Functional Aggregate Queries are FAQs

Hung Q. Ngo
Relational AI Inc.

Based on joint work with Mahmoud Abo Khamis and Atri Rudra

Table of Contents

Table of Contents

Notations

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

Notations

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

Notations

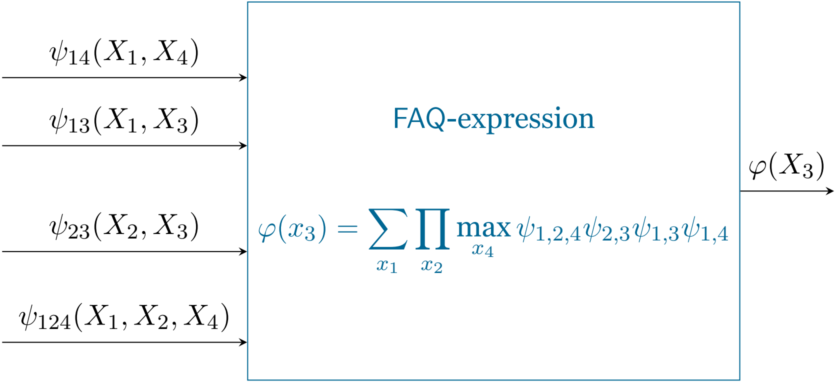
- X_i denote variables, $i \in [n] = \{1, \ldots, n\}$,
- x_i are values in discrete domain $\mathsf{Dom}(X_i)$
- $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathsf{Dom}(X_1) \times \cdots \times \mathsf{Dom}(X_n)$

Notations

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

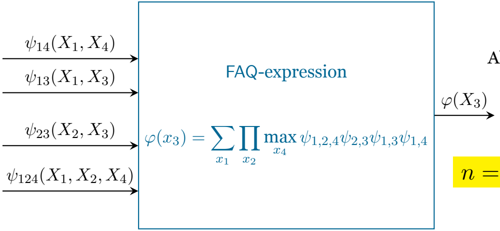
$$
\begin{aligned}
\boldsymbol{x}_S &= (x_i)_{i \in S} \in \prod_{i \in S} \mathsf{Dom}(X_i) \\
\text{e.g. } \boldsymbol{x}_{\{2,5,8\}} &= (x_2, x_5, x_8) \in \mathsf{Dom}(X_2) \times \mathsf{Dom}(X_5) \times \mathsf{Dom}(X_8)
\end{aligned}
$$

Function Aggregate Query: the Problem



All functions have the same range **D**

Function Aggregate Query: the Input

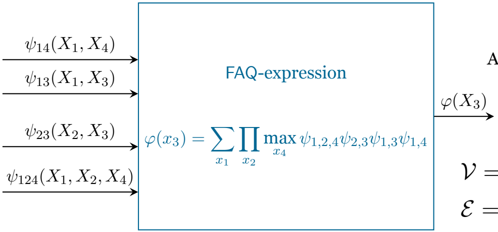


All functions have the same range **D**

$n = 4$

- n *variables* $X_1, \ldots, X_n$

Function Aggregate Query: the Input

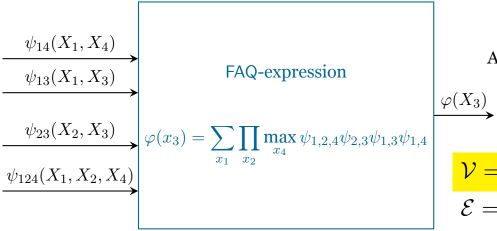


All functions have the same range **D**

$$\mathcal{V} = \{1, 2, 3, 4\}$$
$$\mathcal{E} = \{\{1, 4\}, \{1, 3\}, \{2, 3\}, \{1, 2, 4\}\}$$

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

Function Aggregate Query: the Input



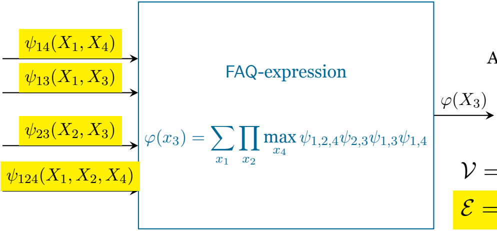
All functions have the same range **D**

$\mathcal{V} = \{1, 2, 3, 4\}$

$\mathcal{E} = \{\{1, 4\}, \{1, 3\}, \{2, 3\}, \{1, 2, 4\}\}$

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

Function Aggregate Query: **the Input**



All functions have the same range **D**

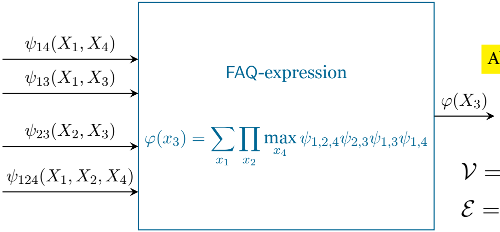
$\mathcal{V} = \{1, 2, 3, 4\}$

$\mathcal{E} = \{\{1, 4\}, \{1, 3\}, \{2, 3\}, \{1, 2, 4\}\}$

- 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

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

Function Aggregate Query: the Input



All functions have the same range **D**

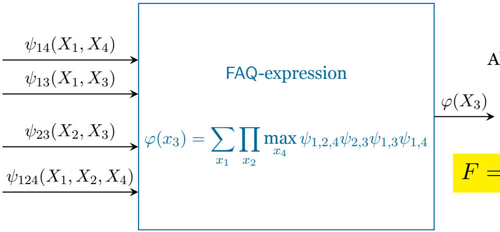
$$\mathcal{V} = \{1, 2, 3, 4\}$$
$$\mathcal{E} = \{\{1, 4\}, \{1, 3\}, \{2, 3\}, \{1, 2, 4\}\}$$

- 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

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

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

Function Aggregate Query: the Input



All functions have the same range **D**

$F = \{3\}$

- n ***variables*** $X_1, \ldots, X_n$
- 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

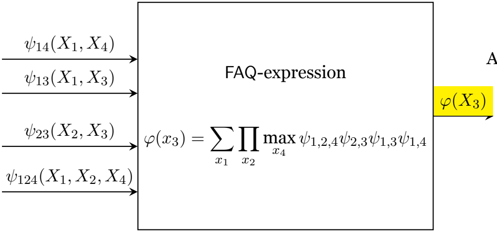
$$\psi_S : \prod_{i \in S} \mathsf{Dom}(X_i) \to \mathbf{D}$$

$\uparrow$

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

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

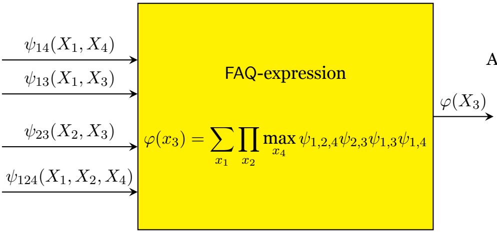
Function Aggregate Query: the Output



All functions have the same range $\mathbf{D}$

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

Function Aggregate Query: the Output

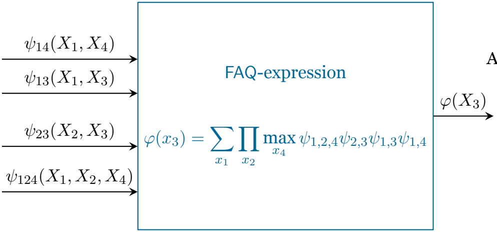


All functions have the same range $\mathbf{D}$

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

$$\varphi(\boldsymbol{x}_{[f]}) = \bigoplus^{(f+1)}_{x_{f+1} \in \mathsf{Dom}(X_{f+1})} \cdots \bigoplus^{(n-1)}_{x_{n-1} \in \mathsf{Dom}(X_{n-1})} \bigoplus^{(n)}_{x_n \in \mathsf{Dom}(X_n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\boldsymbol{x}_S)$$

Function Aggregate Query: the Output



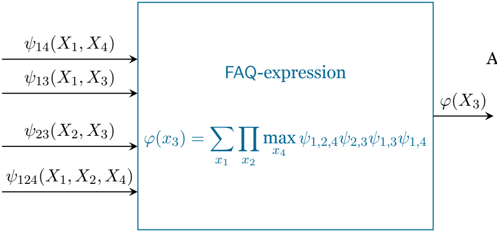
All functions have the same range **D**

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

$$\varphi(\boldsymbol{x}_{[f]}) = \bigoplus^{(f+1)}_{x_{f+1} \in \mathsf{Dom}(X_{f+1})} \cdots \bigoplus^{(n-1)}_{x_{n-1} \in \mathsf{Dom}(X_{n-1})} \bigoplus^{(n)}_{x_n \in \mathsf{Dom}(X_n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\boldsymbol{x}_S)$$

- Every aggregate $\bigoplus^{(i)}$, either

Function Aggregate Query: the Output



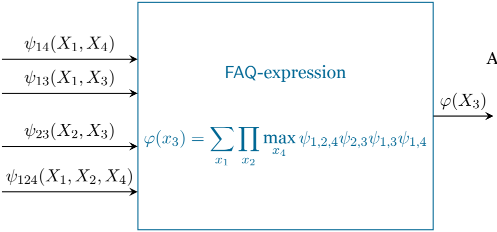
All functions have the same range $\mathbf{D}$

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

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

- Every aggregate $\bigoplus^{(i)}$, either
  - $\left(\mathbf{D}, \bigoplus^{(i)}, \bigotimes\right)$ is a *commutative semiring* (semiring agg)

Function Aggregate Query: the Output



All functions have the same range $\mathbf{D}$

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

$$\varphi(\boldsymbol{x}_{[f]}) = \bigoplus^{(f+1)}_{x_{f+1} \in \mathsf{Dom}(X_{f+1})} \cdots \bigoplus^{(n-1)}_{x_{n-1} \in \mathsf{Dom}(X_{n-1})} \bigoplus^{(n)}_{x_n \in \mathsf{Dom}(X_n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\boldsymbol{x}_S)$$

- Every aggregate $\bigoplus^{(i)}$, either
  - $\left(\mathbf{D}, \bigoplus^{(i)}, \bigotimes\right)$ is a *commutative semiring* (semiring agg)
  - or $\bigoplus^{(i)} = \bigotimes$ (product agg)

SumProd = FAQ-SS *without* free variables

- SumProd: compute the *constant*

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

SumProd = FAQ-SS *without* free variables

- SumProd: compute the *constant*

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

- Where $(\mathbf{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ 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 $\mathbf{0}$	$\mathbf{0} \otimes e = e \otimes \mathbf{0} = \mathbf{0}$
Distributive law	$a \otimes b \oplus a \otimes c = a \otimes (b \oplus c)$

SumProd = FAQ-SS *without* free variables

- SumProd: compute the *constant*

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

- Where $(\mathbf{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ 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 $\mathbf{0}$	$\mathbf{0} \otimes e = e \otimes \mathbf{0} = \mathbf{0}$
Distributive law	$a \otimes b \oplus a \otimes c = a \otimes (b \oplus c)$

- Common examples (there are many more!)

Boolean	$(\{\text{true}, \text{false}\}, \vee, \wedge, \text{false}, \text{true})$
sum-product	$(\mathbb{R}, +, \times, 0, 1)$
max-product	$(\mathbb{R}_+, \max, \times, 0, 1)$
min-plus	$(\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$
min-product	$(\mathbb{R}_{>0} \cup \{\infty\}, \min, \times, \infty, 1)$
set	$(2^U, \cup, \cap)$

Problem (CSP = Boolean Join Query)

is there a "truth assignment" x satisfying all constraints?

Problem (CSP = Boolean Join Query)

is there a "truth assignment" $\boldsymbol{x}$ satisfying all constraints?

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

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

Problem (CSP = Boolean Join Query)

is there a "truth assignment" x satisfying all constraints?

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

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

- constraint $\psi_S : \prod_{i \in S} \mathsf{Dom}(X_i) \to \{\mathsf{true}, \mathsf{false}\}$ with *support* S

Problem (CSP = Boolean Join Query)

is there a "truth assignment" $\boldsymbol{x}$ satisfying all constraints?

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

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

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

Problem (CSP = Boolean Join Query)

is there a "truth assignment" $\boldsymbol{x}$ satisfying all constraints?

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

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

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

Problem (#CSP = Join cardinality query)

count number of satisfying assignments x

Problem (#CSP = Join cardinality query)

count number of satisfying assignments x

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

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

Problem (#CSP = Join cardinality query)

count number of satisfying assignments x

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

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

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

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

Problem (#CSP = Join cardinality query)

count number of satisfying assignments $\boldsymbol{x}$

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

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

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

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

- count 1 only if $\boldsymbol{x}$ satisfies all constraints

Problem (#CSP = Join cardinality query)

count number of satisfying assignments x

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

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

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

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

- count 1 only if $\boldsymbol{x}$ 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 (i.e. GroupBy) Variables

Problem (FAQ-SS with free variables)

Compute the function

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

Problem (Conjunctive Query Evaluation – CQE)

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

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

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

Problem (DB Aggregate Query – count)

```
SELECT count(*)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

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

Problem (DB Aggregate Query – count)

```
SELECT count(*)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

- Sum-product semiring $(\mathbb{R}, +, \times, 0, 1)$,
- FAQ-SS instance:

Problem (DB Aggregate Query – count)

```
SELECT count(*)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

- Sum-product semiring $(\mathbb{R}, +, \times, 0, 1)$,
- FAQ-SS instance:
  - $\psi_R(a,b) = \mathbf{1}_{(a,b)\in R} := \begin{cases} 1 & (a,b) \in R \\ 0 & (a,b) \notin R \end{cases}$

Problem (DB Aggregate Query – count)

```
SELECT count(*)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

- Sum-product semiring $(\mathbb{R}, +, \times, 0, 1)$,
- FAQ-SS instance:
  - $\psi_R(a,b) = \mathbf{1}_{(a,b)\in R} := \begin{cases} 1 & (a,b) \in R \\ 0 & (a,b) \notin R \end{cases}$
  - $\psi_S(a,c) = \mathbf{1}_{(a,c)\in S}$

Problem (DB Aggregate Query – count)

```
SELECT count(*)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

- Sum-product semiring $(\mathbb{R}, +, \times, 0, 1)$,
- FAQ-SS instance:
  - $\psi_R(a,b) = \mathbf{1}_{(a,b)\in R} := \begin{cases} 1 & (a,b) \in R \\ 0 & (a,b) \notin R \end{cases}$
  - $\psi_S(a,c) = \mathbf{1}_{(a,c)\in S}$
  - $\psi_T(b,c) = \mathbf{1}_{(b,c)\in T}$

Problem (DB Aggregate Query – count)

```
SELECT count(*)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

- Sum-product semiring $(\mathbb{R}, +, \times, 0, 1)$,
- FAQ-SS instance:
  - $\psi_R(a,b) = \mathbf{1}_{(a,b)\in R} := \begin{cases} 1 & (a,b) \in R \\ 0 & (a,b) \notin R \end{cases}$
  - $\psi_S(a,c) = \mathbf{1}_{(a,c)\in S}$
  - $\psi_T(b,c) = \mathbf{1}_{(b,c)\in T}$
  - Compute the function

$$\varphi(a) = \sum_b \sum_c \psi_R(a,b)\psi_S(a,c)\psi_R(b,c)$$

Problem (DB Aggregate Query – sum)

```
SELECT sum(T.c)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

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

Problem (DB Aggregate Query – sum)

```
SELECT sum(T.c)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

- Sum-product semiring $(\mathbb{R}, +, \times, 0, 1)$,
- FAQ-SS instance:

Problem (DB Aggregate Query – sum)

```
SELECT sum(T.c)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

- Sum-product semiring $(\mathbb{R}, +, \times, 0, 1)$,
- FAQ-SS instance:
  - $\psi_R(a, b) = \mathbf{1}_{(a,b)\in R}$

Problem (DB Aggregate Query – sum)

```
SELECT sum(T.c)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

- Sum-product semiring $(\mathbb{R}, +, \times, 0, 1)$,
- FAQ-SS instance:
  - $\psi_R(a, b) = \mathbf{1}_{(a,b)\in R}$
  - $\psi_S(a, c) = \mathbf{1}_{(a,c)\in S}$

Problem (DB Aggregate Query – sum)

```
SELECT sum(T.c)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

- Sum-product semiring $(\mathbb{R}, +, \times, 0, 1)$,
- FAQ-SS instance:
  - $\psi_R(a, b) = \mathbf{1}_{(a,b)\in R}$
  - $\psi_S(a, c) = \mathbf{1}_{(a,c)\in S}$
  - $\psi_T(b, c) = \mathbf{1}_{(b,c)\in T}$

Problem (DB Aggregate Query – sum)

