mathcal{E}_7	1			1			
		1					1
	1		1		1	1	1
		1	1		1		

Variable ordering and hypergraph sequence



Variable ordering and hypergraph sequence



Variable ordering and hypergraph sequence

 \mathcal{V}_6

1 2 3 4 5 6

1 1

 \mathcal{E}_6

1 1 1

$U_7 - \{7\}$ 1 1 1 1 1

Variable ordering and hypergraph sequence

\mathcal{V}_6

1 2 3 4 5 6

1 1

\mathcal{E}_6

1 1 1
1 1 1 1 1

Variable ordering and hypergraph sequence

 \mathcal{V}_6

1 2 3 4 5 6

1 1

 \mathcal{E}_6

1 1 1

1 1 1 1 1

← $\partial(6)$

Variable ordering and hypergraph sequence

 \mathcal{V}_6

1 2 3 4 5 6

1 1

 \mathcal{E}_6

1 1 1

1 1 1 1 1

← $\partial(6)$

 \mathcal{U}_6

1 1 1 1 1

Variable ordering and hypergraph sequence

\mathcal{V}_5

1 2 3 4 5

1 1

\mathcal{E}_5

1 1 1

$U_6 - \{6\}$ 1 1 1 1

Variable ordering and hypergraph sequence

- ▶ Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a **variable ordering** v_1, \dots, v_n
- ▶ $\mathcal{H}_n = \mathcal{H}$
- ▶ For $k = n - 1, \dots, 1$, define $\mathcal{H}_k = (\mathcal{V}_k, \mathcal{E}_k)$

$$\partial(v_{k+1}) = \{S \mid S \in \mathcal{E}_{k+1} \text{ and } v_{k+1} \in S\}$$

Variable ordering and hypergraph sequence

- ▶ Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a **variable ordering** v_1, \dots, v_n
- ▶ $\mathcal{H}_n = \mathcal{H}$
- ▶ For $k = n - 1, \dots, 1$, define $\mathcal{H}_k = (\mathcal{V}_k, \mathcal{E}_k)$

$$\partial(v_{k+1}) = \{S \mid S \in \mathcal{E}_{k+1} \text{ and } v_{k+1} \in S\}$$

$$U_{k+1} = \bigcup_{S \in \partial(v_{k+1})} S$$

Variable ordering and hypergraph sequence

- ▶ Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a **variable ordering** v_1, \dots, v_n
- ▶ $\mathcal{H}_n = \mathcal{H}$
- ▶ For $k = n - 1, \dots, 1$, define $\mathcal{H}_k = (\mathcal{V}_k, \mathcal{E}_k)$

$$\partial(v_{k+1}) = \{S \mid S \in \mathcal{E}_{k+1} \text{ and } v_{k+1} \in S\}$$

$$U_{k+1} = \bigcup_{S \in \partial(v_{k+1})} S$$

$$\mathcal{V}_k = \mathcal{V}_{k+1} - \{v_{k+1}\}$$

Variable ordering and hypergraph sequence

- ▶ Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a **variable ordering** v_1, \dots, v_n
- ▶ $\mathcal{H}_n = \mathcal{H}$
- ▶ For $k = n - 1, \dots, 1$, define $\mathcal{H}_k = (\mathcal{V}_k, \mathcal{E}_k)$

$$\partial(v_{k+1}) = \{S \mid S \in \mathcal{E}_{k+1} \text{ and } v_{k+1} \in S\}$$

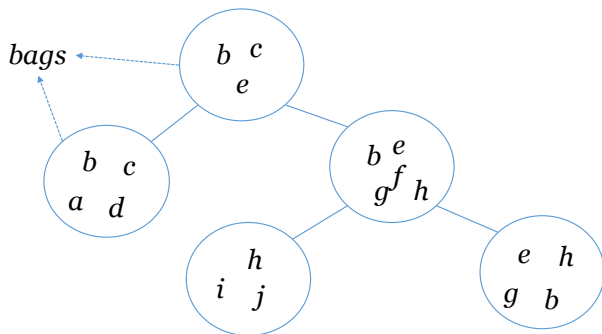
$$U_{k+1} = \bigcup_{S \in \partial(v_{k+1})} S$$

$$\mathcal{V}_k = \mathcal{V}_{k+1} - \{v_{k+1}\}$$

$$\mathcal{E}_k = (\mathcal{E}_{k+1} - \partial(v_{k+1})) \cup \{U_{k+1} - \{v_{k+1}\}\}$$