```
SELECT sum(T.c)
FROM R, S, T
WHERE R.a=S.a AND R.b=T.b AND S.c=T.c
GROUP BY R.a;
```

- Sum-product semiring $(\mathbb{R}, +, \times, 0, 1)$,
- FAQ-SS instance:
  - $\psi_R(a, b) = \mathbf{1}_{(a,b)\in R}$
  - $\psi_S(a, c) = \mathbf{1}_{(a,c)\in S}$
  - $\psi_T(b, c) = \mathbf{1}_{(b,c)\in T}$
  - Compute the function

$$\varphi(a) = \sum_b \sum_c \psi_R(a, b)\psi_S(a, c)\psi_R(b, c)\ c$$

Problem (Matrix Chain Multiplication – MCM)

Let $\boldsymbol{A}_i = (a^{(i)}_{x,y})$, *compute*

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

Problem (Matrix Chain Multiplication – MCM)

Let $\boldsymbol{A}_i = (a^{(i)}_{x,y})$, *compute*

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

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

Problem (Matrix Chain Multiplication – MCM)

Let $\boldsymbol{A}_i = (a^{(i)}_{x,y})$, *compute*

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

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

Problem (Matrix Chain Multiplication – MCM)

Let $\boldsymbol{A}_i = (a^{(i)}_{x,y})$*, compute*

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

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

Problem (Matrix Chain Multiplication – MCM)

Let $\boldsymbol{A}_i = (a^{(i)}_{x,y})$, *compute*

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

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

Problem (Matrix Chain Multiplication – MCM)

Let $\boldsymbol{A}_i = (a^{(i)}_{x,y})$*, compute*

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

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

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

Inference in Probabilistic Graphical Models

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

Inference in Probabilistic Graphical Models

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

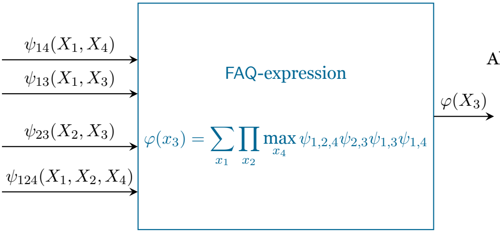
Inference in Probabilistic Graphical Models

- In PGMs, factors are also called *potential functions*
- Three typical inference/learning tasks on PGMs
  - Compute some marginal distribution
  - Compute $p(\boldsymbol{x}_A \mid \boldsymbol{x}_B)$
  - Compute $\text{argmax}_{\boldsymbol{x}_A} p(\boldsymbol{x}_A \mid \boldsymbol{x}_B)$ (e.g. MAP queries)
- These problems are *precisely* FAQ-SS 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?



All functions have the same range $\mathbf{D}$

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

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

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

Quantified Conjunctive Queries

Problem (QCQ with free variables)

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

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

Quantified Conjunctive Queries

Problem (QCQ with free variables)

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

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

- FAQ instance:
  - $(\{0,1\}, \{\max, \times\}, \times)$
  - Compute the function

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

where

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

Quantified Conjunctive Queries

Problem (#QCQ)

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

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

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

Quantified Conjunctive Queries

Problem (#QCQ)

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

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

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

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

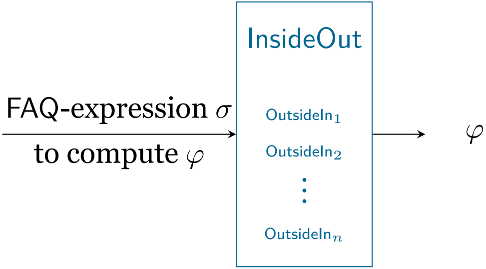
$$\varphi = \sum_{x_1} \cdots \sum_{x_f} \bigoplus_{x_{f+1} \in \{0,1\}}^{(f+1)} \cdots \bigoplus_{x_n \in \{0,1\}}^{(n)} \prod_{S \in \mathcal{E}} \psi_S(\boldsymbol{x}_S)$$

where

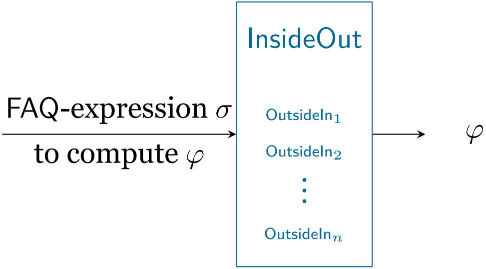
$$\oplus^{(i)} = \begin{cases} \max & \text{if } Q_i = \exists \\ \times & \text{if } Q_i = \forall \end{cases}$$

Table of Contents

The Algorithm

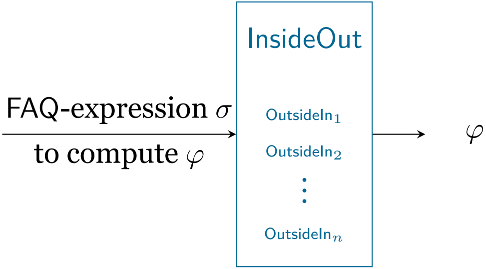


The Algorithm



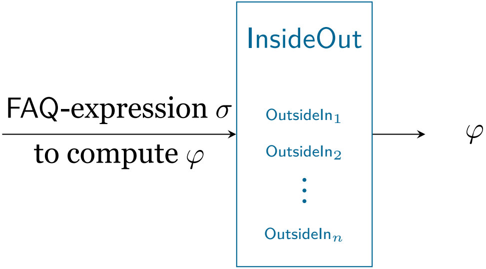
- InsideOut represents dynamic programming

The Algorithm



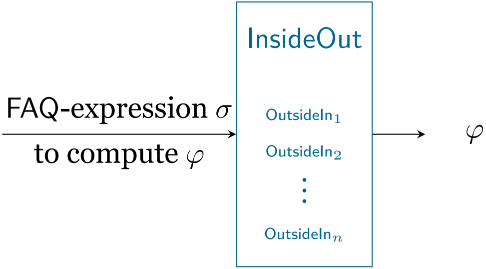
- InsideOut represents dynamic programming
  - is variable elimination with new twists
    - Variable Elimination (Zhang-Poole, JAIR 1996)
    - Bucket Elimination (Dechter, Artif. Intell. 1999)

The Algorithm



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

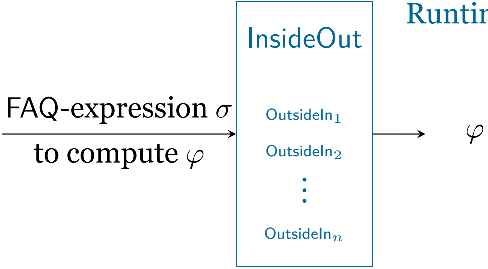
The Algorithm



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

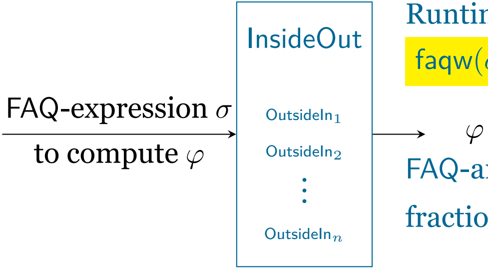
The Algorithm

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



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

The Algorithm



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

$\mathsf{faqw}(\sigma)$ is the "width" of σ

FAQ-analog of fractional hypertree width

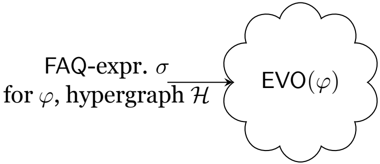
- InsideOut represents dynamic programming
  - is variable elimination with new twists
    - Variable Elimination (Zhang-Poole, JAIR 1996)
    - Bucket Elimination (Dechter, Artif. Intell. 1999)
- OutsideIn represents backtracking search
  - any worst-case optimal join algorithm, such as
    - NPRR (Ngo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

Roadmap

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

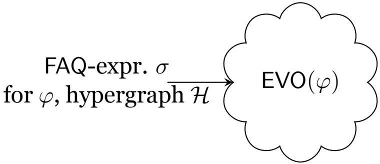
Roadmap

$\mathsf{EVO}(\varphi)$ = set of expressions
"semantically equivalent" to φ



Roadmap

$\mathsf{EVO}(\varphi)$ = set of expressions "semantically equivalent" to φ

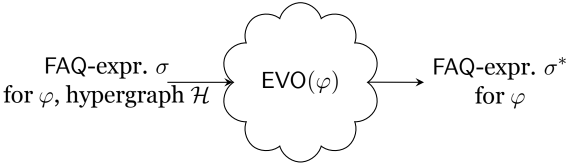


$$\varphi(x_3) = \sum_{x_2} \sum_{x_1} \sum_{x_5} \max_{x_4} \psi_{1,2,4}\psi_{2,3}\psi_{1,3}\psi_{1,4}\psi_{2,5}$$

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

Roadmap

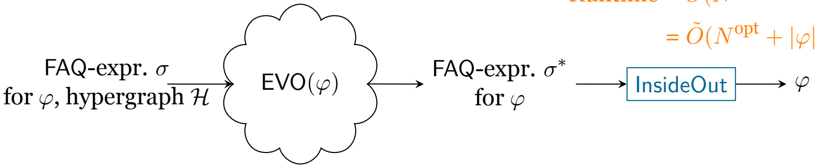
$$\sigma^* = \arg\min_{\tau \in \mathsf{EVO}(\varphi)} \mathsf{faqw}(\tau)$$



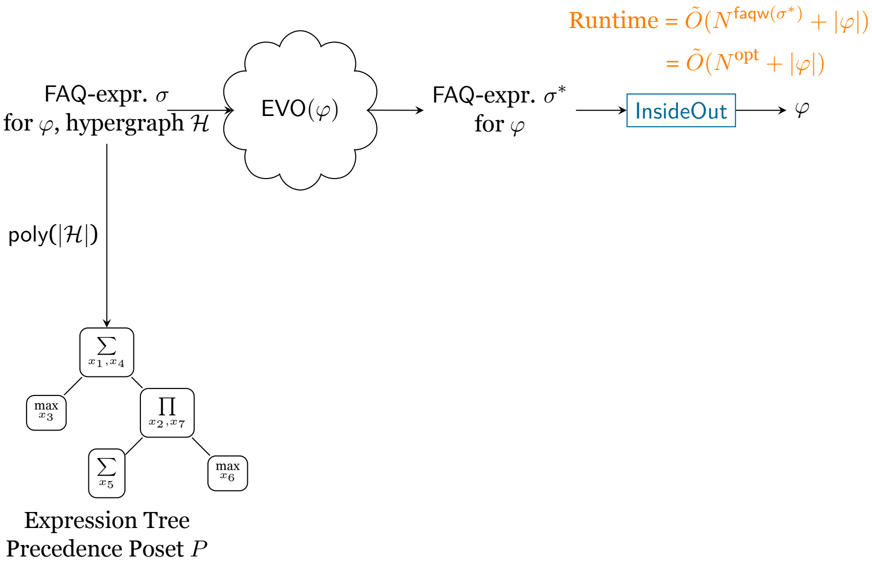
Roadmap

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

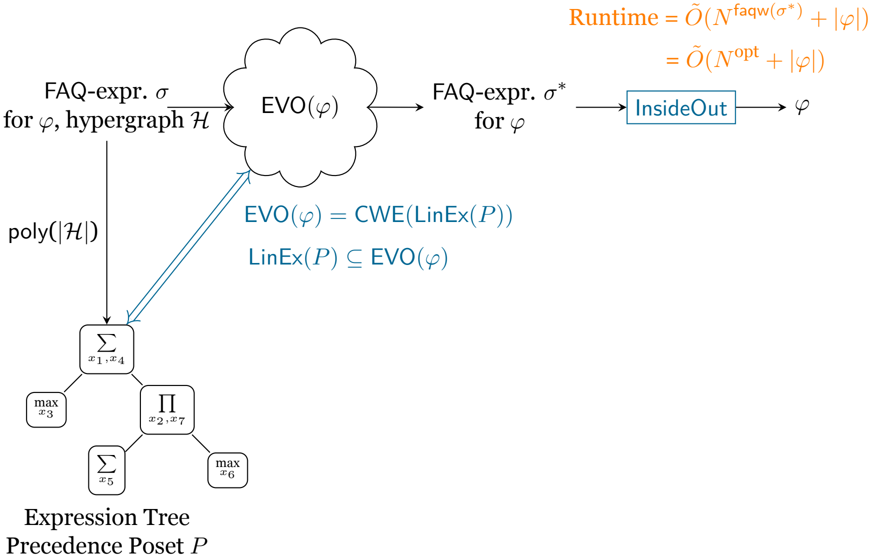
$$\text{Runtime} = \tilde{O}(N^{\mathsf{faqw}(\sigma^*)} + |\varphi|)$$
$$= \tilde{O}(N^{\text{opt}} + |\varphi|)$$



Roadmap

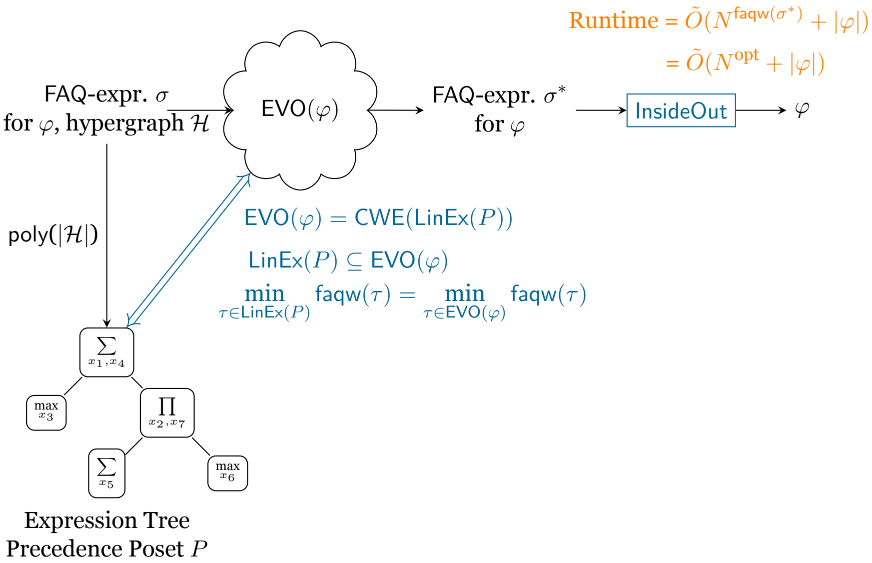

Runtime = $\tilde{O}(N^{\mathsf{faqw}(\sigma^*)} + |\varphi|)$
= $\tilde{O}(N^{\mathrm{opt}} + |\varphi|)$
FAQ-expr. σ
for φ, hypergraph $\mathcal{H}$
EVO(φ)
FAQ-expr. σ^*
for φ
InsideOut
φ
poly($|\mathcal{H}|$)
$\sum_{x_1, x_4}$
$\max_{x_3}$
$\prod_{x_2, x_7}$
$\sum_{x_5}$
$\max_{x_6}$
Expression Tree
Precedence Poset P

Roadmap



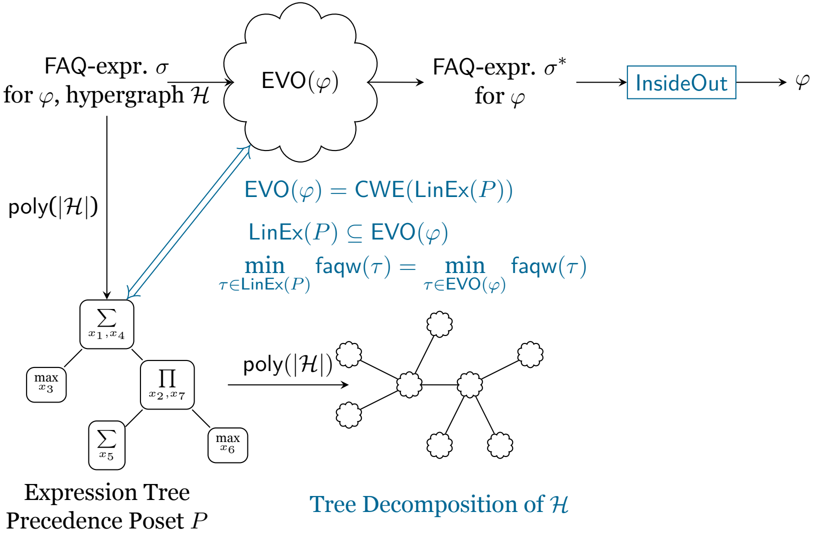
Runtime = $\tilde{O}(N^{\mathsf{faqw}(\sigma^*)} + |\varphi|)$
= $\tilde{O}(N^{\mathrm{opt}} + |\varphi|)$
FAQ-expr. σ
for φ, hypergraph $\mathcal{H}$
EVO(φ)
FAQ-expr. σ^*
for φ
InsideOut
φ
poly($|\mathcal{H}|$)
EVO(φ) = CWE(LinEx(P))
LinEx(P) $\subseteq$ EVO(φ)
$\sum_{x_1, x_4}$
$\max_{x_3}$
$\prod_{x_2, x_7}$
$\sum_{x_5}$
$\max_{x_6}$
Expression Tree
Precedence Poset P

Roadmap

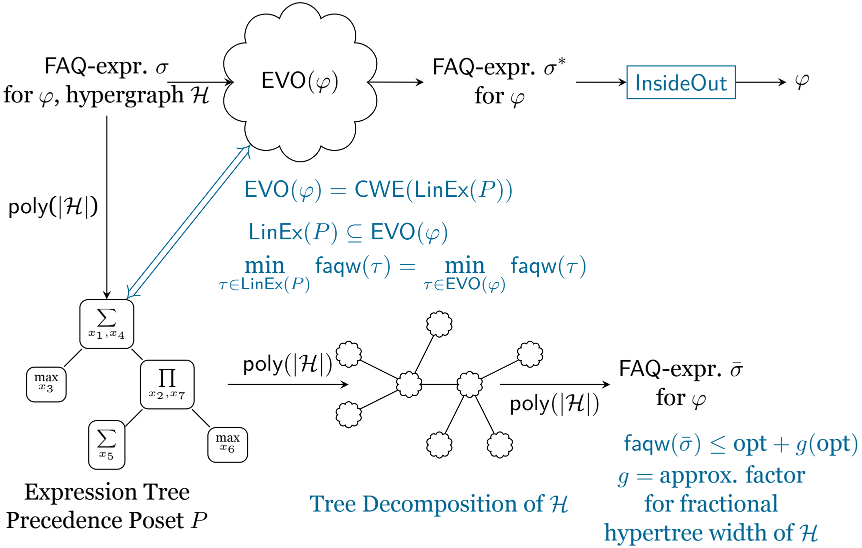


Runtime = $\tilde{O}(N^{\mathsf{faqw}(\sigma^*)} + |\varphi|)$
= $\tilde{O}(N^{\text{opt}} + |\varphi|)$
FAQ-expr. σ
for φ, hypergraph $\mathcal{H}$
$\mathsf{EVO}(\varphi)$
FAQ-expr. σ^*
for φ
InsideOut
φ
$\mathsf{poly}(|\mathcal{H}|)$
$\mathsf{EVO}(\varphi) = \mathsf{CWE}(\mathsf{LinEx}(P))$
$\mathsf{LinEx}(P) \subseteq \mathsf{EVO}(\varphi)$
$\min_{\tau \in \mathsf{LinEx}(P)} \mathsf{faqw}(\tau) = \min_{\tau \in \mathsf{EVO}(\varphi)} \mathsf{faqw}(\tau)$
$\sum_{x_1, x_4}$
$\max_{x_3}$
$\prod_{x_2, x_7}$
$\sum_{x_5}$
$\max_{x_6}$
Expression Tree
Precedence Poset P

Roadmap

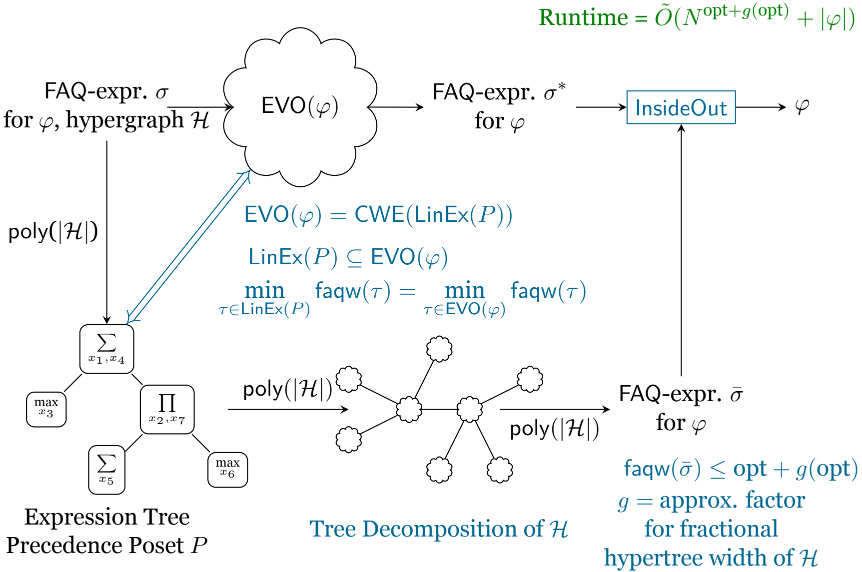

FAQ-expr. σ for φ, hypergraph $\mathcal{H}$
$\mathsf{EVO}(\varphi)$
FAQ-expr. σ^* for φ
InsideOut
φ
$\mathsf{poly}(|\mathcal{H}|)$
$\mathsf{EVO}(\varphi) = \mathsf{CWE}(\mathsf{LinEx}(P))$
$\mathsf{LinEx}(P) \subseteq \mathsf{EVO}(\varphi)$
$\min_{\tau \in \mathsf{LinEx}(P)} \mathsf{faqw}(\tau) = \min_{\tau \in \mathsf{EVO}(\varphi)} \mathsf{faqw}(\tau)$
$\sum_{x_1, x_4}$
$\max_{x_3}$
$\prod_{x_2, x_7}$
$\sum_{x_5}$
$\max_{x_6}$
$\mathsf{poly}(|\mathcal{H}|)$
Expression Tree
Precedence Poset P
Tree Decomposition of $\mathcal{H}$

Roadmap



FAQ-expr. σ
for φ, hypergraph $\mathcal{H}$
EVO(φ)
FAQ-expr. σ^*
for φ
InsideOut
φ
poly($|\mathcal{H}|$)
EVO(φ) = CWE(LinEx(P))
LinEx(P) $\subseteq$ EVO(φ)
$\min_{\tau \in \mathsf{LinEx}(P)} \mathsf{faqw}(\tau) = \min_{\tau \in \mathsf{EVO}(\varphi)} \mathsf{faqw}(\tau)$
$\sum_{x_1, x_4}$
$\max_{x_3}$
$\prod_{x_2, x_7}$
$\sum_{x_5}$
$\max_{x_6}$
poly($|\mathcal{H}|$)
poly($|\mathcal{H}|$)
FAQ-expr. $\bar{\sigma}$
for φ
$\mathsf{faqw}(\bar{\sigma}) \leq \text{opt} + g(\text{opt})$
g = approx. factor
for fractional
hypertree width of $\mathcal{H}$
Expression Tree
Precedence Poset P
Tree Decomposition of $\mathcal{H}$

Roadmap



Runtime = $\tilde{O}(N^{\text{opt}+g(\text{opt})} + |\varphi|)$
FAQ-expr. σ for φ, hypergraph $\mathcal{H}$
EVO(φ)
FAQ-expr. σ^* for φ
InsideOut
φ
poly($|\mathcal{H}|$)
EVO(φ) = CWE(LinEx(P))
LinEx(P) $\subseteq$ EVO(φ)
$\min_{\tau \in \text{LinEx}(P)} \text{faqw}(\tau) = \min_{\tau \in \text{EVO}(\varphi)} \text{faqw}(\tau)$
$\sum_{x_1, x_4}$
$\max_{x_3}$
$\prod_{x_2, x_7}$
$\sum_{x_5}$
$\max_{x_6}$
poly($|\mathcal{H}|$)
poly($|\mathcal{H}|$)
FAQ-expr. $\bar{\sigma}$ for φ
faqw($\bar{\sigma}$) $\leq$ opt + g(opt)
g = approx. factor for fractional hypertree width of $\mathcal{H}$
Expression Tree Precedence Poset P
Tree Decomposition of $\mathcal{H}$

Some Corollaries

Problem	Previous Algo.	InsideOut
#QCQ	No non-trivial algo	$\tilde{O}(N^{\mathsf{faqw}(\varphi)} + \lvert\varphi\rvert)$
QCQ	$\tilde{O}(N^{\mathsf{PW}(\mathcal{H})} + \lvert\varphi\rvert)$ Chen-Dalmau (LICS 2012)	$\tilde{O}(N^{\mathsf{faqw}(\varphi)} + \lvert\varphi\rvert)$ $\mathsf{faqw}(\varphi) \lneqq \mathsf{PW}(\varphi)$
#CQ	$\tilde{O}(N^{\mathsf{DM}(\mathcal{H})} + \lvert\varphi\rvert)$ Durand-Mengel (ICDT 2013)	$\tilde{O}(N^{\mathsf{faqw}(\varphi)} + \lvert\varphi\rvert)$ $\mathsf{DM}(\mathcal{H}) = \mathsf{faqw}(\varphi)$
Joins	$\tilde{O}\left(N^{\mathsf{fhtw}(\mathcal{H})} + \lvert\varphi\rvert\right)$ Grohe-Marx (SODA'06)	$\tilde{O}\left(N^{\mathsf{faqw}(\varphi)} + \lvert\varphi\rvert\right)$ $\mathsf{fhtw}(\mathcal{H}) = \mathsf{faqw}(\varphi)$
Marginal Distrib. MAP query	$\tilde{O}(N^{\mathsf{htw}(\varphi)} + \lvert\varphi\rvert)$ $\tilde{O}(N^{\mathsf{htw}(\varphi)} + \lvert\varphi\rvert)$ Kask et al. (Artif. Intel. 2005)	$\tilde{O}(N^{\mathsf{faqw}(\varphi)} + \lvert\varphi\rvert)$ $\tilde{O}(N^{\mathsf{faqw}(\varphi)} + \lvert\varphi\rvert)$ $\mathsf{faqw}(\varphi) \lneqq \mathsf{htw}(\varphi)$
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

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

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 $\langle \boldsymbol{x}_F, \varphi[\boldsymbol{x}_F] \rangle$
  but only store tuples for which $\varphi[\boldsymbol{x}_F] \neq \mathbf{0}$
- 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 $\langle \boldsymbol{x}_F, \varphi[\boldsymbol{x}_F] \rangle$ but only store tuples for which $\varphi[\boldsymbol{x}_F] \neq \mathbf{0}$
- Called the listing representation

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

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

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 $\langle \boldsymbol{x}_F, \varphi[\boldsymbol{x}_F] \rangle$
  but only store tuples for which $\varphi[\boldsymbol{x}_F] \neq \mathbf{0}$
- Called the listing representation

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

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

- Other representations $\Rightarrow$ different sets of results

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 $\langle \boldsymbol{x}_F, \varphi[\boldsymbol{x}_F] \rangle$ but only store tuples for which $\varphi[\boldsymbol{x}_F] \neq \mathbf{0}$
- Called the listing representation

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

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

- Other representations $\Rightarrow$ different sets of results
  - SAT $\in$ **P** for β-acyclic queries

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 $\langle \boldsymbol{x}_F, \varphi[\boldsymbol{x}_F] \rangle$ but only store tuples for which $\varphi[\boldsymbol{x}_F] \neq \mathbf{0}$
- Called the listing representation

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

ID	Name	Salary
1234	Dan Suciu	$400,000
5678	Current Hung Ngo	$100,000
9999	Future Hung Ngo	$1,000,000

- Other representations $\Rightarrow$ different sets of results
  - SAT $\in$ **P** for β-acyclic queries
  - Minesweeper and Tetris join algorithms

Notation: conditional factor

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

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

where

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

Notation: conditional factor

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

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

where

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

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

Notation: conditional factor

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

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

where

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

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

OutsideIn for FAQ-SS *without* free variables

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

OutsideIn for FAQ-SS *without* free variables

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

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

OutsideIn for FAQ-SS *without* free variables

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

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

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

OutsideIn for FAQ-SS *without* free variables

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

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

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

Runtime $O(D^n)$ ↩ $x_1 \in \mathsf{Dom}(X_1)$

OutsideIn for FAQ-SS *without* free variables

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

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

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

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

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

OutsideIn for FAQ-SS *without* free variables

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

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

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

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

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

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

OutsideIn = Backtracking Search

OutsideIn = Backtracking Search

- Question: How good is OutsideIn?

OutsideIn = Backtracking Search

- Question: How good is OutsideIn?
- Answer: Worst-case optimal for joins!

OutsideIn = Backtracking Search

- Question: How good is OutsideIn?
- 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) AGM-bound
  - Ngo, Ré, Porat, Rudra (PODS'12) NPRR algorithm

OutsideIn = Backtracking Search

- Question: How good is OutsideIn?
- 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

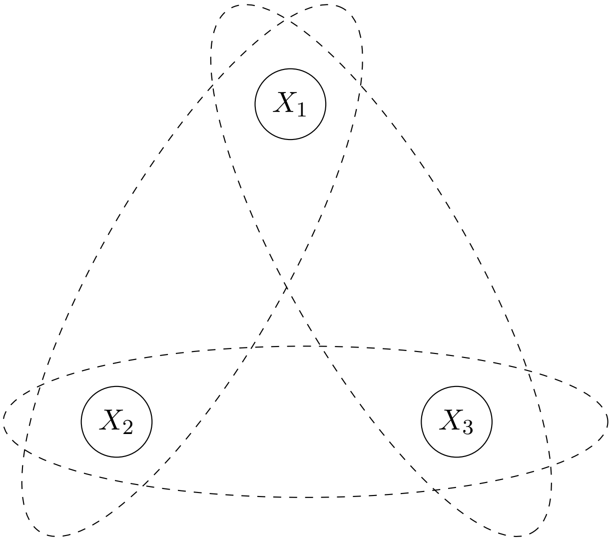
OutsideIn = Backtracking Search

- Question: How good is OutsideIn?
- 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

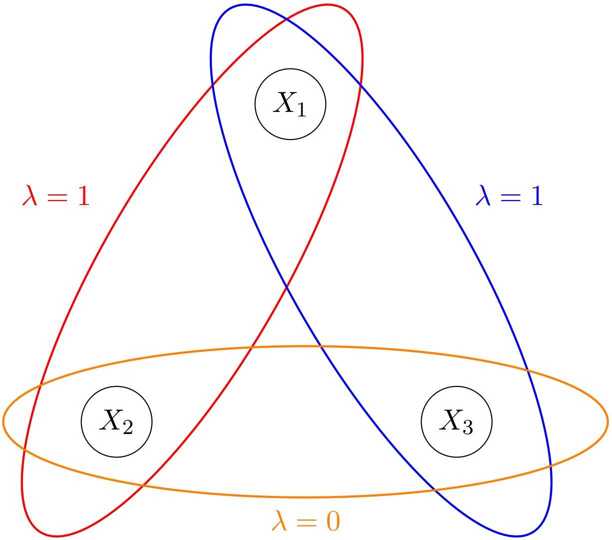
OutsideIn = Backtracking Search

- **Question:** How good is OutsideIn?
- **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
  - Abo Khamis, Ngo, Suciu (PODS'16) — PANDA

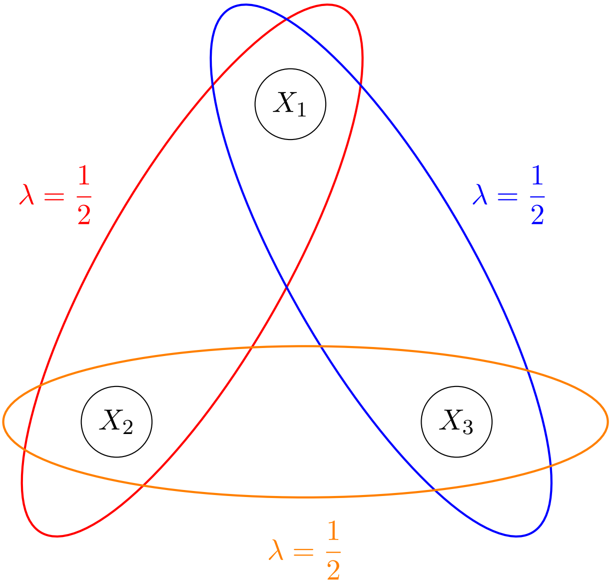
Background: fractional edge cover of $\mathcal{H}$



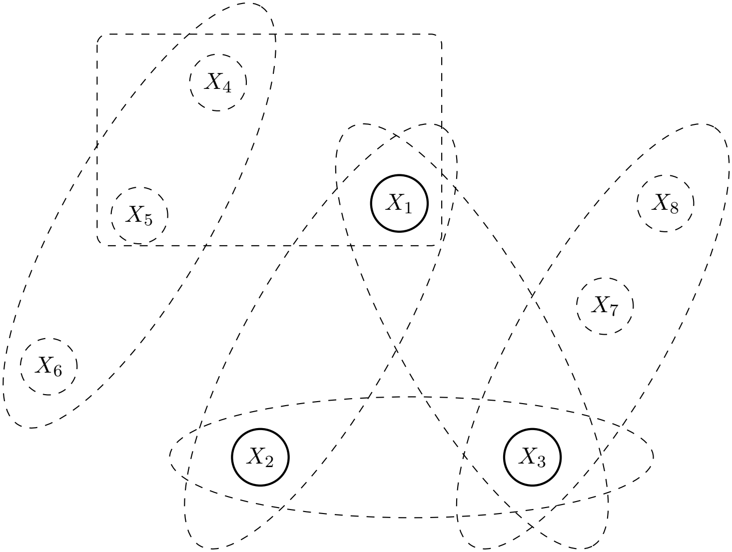
Background: fractional edge cover of $\mathcal{H}$