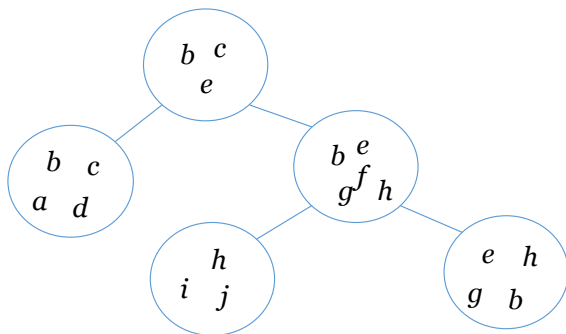
Tree decomposition of $\mathcal{H} = (\mathcal{V}, \mathcal{E})$

$\{a, b, d\}, \{c, d\}, \{b, c, d\}, \{b, e\}, \{c, e\}, \{b, e, f\},$
 $\{b, e, g\}, \{g, f, h\}, \{i, j, h\}, \{e, g\}, \{e, g, b\}, \{e, b, h\}.$



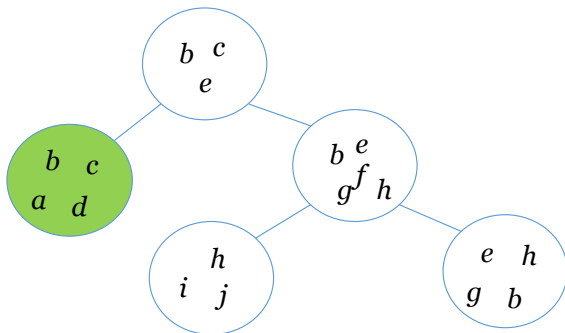
Every hyperedge is covered by some bag

$\{a, b, d\}$, $\{c, d\}$, $\{b, c, d\}$, $\{b, e\}$, $\{c, e\}$, $\{b, e, f\}$,
 $\{b, e, g\}$, $\{g, f, h\}$, $\{i, j, h\}$, $\{e, g\}$, $\{e, g, b\}$, $\{e, b, h\}$.



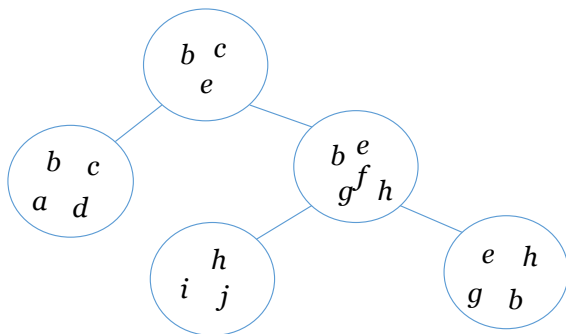
Every hyperedge is covered by some bag

$\{a, b, d\}$, $\{c, d\}$, $\{b, c, d\}$, $\{b, e\}$, $\{c, e\}$, $\{b, e, f\}$,
 $\{b, e, g\}$, $\{g, f, h\}$, $\{i, j, h\}$, $\{e, g\}$, $\{e, g, b\}$, $\{e, b, h\}$.