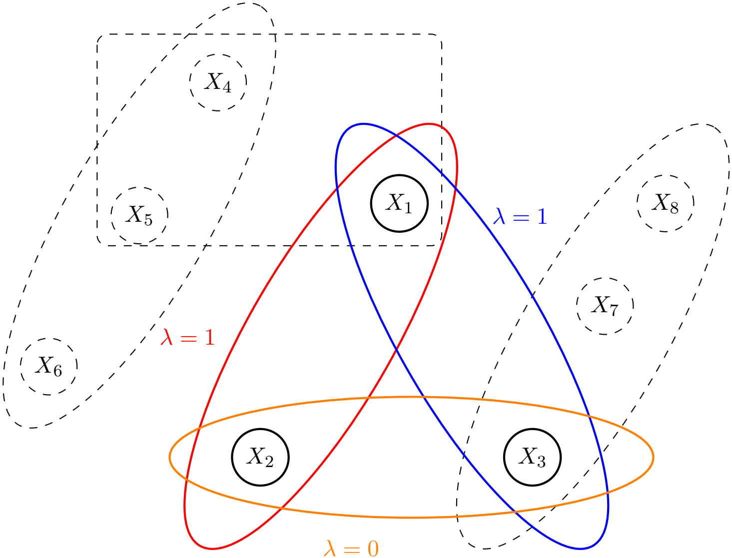
Background: fractional edge cover of $\mathcal{H}$



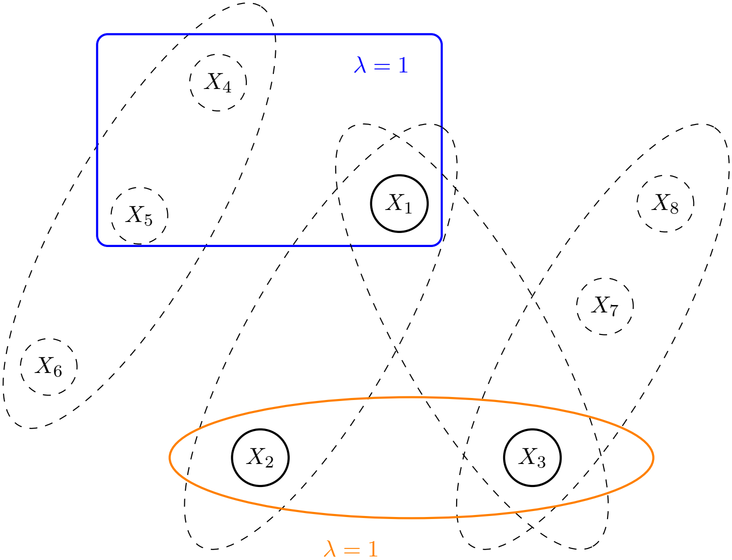
Background: fractional edge cover of $B = \{1, 2, 3\}$



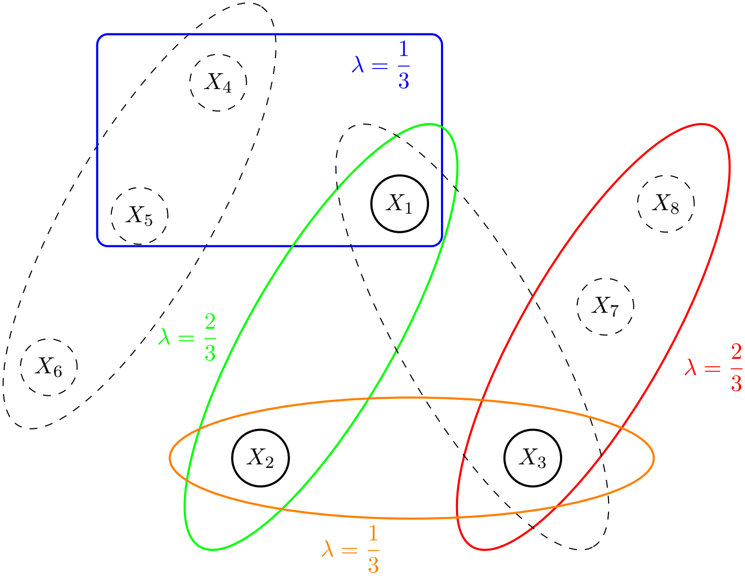
Background: fractional edge cover of $B = \{1, 2, 3\}$



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.

$$\begin{aligned} \sum_{S : v \in S} \lambda_S &\geq 1, \ \forall v \in B \\ \lambda_S &\geq 0, \ \forall S \in \mathcal{E}. \end{aligned}$$

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

Pros of OutsideIn

Theorem (Runtime of OutsideIn)

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

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

Pros of OutsideIn

Theorem (Runtime of OutsideIn)

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

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

Pros of OutsideIn

Theorem (Runtime of OutsideIn)

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

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

AGM-*bound*

- OutsideIn is worst-case optimal for joins
- Very simple to implement

Pros of OutsideIn

Theorem (Runtime of OutsideIn)

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

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

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

Pros of OutsideIn

Theorem (Runtime of OutsideIn)

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

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

- OutsideIn 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 OutsideIn

Theorem (Runtime of OutsideIn)

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

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

AGM-*bound*

- OutsideIn 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 OutsideIn

Theorem (Runtime of OutsideIn)

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

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

- OutsideIn 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* 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* $\mathsf{AGM}(\mathcal{H})$
  - $3/2$ 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* 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* $\mathsf{AGM}(\mathcal{H})$
  - $3/2$ 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* $\mathsf{AGM}(\mathcal{H})$
  - $3/2$ 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* $\mathsf{AGM}(\mathcal{H})$

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* $\mathsf{AGM}(\mathcal{H})$
  - $3/2$ 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* $\mathsf{AGM}(\mathcal{H})$
  - $4/3$ 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* $\mathsf{AGM}(\mathcal{H})$
  - $3/2$ 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* $\mathsf{AGM}(\mathcal{H})$
  - $4/3$ 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

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

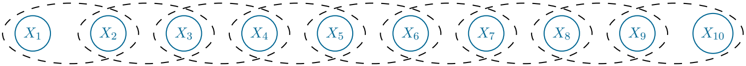
  on any (α-)acyclic query

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 $O(N^k)$.

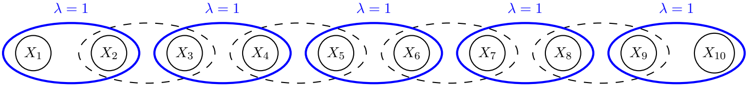


Cons of OutsideIn

- 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 $O(N^k)$.

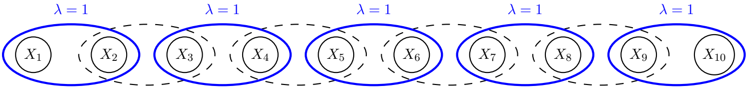


Cons of OutsideIn

- 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 $O(N^k)$.



- More generally, horrible whenever fhtw < AGM-bound.

Table of Contents

Example: $\varphi() = \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d) \times b$

$$\varphi = \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d)b$$

Example: $\varphi() = \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d) \times b$

$$
\begin{aligned}
\varphi &= \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d)b \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c) \sum_d T(c,d)
\end{aligned}
$$

Example: $\varphi() = \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d) \times b$

$$\begin{aligned}
\varphi &= \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d)b \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c) \sum_d T(c,d) \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c) F(c)
\end{aligned}$$

Example: $\varphi() = \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d) \times b$

$$
\begin{aligned}
\varphi &= \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d)b \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c) \sum_d T(c,d) \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c)F(c) \\
&= \sum_a \sum_b b \cdot R(a,b) \sum_c S(b,c)F(c)
\end{aligned}
$$

Example: $\varphi() = \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d) \times b$

$$
\begin{aligned}
\varphi &= \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d)b \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c) \sum_d T(c,d) \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c)F(c) \\
&= \sum_a \sum_b b \cdot R(a,b) \sum_c S(b,c)F(c) \\
&= \sum_a \sum_b b \cdot R(a,b)G(b)
\end{aligned}
$$

Example: $\varphi() = \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d) \times b$

$$\begin{aligned}
\varphi &= \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d)b \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c) \sum_d T(c,d) \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c)F(c) \\
&= \sum_a \sum_b b \cdot R(a,b) \sum_c S(b,c)F(c) \\
&= \sum_a \sum_b b \cdot R(a,b)G(b) \\
&= \sum_a \sum_b b \cdot R(a,b)G(b)
\end{aligned}$$

Example: $\varphi() = \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d) \times b$

$$
\begin{aligned}
\varphi &= \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d)b \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c) \sum_d T(c,d) \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c)F(c) \\
&= \sum_a \sum_b b \cdot R(a,b) \sum_c S(b,c)F(c) \\
&= \sum_a \sum_b b \cdot R(a,b)G(b) \\
&= \sum_a \sum_b b \cdot R(a,b)G(b) \\
&= \sum_a H(a).
\end{aligned}
$$

Example: $\varphi() = \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d) \times b$

$$
\begin{aligned}
\varphi &= \sum_a \sum_b \sum_c \sum_d R(a,b)S(b,c)T(c,d)b \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c) \sum_d T(c,d) \\
&= \sum_a \sum_b \sum_c b \cdot R(a,b)S(b,c)F(c) \\
&= \sum_a \sum_b b \cdot R(a,b) \sum_c S(b,c)F(c) \\
&= \sum_a \sum_b b \cdot R(a,b)G(b) \\
&= \sum_a \sum_b b \cdot R(a,b)G(b) \\
&= \sum_a H(a).
\end{aligned}
$$

Runtime: $\tilde{O}(|R| + |S| + |T|)$.

FAQ-SS *without* free variables

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

FAQ-SS *without* free variables

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

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

FAQ-SS *without* free variables

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

FAQ-SS *without* free variables

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

FAQ-SS *without* free variables

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

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

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

↑

$\otimes$ distributes over $\oplus$

FAQ-SS *without* free variables

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

FAQ-SS *without* free variables

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

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

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

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

FAQ-SS *without* free variables

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

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

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

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

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

FAQ-SS *without* free variables

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

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

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

FAQ-SS *without* free variables

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

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

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

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

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

FAQ-SS *without* free variables

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

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

FAQ-SS *without* free variables

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

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

New instance of FAQ-SS

$$
\begin{aligned}
\mathcal{H}_{n-1} &= ([n-1], \mathcal{E}_{n-1}) \\
\mathcal{E}_{n-1} &= \mathcal{E} + (U_n - \{n\}) - \{S \mid n \in S\}
\end{aligned}
$$

Naïve InsideOut = Variable Elimination + OutsideIn

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

Naïve InsideOut = Variable Elimination + OutsideIn

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

Naïve InsideOut = Variable Elimination + OutsideIn

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

$$\uparrow$$

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

Naïve InsideOut = Variable Elimination + OutsideIn

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

$\uparrow$

But how do we compute this? $\psi_{U_n - \{n\}}(\boldsymbol{x}_{U_n - \{n\}})$

Naïve InsideOut = Variable Elimination + OutsideIn

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

$$\uparrow$$

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

Run the OutsideIn algorithm!

Naïve InsideOut = Variable Elimination + OutsideIn

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

$\uparrow$

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

Run the OutsideIn algorithm!

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

Naïve InsideOut = Variable Elimination + OutsideIn

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

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 + OutsideIn

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

Theorem (Runtime of Naïve InsideOut)

For a given variable ordering $v_1, \ldots, v_n$, 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)

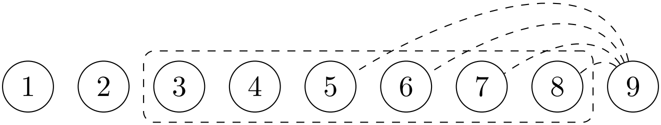
For a "good" variable ordering, PGM inference can be done in time $O(mn^2 D^{w+1})$ where w = tree-width$(\mathcal{H})$ = $\mathrm{tw}(\mathcal{H})$.

Pros and Cons of Naïve InsideOut

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

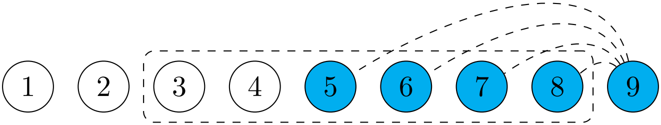
Pros and Cons of Naïve InsideOut

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



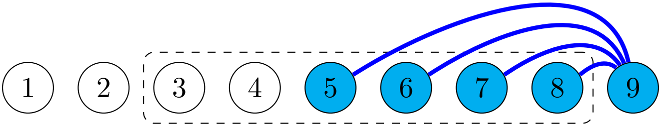
Pros and Cons of Naïve InsideOut

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



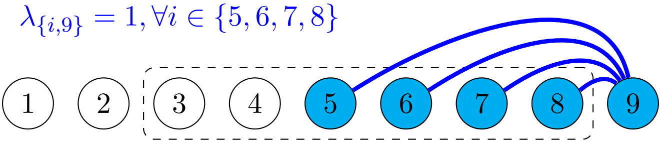
Pros and Cons of Naïve InsideOut

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



Pros and Cons of Naïve InsideOut

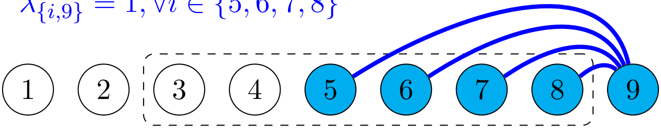
- The good: $O(kN)$-time for k-path query!
- The bad: consider the following (part of a) graph



Pros and Cons of Naïve InsideOut

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

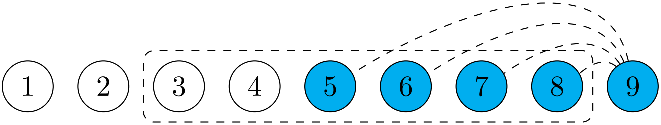
$\lambda_{\{i,9\}} = 1, \forall i \in \{5,6,7,8\}$



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

Pros and Cons of Naïve InsideOut

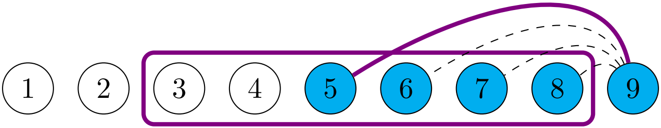
- The good: $O(kN)$-time for k-path query!
- The bad: consider the following (part of a) graph



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

Pros and Cons of Naïve InsideOut

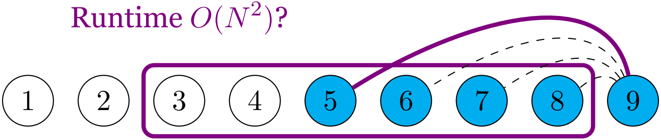
- The good: $O(kN)$-time for k-path query!
- The bad: consider the following (part of a) graph



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

Pros and Cons of Naïve InsideOut

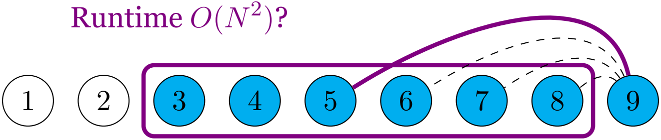
- The good: $O(kN)$-time for k-path query!
- The bad: consider the following (part of a) graph



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

Pros and Cons of Naïve InsideOut

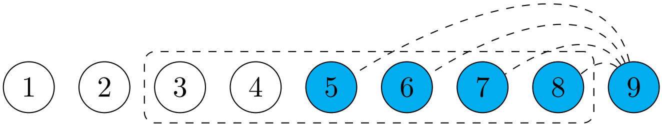
- The good: $O(kN)$-time for k-path query!
- The bad: consider the following (part of a) graph



- Say $|\psi_S| = N$, then runtime $O(N^4)$.
- Good idea, but naïve application increases U_9

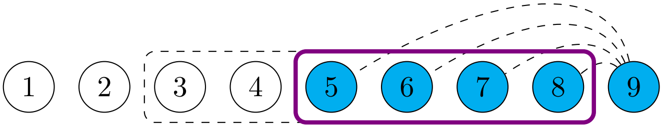
Indicator projection

- Projection onto U_9:



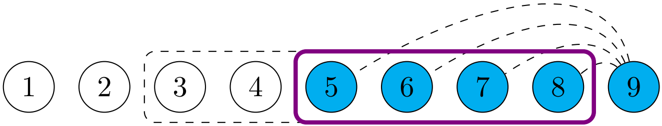
Indicator projection

- Projection onto U_9:



Indicator projection

- Projection onto U_9:



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

$$\pi_U \psi_S : \prod_{i \in S \cap U} \mathsf{Dom}(X_i) \to \{\mathbf{0}, \mathbf{1}\},$$

where

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

Not so naïve InsideOut Example

$$\varphi = \sum_{a,b,c,d,e,f} R(a,b)S(a,c)T(b,c,d,e)W(e,f)V(d,f)$$

Not so naïve InsideOut Example

$$\varphi = \sum_{a,b,c,d,e,f} R(a,b)S(a,c)T(b,c,d,e)W(e,f)V(d,f)$$

$$= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e) \underbrace{\sum_{f} \pi_{de}T(d,e) W(e,f)V(d,f)}_{\tilde{O}(N^{3/2})}$$

Not so naïve InsideOut Example

$$
\begin{aligned}
\varphi &= \sum_{a,b,c,d,e,f} R(a,b)S(a,c)T(b,c,d,e)W(e,f)V(d,f) \\
&= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e) \underbrace{\sum_{f} \pi_{de}T(d,e)\, W(e,f)V(d,f)}_{\tilde{O}(N^{3/2})} \\
&= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e)M_1(d,e)
\end{aligned}
$$

Not so naïve InsideOut Example

$$\varphi = \sum_{a,b,c,d,e,f} R(a,b)S(a,c)T(b,c,d,e)W(e,f)V(d,f)$$

$$= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e) \underbrace{\sum_{f} \pi_{de}T(d,e)\, W(e,f)V(d,f)}_{\tilde{O}(N^{3/2})}$$

$$= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e)M_1(d,e)$$

$$= \sum_{a,b,c} R(a,b)S(a,c) \sum_{d,e} \pi_b R(b) \pi_S(c)\, T(b,c,d,e)M_1(d,e)$$

Not so naïve InsideOut Example

$$
\begin{aligned}
\varphi &= \sum_{a,b,c,d,e,f} R(a,b)S(a,c)T(b,c,d,e)W(e,f)V(d,f) \\
&= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e) \underbrace{\sum_{f} \pi_{de}T(d,e)\, W(e,f)V(d,f)}_{\tilde{O}(N^{3/2})} \\
&= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e)M_1(d,e) \\
&= \sum_{a,b,c} R(a,b)S(a,c) \sum_{d,e} \pi_b R(b)\pi_S(c)\, T(b,c,d,e)M_1(d,e) \\
&= \sum_{a,b,c} R(a,b)S(a,c)M_2(b,c)
\end{aligned}
$$

Not so naïve InsideOut Example

$$\begin{aligned}
\varphi &= \sum_{a,b,c,d,e,f} R(a,b)S(a,c)T(b,c,d,e)W(e,f)V(d,f) \\
&= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e) \underbrace{\sum_{f} \pi_{de}T(d,e)\, W(e,f)V(d,f)}_{\tilde{O}(N^{3/2})} \\
&= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e)M_1(d,e) \\
&= \sum_{a,b,c} R(a,b)S(a,c) \sum_{d,e} \pi_b R(b)\pi_S(c)\, T(b,c,d,e)M_1(d,e) \\
&= \sum_{a,b,c} R(a,b)S(a,c)M_2(b,c) \\
&= \underbrace{\sum_{a,b,c} R(a,b)S(a,c)M_2(b,c)}_{\tilde{O}(N^{3/2})}
\end{aligned}$$

InsideOut = VE + indicator-projections + OutsideIn

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

InsideOut = VE + indicator-projections + OutsideIn

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

InsideOut = VE + indicator-projections + OutsideIn

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

InsideOut = VE + indicator-projections + OutsideIn

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

Time $\leq mn\mathsf{AGM}(U_n)$

InsideOut = VE + indicator-projections + OutsideIn

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

Theorem (Runtime of InsideOut)

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

$$mn \sum_{k=1}^{n} \mathsf{AGM}(U_k) \leq mn \sum_{k=1}^{n} N^{\rho^*_{\mathcal{H}}(U_k)} \leq mn^2 N^{\max_k \rho^*_{\mathcal{H}}(U_k)}$$

InsideOut = VE + indicator-projections + OutsideIn

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

Theorem (Runtime of InsideOut)

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

$$mn \sum_{k=1}^{n} \mathsf{AGM}(U_k) \leq mn \sum_{k=1}^{n} N^{\rho^*_{\mathcal{H}}(U_k)} \leq mn^2 N^{\max_k \rho^*_{\mathcal{H}}(U_k)}$$

Corollary (New (textbook?) result)

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

InsideOut = VE + indicator-projections + OutsideIn