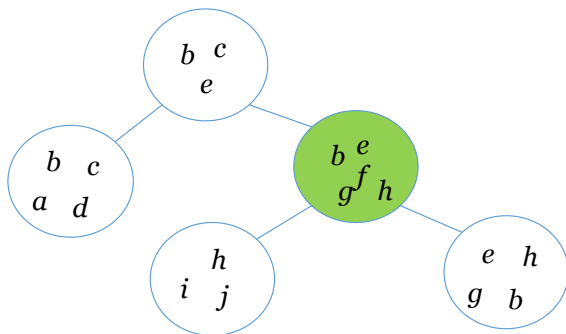
Every hyperedge is covered by some bag

$\{a,b,d\}$, $\{c,d\}$, $\{b,c,d\}$, $\{b,e\}$, $\{c,e\}$, $\{b,e,f\}$,
 $\{b,e,g\}$, $\{g,f,h\}$, $\{i,j,h\}$, $\{e,g\}$, $\{e,g,b\}$, $\{e,b,h\}$.



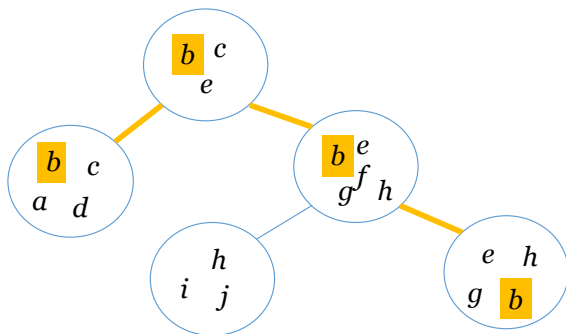
Every hyperedge is covered by some bag

$\{a,b,d\}$, $\{c,d\}$, $\{b,c,d\}$, $\{b,e\}$, $\{c,e\}$, $\{b,e,f\}$,
 $\{b,e,g\}$, $\{g,f,h\}$, $\{i,j,h\}$, $\{e,g\}$, $\{e,g,b\}$, $\{e,b,h\}$.



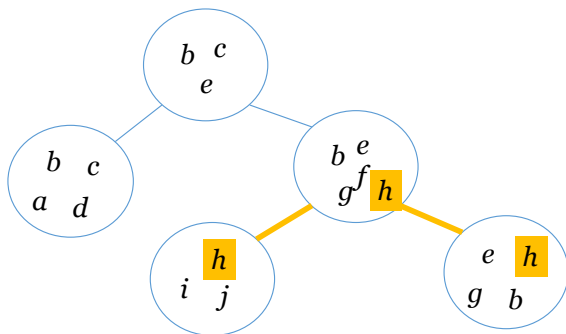
Running intersection property (RIP)

$\{a, b, d\}$, $\{c, d\}$, $\{b, c, d\}$, $\{b, e\}$, $\{c, e\}$, $\{b, e, f\}$,
 $\{b, e, g\}$, $\{g, f, h\}$, $\{i, j, h\}$, $\{e, g\}$, $\{e, g, b\}$, $\{e, b, h\}$.



Running intersection property (RIP)

$\{a,b,d\}$, $\{c,d\}$, $\{b,c,d\}$, $\{b,e\}$, $\{c,e\}$, $\{b,e,f\}$,
 $\{b,e,g\}$, $\{g,f,h\}$, $\{i,j,h\}$, $\{e,g\}$, $\{e,g,b\}$, $\{e,b,h\}$.



Variable Ordering and Tree decomposition

- ▶ Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a **variable ordering** v_1, \dots, v_n
- ▶ $\mathcal{H}_n = \mathcal{H}$
- ▶ For $k = n - 1, \dots, 1$, define $\mathcal{H}_k = (\mathcal{V}_k, \mathcal{E}_k)$

$$\partial(v_{k+1}) = \{S \mid S \in \mathcal{E}_{k+1} \text{ and } v_{k+1} \in S\}$$

$$U_{k+1} = \bigcup_{S \in \partial(v_{k+1})} S$$

$$\mathcal{V}_k = \mathcal{V}_{k+1} - \{v_{k+1}\}$$

$$\mathcal{E}_k = (\mathcal{E}_{k+1} - \partial(v_{k+1})) \cup \{U_{k+1} - \{v_{k+1}\}\}$$

Proposition (Folklorish)

*There is a tree decomposition whose bags are the sets U_k **and vice versa.***