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

Theorem (Runtime of InsideOut)

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

$$mn \sum_{k=1}^{n} \mathsf{AGM}(U_k) \leq mn \sum_{k=1}^{n} N^{\rho^*_{\mathcal{H}}(U_k)} \leq mn^2 N^{\max_k \rho^*_{\mathcal{H}}(U_k)}$$

Corollary (Grohe-Marx (SODA'06))

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

InsideOut = VE + indicator-projections + OutsideIn

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

Theorem (Runtime of InsideOut)

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

$$mn \sum_{k=1}^{n} \mathsf{AGM}(U_k) \leq mn \sum_{k=1}^{n} N^{\rho^*_{\mathcal{H}}(U_k)} \leq mn^2 N^{\max_k \rho^*_{\mathcal{H}}(U_k)}$$

Corollary (Grohe-Marx (SODA'06))

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

InsideOut = VE + indicator-projections + OutsideIn

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

Theorem (Runtime of InsideOut)

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

$$mn \sum_{k=1}^{n} \mathsf{AGM}(U_k) \leq mn \sum_{k=1}^{n} N^{\rho^*_{\mathcal{H}}(U_k)} \leq mn^2 N^{\max_k \rho^*_{\mathcal{H}}(U_k)}$$

Corollary (New?)

Join cardinality and any aggregate query (over a conjunction) can be computed in time $\tilde{O}(N^{\mathsf{fhtw}(\mathcal{H})})$

InsideOut = VE + indicator-projections + OutsideIn

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

Theorem (Runtime of InsideOut)

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

$$mn \sum_{k=1}^{n} \text{AGM}(U_k) \leq mn \sum_{k=1}^{n} N^{\rho^*_{\mathcal{H}}(U_k)} \leq mn^2 N^{\max_k \rho^*_{\mathcal{H}}(U_k)}$$

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

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

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

InsideOut for general FAQ

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Table of Contents

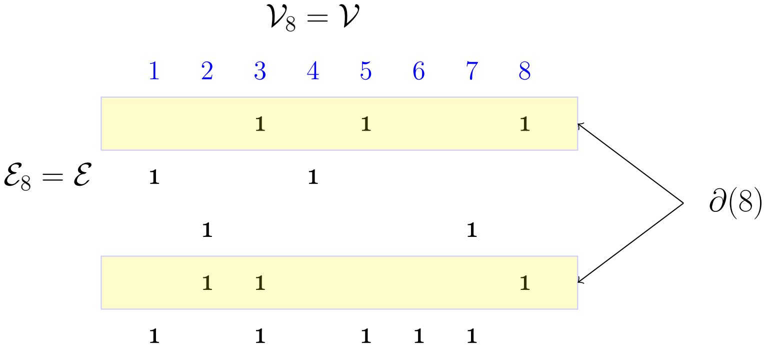
Variable ordering and support set sequence

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

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

1	2	3	4	5	6	7	8
		1		**1**			**1**
1			**1**				
	1					**1**	
	1	**1**					**1**
1		**1**		**1**	**1**	**1**	

Variable ordering and support set sequence



Variable ordering and support set 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	
U_8		1	1		1			1

$\partial(8)$

Variable ordering and support set sequence

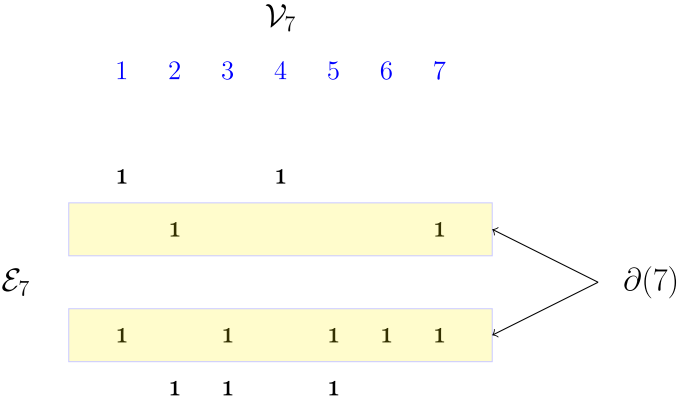
	$\mathcal{V}_7$						
	1	2	3	4	5	6	7
	1			**1**			
		1					**1**
$\mathcal{E}_7$							
	1		**1**		**1**	**1**	**1**
$U_8 - \{8\}$		**1**	**1**		**1**		

Variable ordering and support set sequence

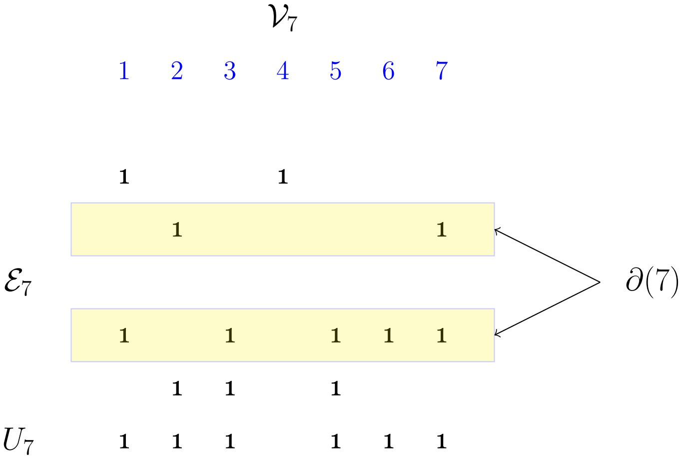
$\mathcal{V}_7$

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

Variable ordering and support set sequence



Variable ordering and support set sequence



Variable ordering and support set 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 support set 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 support set sequence

$\mathcal{V}_6$

	1	2	3	4	5	6	
	1			**1**			
$\mathcal{E}_6$							
		1	**1**		**1**		
	1	**1**	**1**		**1**	**1**	$\leftarrow \partial(6)$

Variable ordering and support set sequence

$\mathcal{V}_6$

	1	2	3	4	5	6	
	1			**1**			
$\mathcal{E}_6$							
		1	**1**		**1**		
	1	**1**	**1**		**1**	**1**	$\longleftarrow \partial(6)$
U_6	**1**	**1**	**1**		**1**	**1**	

Variable ordering and support set 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 support set sequence

- Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a variable ordering $\sigma = (v_1, \ldots, v_n)$

Variable ordering and support set sequence

- Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a variable ordering $\sigma = (v_1, \ldots, v_n)$
- $\mathcal{H}_n^\sigma = \mathcal{H}$.

Variable ordering and support set sequence

- Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a variable ordering $\sigma = (v_1, \ldots, v_n)$
- $\mathcal{H}_n^\sigma = \mathcal{H}$.
- For $k = n-1, \ldots, 0$, define $\mathcal{H}_k^\sigma = (\mathcal{V}_k^\sigma, \mathcal{E}_k^\sigma)$

Variable ordering and support set sequence

- Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a variable ordering $\sigma = (v_1, \ldots, v_n)$
- $\mathcal{H}_n^\sigma = \mathcal{H}$.
- For $k = n-1, \ldots, 0$, define $\mathcal{H}_k^\sigma = (\mathcal{V}_k^\sigma, \mathcal{E}_k^\sigma)$

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

Variable ordering and support set sequence

- Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a variable ordering $\sigma = (v_1, \ldots, v_n)$
- $\mathcal{H}_n^\sigma = \mathcal{H}$.
- For $k = n-1, \ldots, 0$, define $\mathcal{H}_k^\sigma = (\mathcal{V}_k^\sigma, \mathcal{E}_k^\sigma)$

$$\begin{aligned} \partial^\sigma(v_{k+1}) &= \left\{ S \mid S \in \mathcal{E}_{k+1}^\sigma \text{ and } v_{k+1} \in S \right\} \\ U_{k+1}^\sigma &= \bigcup_{S \in \partial^\sigma(v_{k+1})} S \end{aligned}$$

Variable ordering and support set sequence

- Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a variable ordering $\sigma = (v_1, \ldots, v_n)$
- $\mathcal{H}_n^\sigma = \mathcal{H}$.
- For $k = n-1, \ldots, 0$, define $\mathcal{H}_k^\sigma = (\mathcal{V}_k^\sigma, \mathcal{E}_k^\sigma)$

$$\begin{aligned}
\partial^\sigma(v_{k+1}) &= \left\{S \mid S \in \mathcal{E}_{k+1}^\sigma \text{ and } v_{k+1} \in S\right\} \\
U_{k+1}^\sigma &= \bigcup_{S \in \partial^\sigma(v_{k+1})} S \\
\mathcal{V}_k^\sigma &= \mathcal{V}_{k+1}^\sigma - \{v_{k+1}\}
\end{aligned}$$

Variable ordering and support set sequence

- Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a variable ordering $\sigma = (v_1, \ldots, v_n)$
- $\mathcal{H}_n^\sigma = \mathcal{H}$.
- For $k = n-1, \ldots, 0$, define $\mathcal{H}_k^\sigma = (\mathcal{V}_k^\sigma, \mathcal{E}_k^\sigma)$

$$\begin{aligned}
\partial^\sigma(v_{k+1}) &= \left\{S \mid S \in \mathcal{E}_{k+1}^\sigma \text{ and } v_{k+1} \in S\right\} \\
U_{k+1}^\sigma &= \bigcup_{S \in \partial^\sigma(v_{k+1})} S \\
\mathcal{V}_k^\sigma &= \mathcal{V}_{k+1}^\sigma - \{v_{k+1}\} \\
\mathcal{E}_k^\sigma &= \left(\mathcal{E}_{k+1}^\sigma - \partial^\sigma(v_{k+1})\right) \cup \{U_{k+1}^\sigma - \{v_{k+1}\}\}
\end{aligned}$$

Variable ordering and support set sequence

- Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a variable ordering $\sigma = (v_1, \ldots, v_n)$
- $\mathcal{H}_n^\sigma = \mathcal{H}$.
- For $k = n-1, \ldots, 0$, define $\mathcal{H}_k^\sigma = (\mathcal{V}_k^\sigma, \mathcal{E}_k^\sigma)$

$$
\begin{aligned}
\partial^\sigma(v_{k+1}) &= \left\{ S \mid S \in \mathcal{E}_{k+1}^\sigma \text{ and } v_{k+1} \in S \right\} \\
U_{k+1}^\sigma &= \bigcup_{S \in \partial^\sigma(v_{k+1})} S \\
\mathcal{V}_k^\sigma &= \mathcal{V}_{k+1}^\sigma - \{v_{k+1}\} \\
\mathcal{E}_k^\sigma &= \left(\mathcal{E}_{k+1}^\sigma - \partial^\sigma(v_{k+1})\right) \cup \{U_{k+1}^\sigma - \{v_{k+1}\}\}
\end{aligned}
$$

- $(U_n^\sigma, U_{n-1}^\sigma, \cdots, U_1^\sigma)$ is called the suport set sequence of σ.

FAQ-width of a variable ordering σ

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

FAQ-width of a variable ordering σ

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

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

FAQ-width of a variable ordering σ

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

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

Dynamic Programming for

for Matrix Chain Multiplication

and The FFT

FAQ-width of a variable ordering σ

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

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

$$\leq \quad \tilde{O}(mn)\left[\sum_{k \in K} N^{\rho^*_{\mathcal{H}}(U_k^\sigma)} + |\varphi|\right]$$

FAQ-width of a variable ordering σ

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

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

$$\leq \quad \tilde{O}(mn)\left[\sum_{k\in K} N^{\rho^*_{\mathcal{H}}(U_k^\sigma)} + |\varphi|\right]$$

Definition (FAQ-width of a variable ordering σ)

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

FAQ-width of a variable ordering σ

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

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

$$\leq \quad \tilde{O}(mn)\left[\sum_{k\in K} N^{\rho^*_{\mathcal{H}}(U_k^\sigma)} + |\varphi|\right]$$

$$\leq \quad \tilde{O}(mn)\left[nN^{\mathsf{faqw}(\sigma)} + |\varphi|\right].$$

Definition (FAQ-width of a variable ordering σ)

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

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 $\sigma = (v_1, \ldots, v_n)$ 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 $\sigma = (v_1, \ldots, v_n)$ that satisfies:

(a) The set $\{v_1, \ldots, v_f\}$ 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

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

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

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

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

EVO(φ) 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

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

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

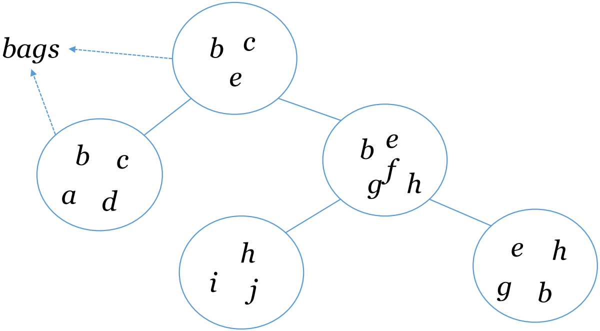
$\mathsf{EVO}(\varphi)$ denotes the set of all φ-equivalent variable orderings.

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

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

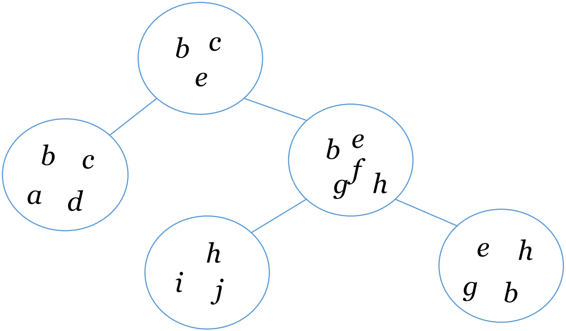
Detour: Tree decomposition of $\mathcal{H} = (\mathcal{V}, \mathcal{E})$

{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}.



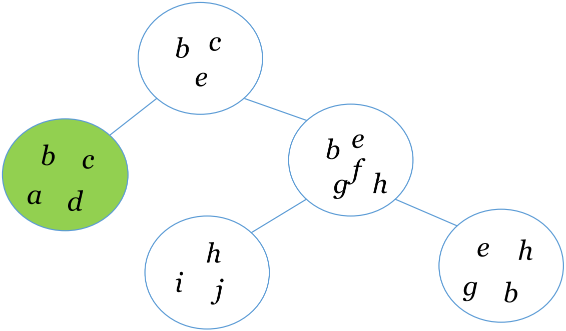
Every hyperedge is covered by some bag

{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}.



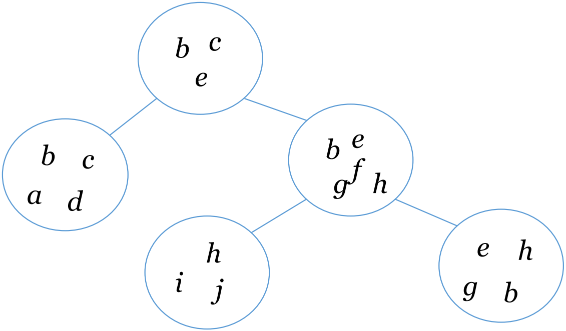
Every hyperedge is covered by some bag

{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}.



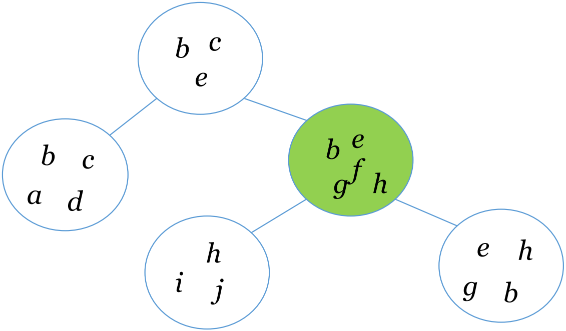
Every hyperedge is covered by some bag

{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}.



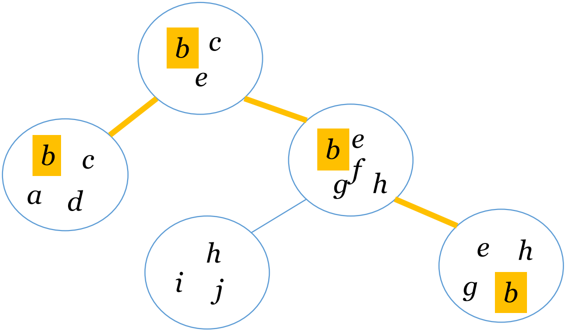
Every hyperedge is covered by some bag

{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}.



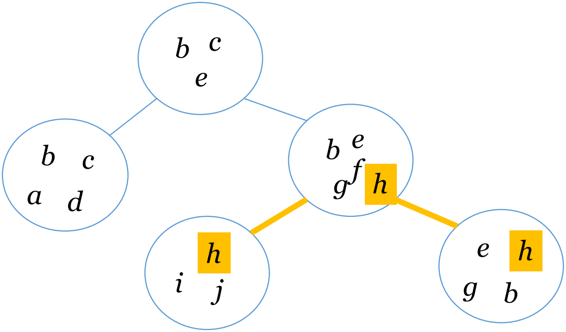
Running intersection property (RIP)

{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}.



Tree decomposition, formally

- Hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$
- A tree decomposition is a pair (T, χ)
  - $T = (V(T), E(T))$ is a tree
  - $\chi : V(T) \to 2^{\mathcal{V}}$, $\chi(t)$ called *bag*
  - Coverage property :

$$\forall F \in \mathcal{E}, \qquad \exists t \in V(T) \text{ s.t. } F \subseteq \chi(t).$$

  - RIP :

$$\forall v \in \mathcal{V}, \qquad \{t \mid t \in V(T), v \in \chi(t)\} \text{ is a connected sub-tree}$$

  - Non-redundant if no bag is contained in another.

Variable Ordering and Tree decomposition

- Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and a variable ordering $\sigma = (v_1, \ldots, v_n)$
- Corresponding support sets $U^\sigma_n, U^\sigma_{n-1}, \ldots, U^\sigma_1$

Proposition (Folklorish)

- *For every variable ordering σ, there is a tree decomposition whose bags are precisely the sets U^σ_k*
- *For every tree decomposition (T, χ), there is a variable ordering σ such that every U^σ_k is covered by some bag.*

faqw in terms of tree decompositions

Definition (FAQ-width of a variable ordering σ)

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

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

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

faqw in terms of tree decompositions

Definition (FAQ-width of a variable ordering σ)

$$\mathsf{faqw}(\sigma) := \max_{k \in K} \rho^*_{\mathcal{H}}(U^\sigma_k).$$

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

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

- Both can be rephrased in terms of tree decompositions and their bags

faqw in terms of tree decompositions

Definition (FAQ-width of a variable ordering σ)

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

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

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

- Both can be rephrased in terms of tree decompositions and their bags
  - Then InsideOut becomes message passing

faqw in terms of tree decompositions

Definition (FAQ-width of a variable ordering σ)

$$\mathsf{faqw}(\sigma) := \max_{k \in K} \rho^*_{\mathcal{H}}(U^\sigma_k).$$

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

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

- Both can be rephrased in terms of tree decompositions and their bags
  - Then InsideOut becomes message passing
- But mathematically very awkward,

faqw in terms of tree decompositions

Definition (FAQ-width of a variable ordering σ)

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

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

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

- Both can be rephrased in terms of tree decompositions and their bags
  - Then InsideOut becomes message passing
- But mathematically very awkward,
  - Except for when $\mathsf{EVO}(\varphi)$ contains all permutations

faqw in terms of tree decompositions

Definition (FAQ-width of a variable ordering σ)

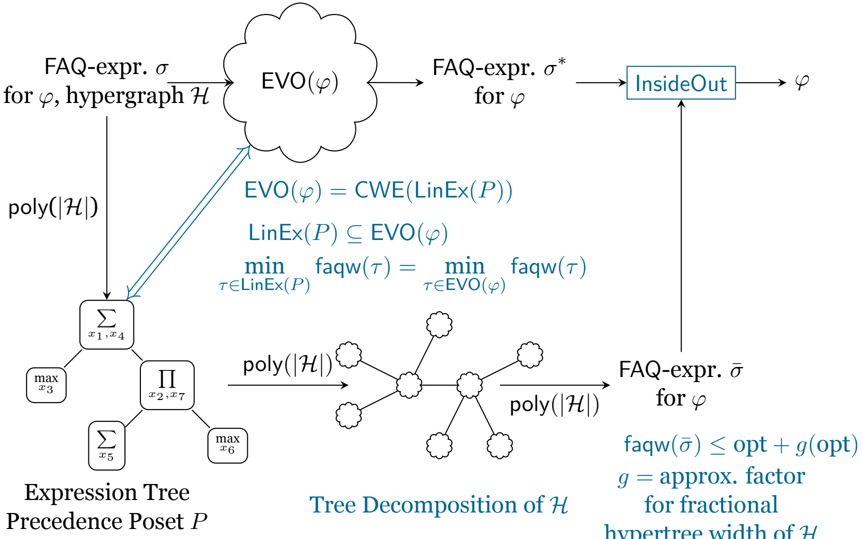
$$\mathsf{faqw}(\sigma) := \max_{k \in K} \rho^*_{\mathcal{H}}(U_k^\sigma).$$

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

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

- Both can be rephrased in terms of tree decompositions and their bags
  - Then InsideOut becomes message passing
- But mathematically very awkward,
  - Except for when $\mathsf{EVO}(\varphi)$ contains all permutations
  - In which case $\mathsf{faqw}(\varphi) = \mathsf{fhtw}(\mathcal{H})$

Roadmap Reminder



Characterizing EVO(φ)

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

Characterizing EVO(φ)

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

Soundness

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

Characterizing EVO(φ)

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

Soundness

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

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

Characterizing EVO(φ)

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

Soundness

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

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

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

Characterizing EVO(φ)

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

Soundness

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

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

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

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

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

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

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

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:

$$u \prec v \text{ whenever } u \in L \text{ and } v \notin L.$$

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

Theorem

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

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

we have an approximation for $\mathsf{faqw}(\varphi)$ *within abound of*

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

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

Theorem

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

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

we have an approximation for $\mathsf{faqw}(\varphi)$ *within abound of*

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

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

Theorem

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

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

we have an approximation for $\mathsf{faqw}(\varphi)$ *within abound of*

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

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

Theorem

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

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

we have an approximation for $\mathsf{faqw}(\varphi)$ *within abound of*

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

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

Theorem

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

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

we have an approximation for $\mathsf{faqw}(\varphi)$ *within abound of*

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

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

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

Theorem

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

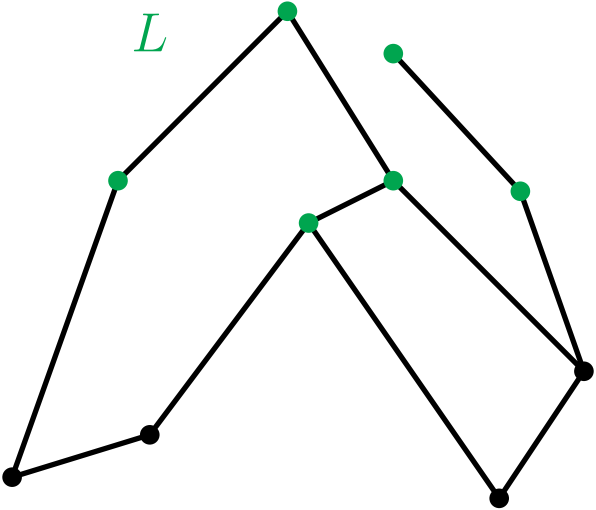
$$O(\mathsf{fhtw}(\mathcal{H})^3),$$

we have an approximation for $\mathsf{faqw}(\varphi)$ *within abound of*

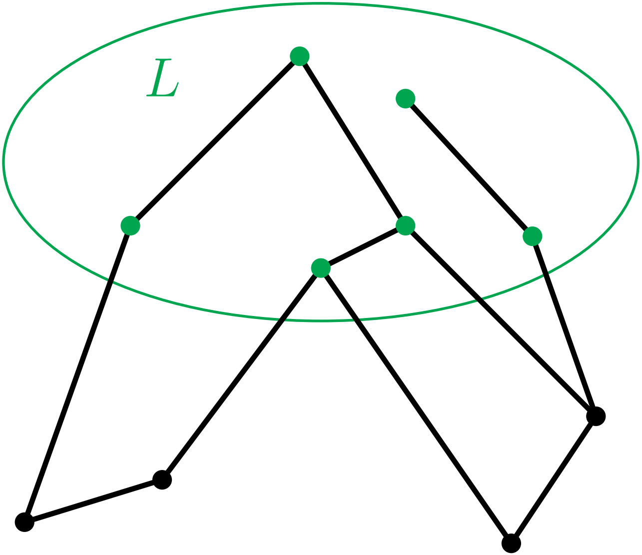
$$\mathrm{OPT} + O(\mathrm{OPT}^3).$$

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

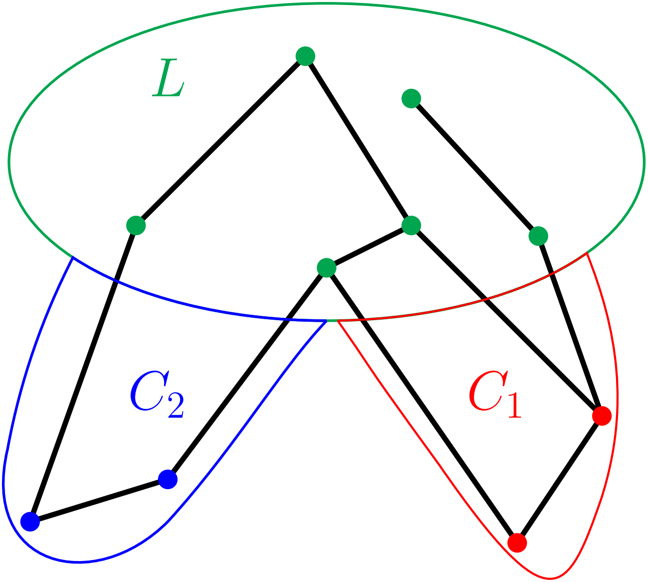
Approximating faqw(φ)



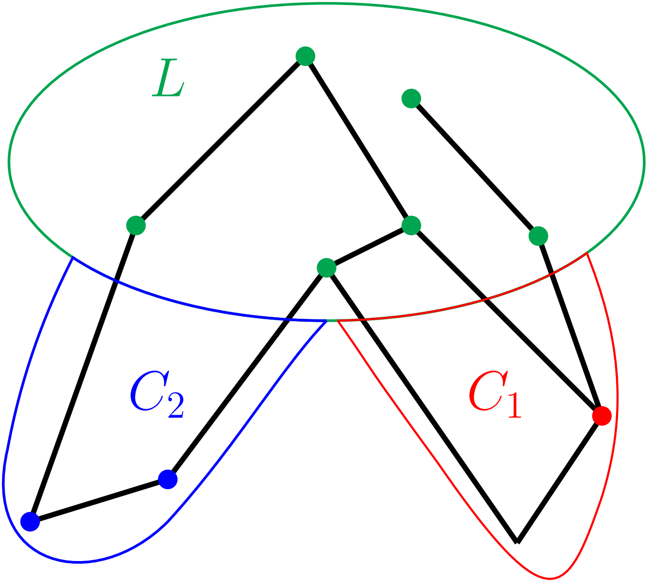
Approximating faqw(φ)



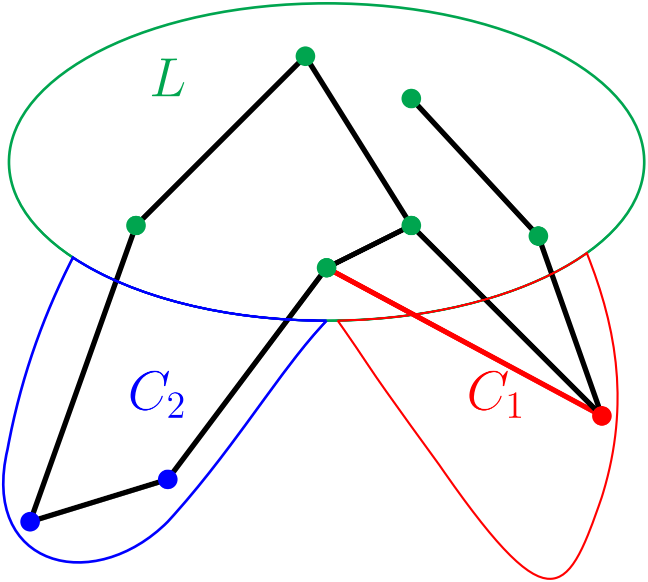
Approximating faqw(φ)



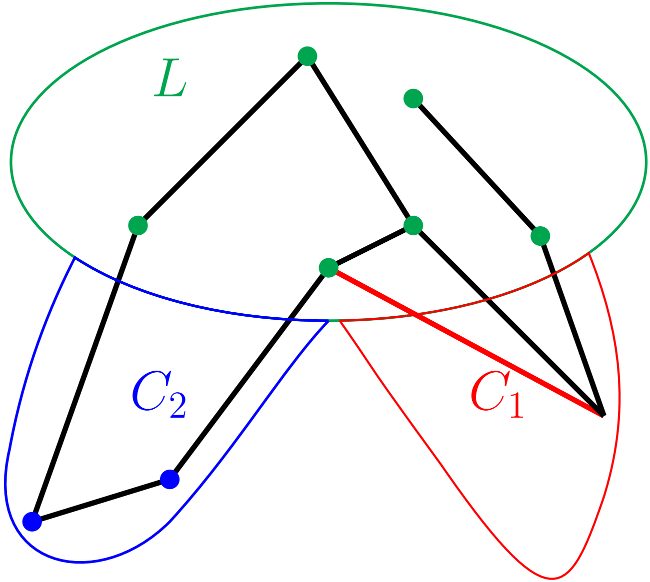
Approximating faqw(φ)



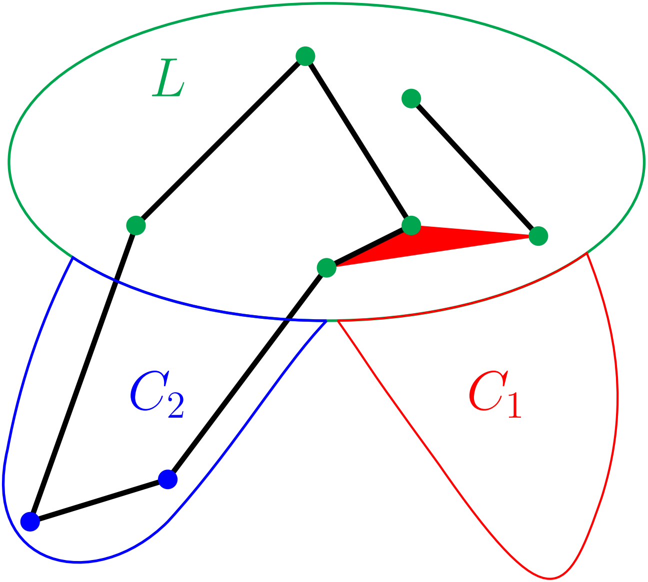
Approximating faqw(φ)



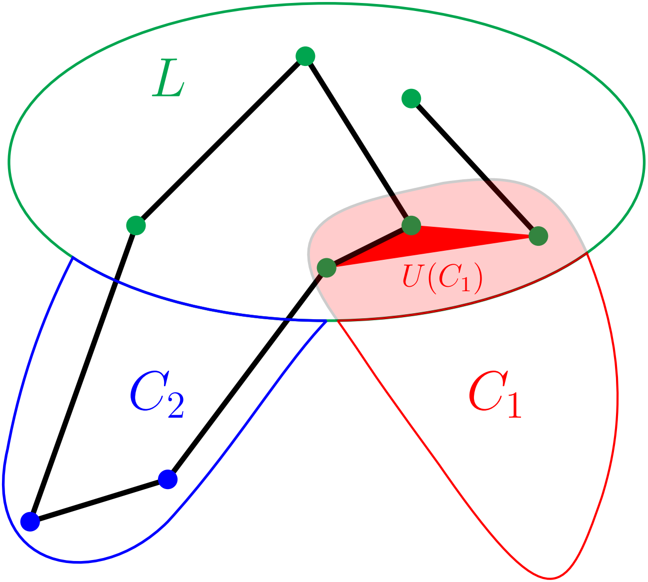
Approximating faqw(φ)



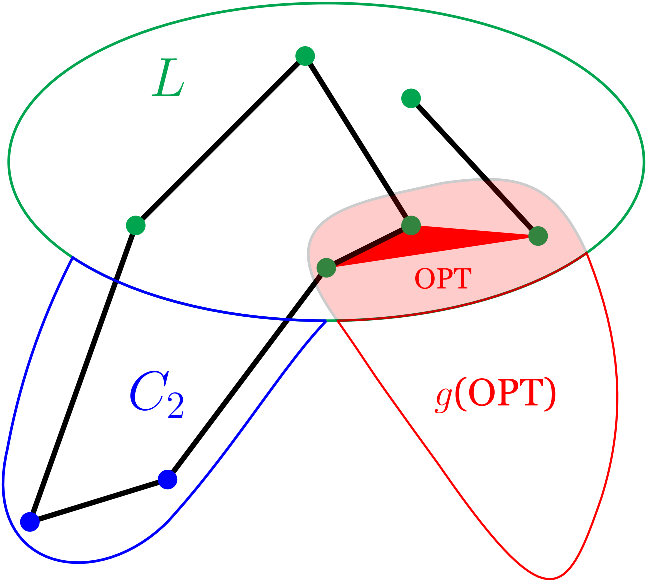
Approximating faqw(φ)



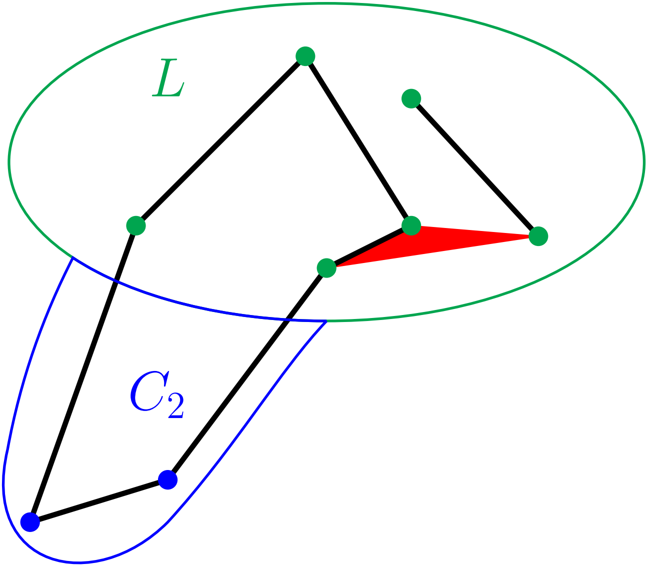
Approximating faqw(φ)



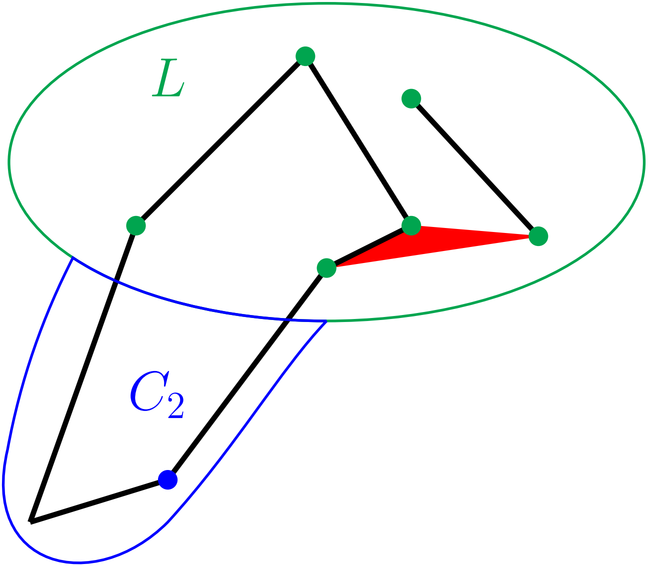
Approximating faqw(φ)



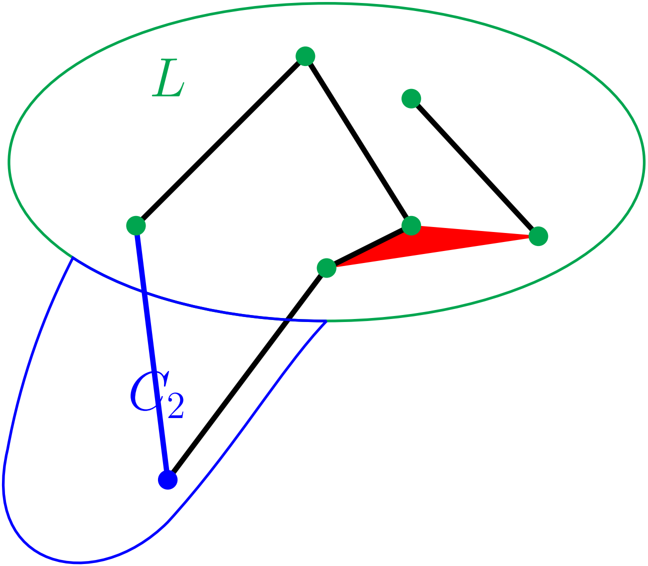
Approximating faqw(φ)



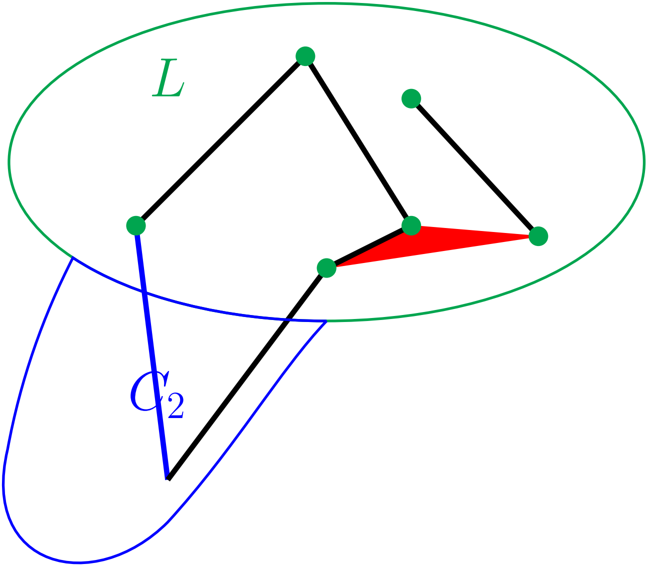
Approximating faqw(φ)



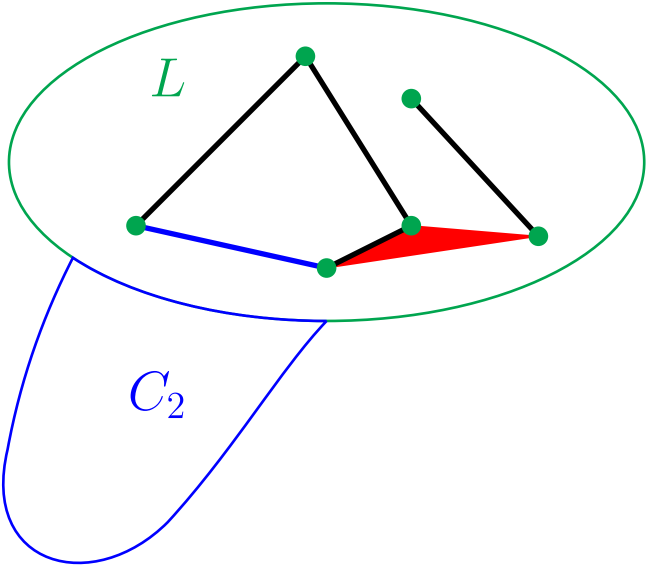
Approximating faqw(φ)



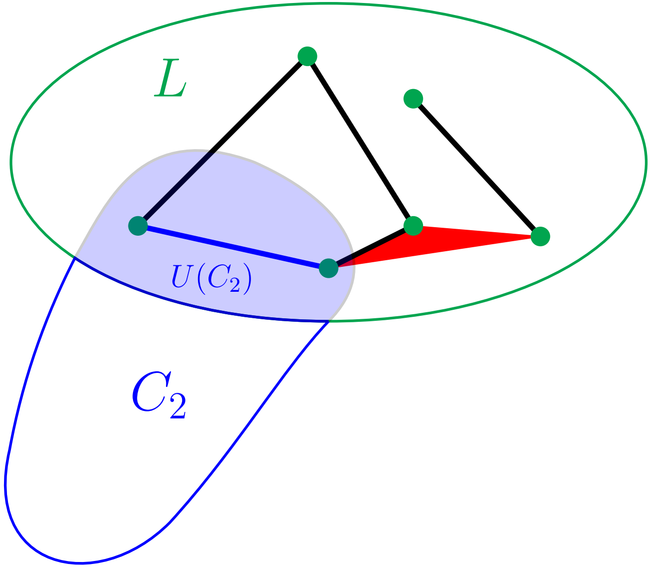
Approximating faqw(φ)



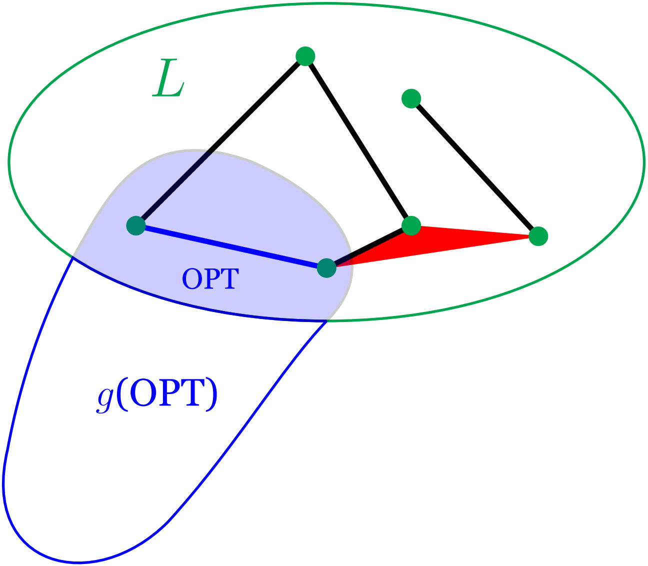
Approximating faqw(φ)



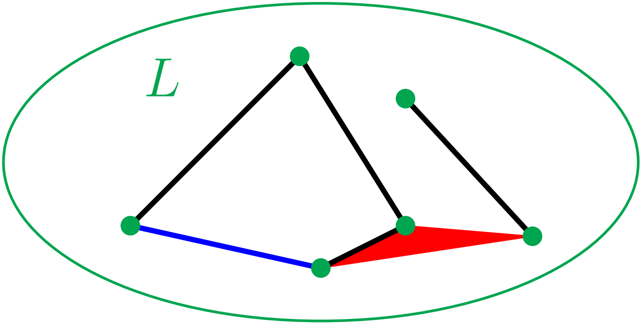
Approximating faqw(φ)



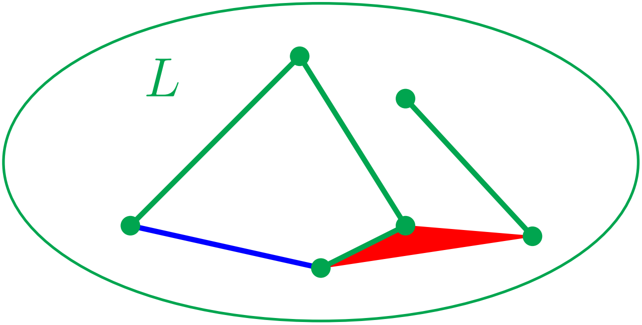
Approximating faqw(φ)



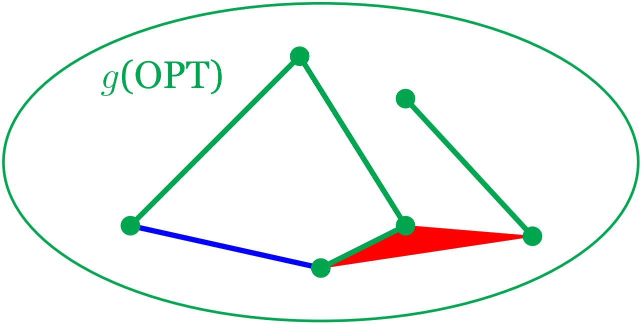
Approximating faqw(φ)



Approximating faqw(φ)



Approximating faqw(φ)



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

Corollary

FAQ *with two block of semiring aggregates is solvable in time*

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

Approximating faqw(φ)

Corollary (Durand-Mengel [ICDT'13])

#CQ is solvable in time

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

FAQ with only semiring aggregates

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

FAQ with only semiring aggregates

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

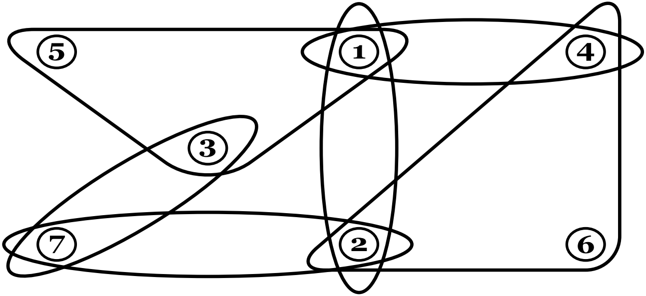
- Expression Tree defines the precedence poset

FAQ with only semiring aggregates

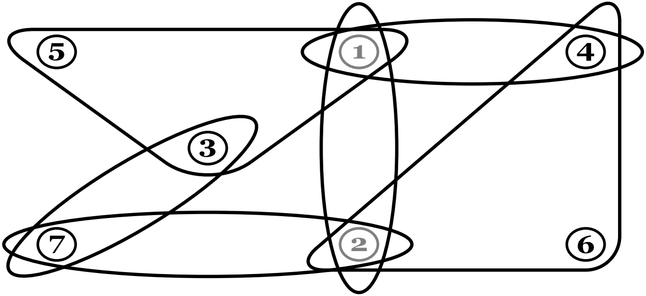
$$\varphi(\boldsymbol{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \cdots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\boldsymbol{x}_S)$$

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

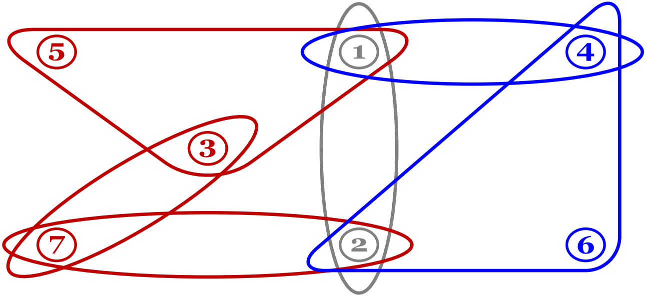
$$\varphi = \sum_{x_1} \sum_{x_2} \max_{x_3} \sum_{x_4} \sum_{x_5} \max_{x_6} \max_{x_7} \psi_{12}\, \psi_{135}\, \psi_{14}\, \psi_{246}\, \psi_{27}\, \psi_{37}$$



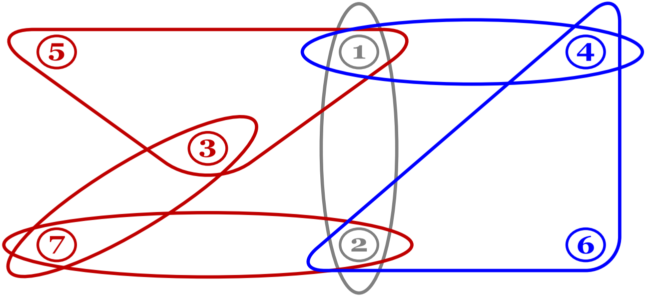
$$\varphi = \sum_{x_1} \sum_{x_2} \max_{x_3} \sum_{x_4} \sum_{x_5} \max_{x_6} \max_{x_7} \psi_{12}\, \psi_{135}\, \psi_{14}\, \psi_{246}\, \psi_{27}\, \psi_{37}$$



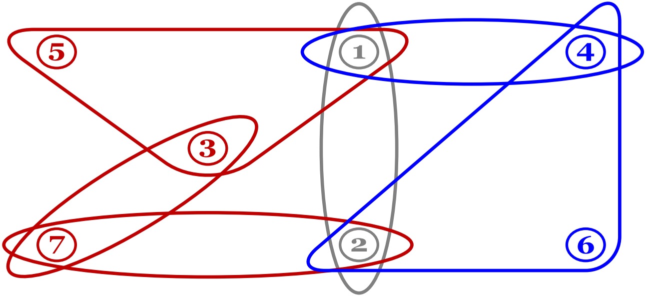
$$\varphi = \sum_{x_1} \sum_{x_2} \max_{x_3} \sum_{x_4} \sum_{x_5} \max_{x_6} \max_{x_7} \psi_{12}\, \psi_{135}\, \psi_{14}\, \psi_{246}\, \psi_{27}\, \psi_{37}$$



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

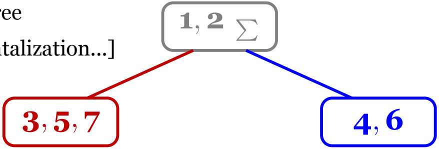


$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \max_{x_7} \psi_{135}\, \psi_{27}\, \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14}\, \psi_{246} \right)$$

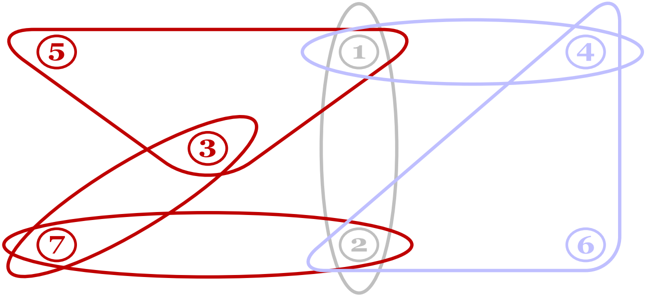


Expression Tree

[Compartmentalization...]

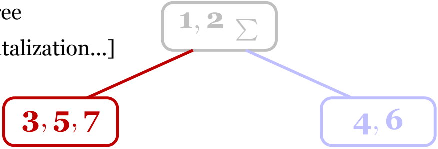


$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \max_{x_7} \psi_{135}\, \psi_{27}\, \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14}\, \psi_{246} \right)$$

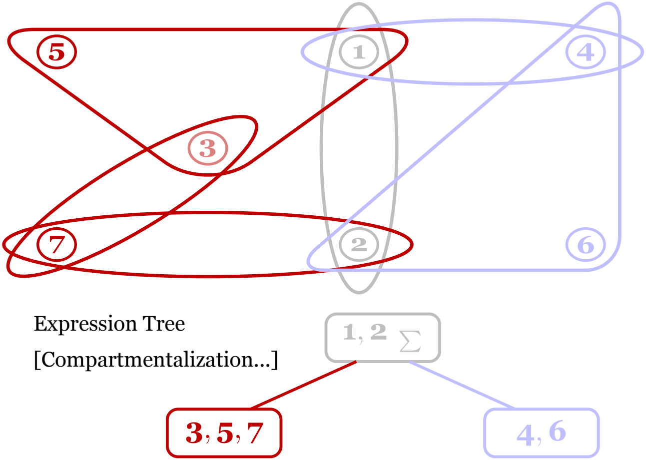


Expression Tree

[Compartmentalization...]



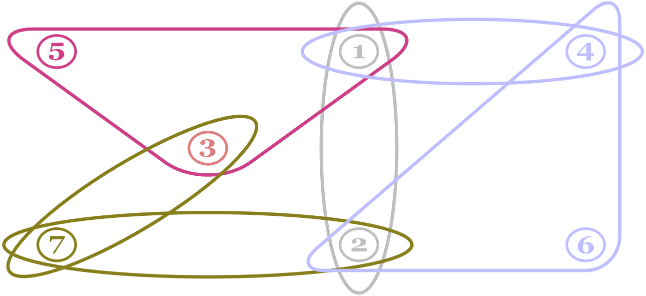
$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \max_{x_7} \psi_{135}\, \psi_{27}\, \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14}\, \psi_{246} \right)$$



Expression Tree

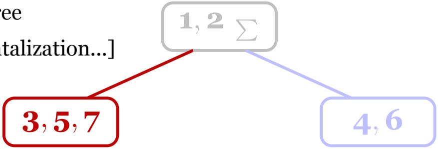
[Compartmentalization...]

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

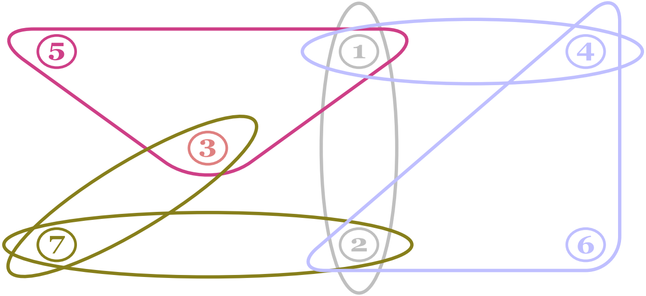


Expression Tree

[Compartmentalization...]

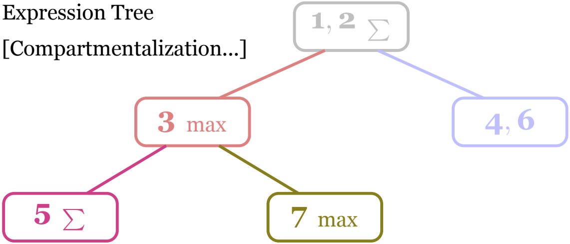


$$\varphi = \sum_{x_1}\sum_{x_2}\psi_{12}\left(\max_{x_3}\sum_{x_5}\psi_{135}\max_{x_7}\psi_{27}\,\psi_{37}\right)\left(\sum_{x_4}\max_{x_6}\psi_{14}\,\psi_{246}\right)$$

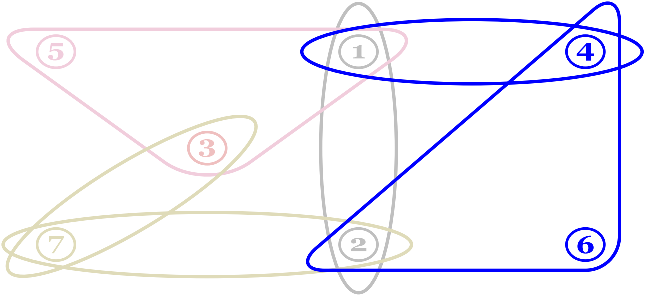


Expression Tree

[Compartmentalization...]

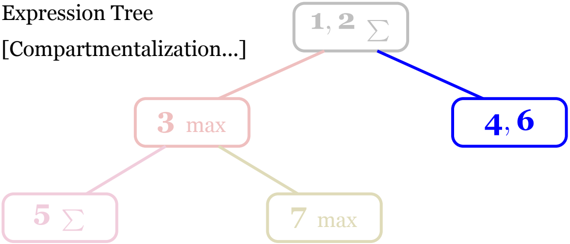


$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \psi_{135} \max_{x_7} \psi_{27}\, \psi_{37} \right) \left(\sum_{x_4} \max_{x_6} \psi_{14}\, \psi_{246} \right)$$

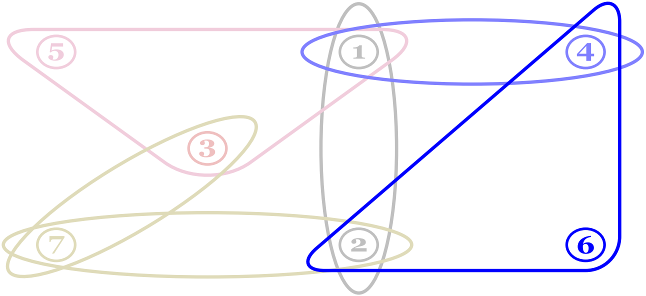


Expression Tree

[Compartmentalization...]

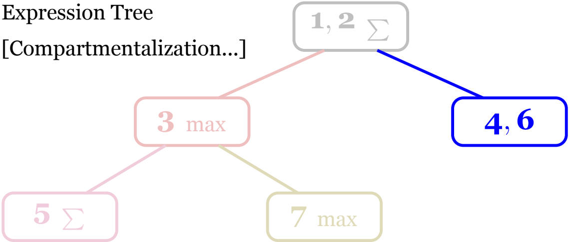


$$\varphi = \sum_{x_1}\sum_{x_2}\psi_{12}\left(\max_{x_3}\sum_{x_5}\psi_{135}\max_{x_7}\psi_{27}\,\psi_{37}\right)\left(\sum_{x_4}\max_{x_6}\psi_{14}\,\psi_{246}\right)$$

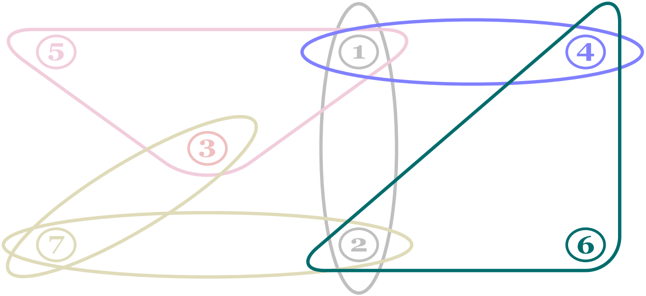


Expression Tree

[Compartmentalization...]

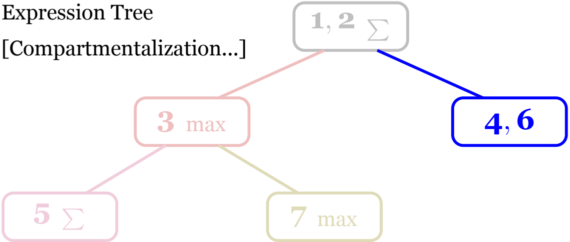


$$\varphi = \sum_{x_1}\sum_{x_2}\psi_{12}\left(\max_{x_3}\sum_{x_5}\psi_{135}\max_{x_7}\psi_{27}\,\psi_{37}\right)\left(\sum_{x_4}\max_{x_6}\psi_{14}\,\psi_{246}\right)$$

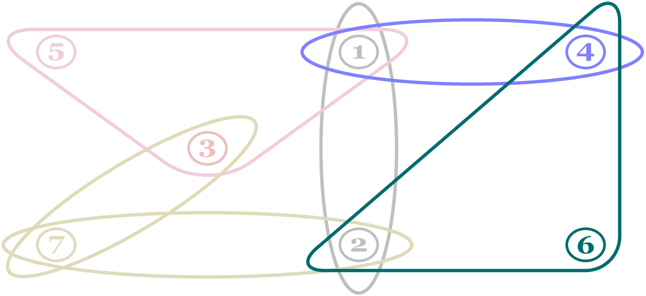


Expression Tree

[Compartmentalization...]

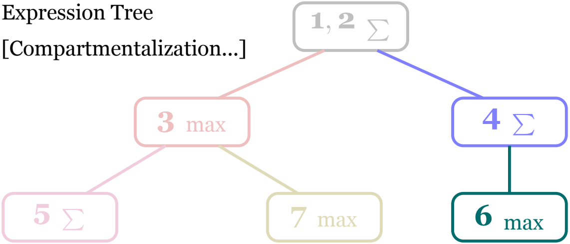


$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \psi_{135} \max_{x_7} \psi_{27} \psi_{37} \right) \left(\sum_{x_4} \psi_{14} \max_{x_6} \psi_{246} \right)$$

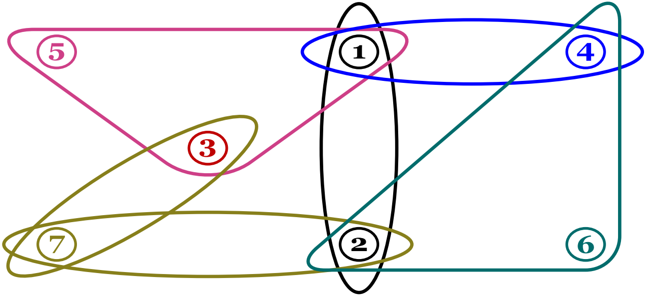


Expression Tree

[Compartmentalization...]

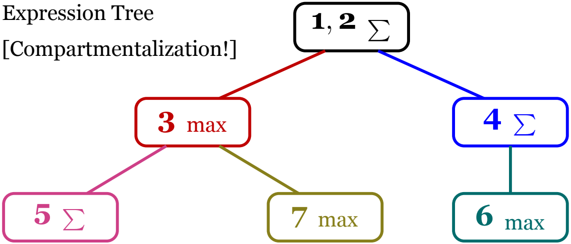


$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \sum_{x_5} \psi_{135} \max_{x_7} \psi_{27}\, \psi_{37} \right) \left(\sum_{x_4} \psi_{14} \max_{x_6} \psi_{246} \right)$$

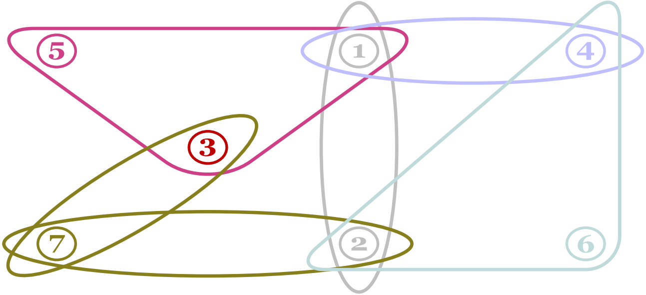


Expression Tree

[Compartmentalization!]

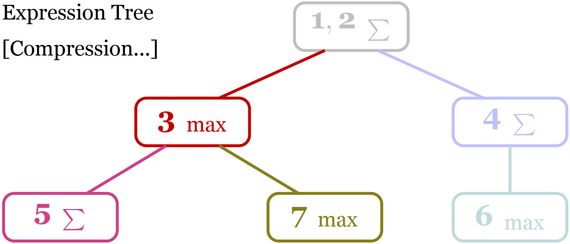


$$\varphi = \sum_{x_1}\sum_{x_2}\psi_{12}\left(\max_{x_3}\sum_{x_5}\psi_{135}\max_{x_7}\psi_{27}\,\psi_{37}\right)\left(\sum_{x_4}\psi_{14}\max_{x_6}\psi_{246}\right)$$

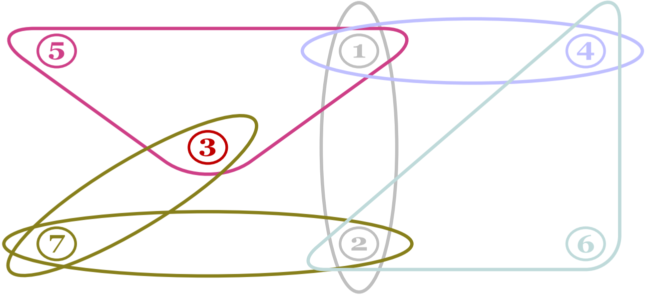


Expression Tree

[Compression...]

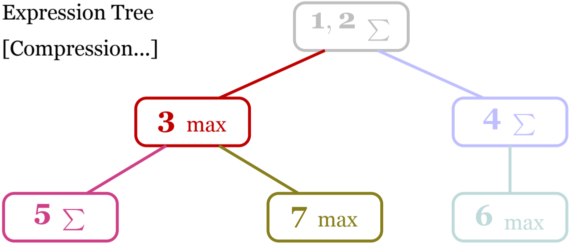


$$\varphi = \sum_{x_1}\sum_{x_2}\psi_{12}\left(\max_{x_3}\max_{x_7}\psi_{27}\,\psi_{37}\sum_{x_5}\psi_{135}\right)\left(\sum_{x_4}\psi_{14}\max_{x_6}\psi_{246}\right)$$

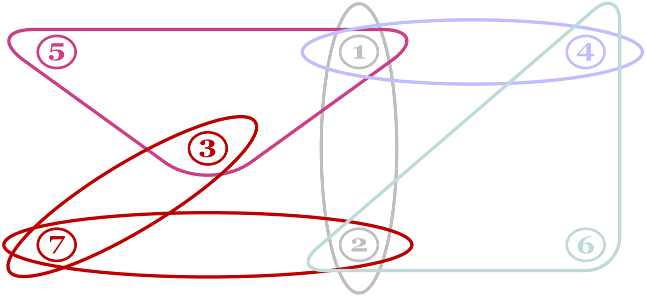


Expression Tree

[Compression...]

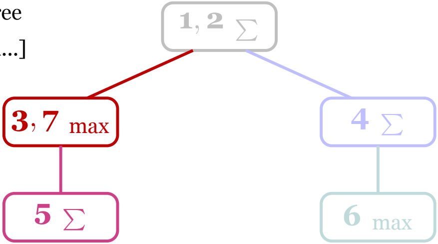


$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \max_{x_7} \psi_{27}\, \psi_{37} \sum_{x_5} \psi_{135} \right) \left(\sum_{x_4} \psi_{14} \max_{x_6} \psi_{246} \right)$$

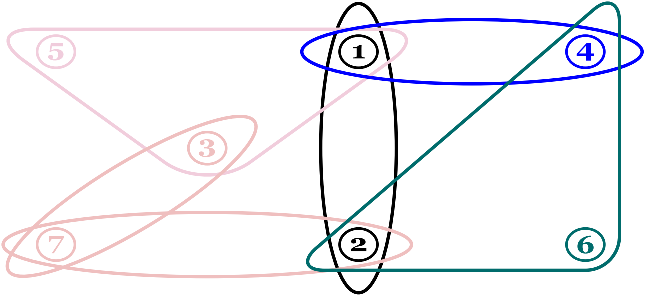


Expression Tree

[Compression...]

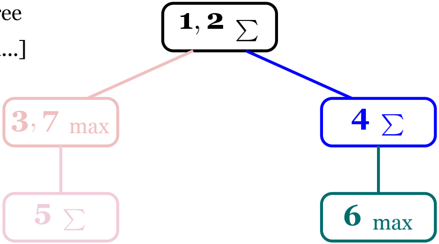


$$\varphi = \sum_{x_1} \sum_{x_2} \psi_{12} \left(\max_{x_3} \max_{x_7} \psi_{27}\, \psi_{37} \sum_{x_5} \psi_{135} \right) \left(\sum_{x_4} \psi_{14} \max_{x_6} \psi_{246} \right)$$

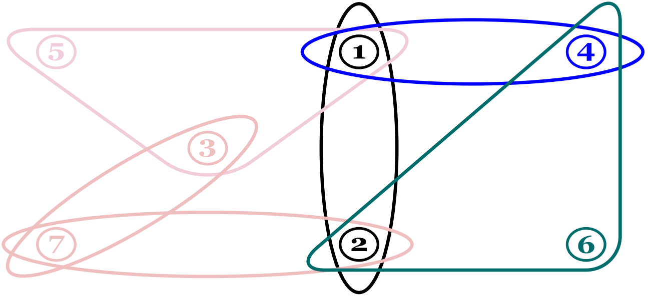


Expression Tree

[Compression...]

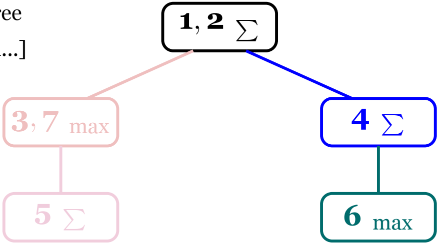


$$\varphi = \sum_{x_1} \sum_{x_2} \sum_{x_4} \psi_{12}\, \psi_{14} \left(\max_{x_3} \max_{x_7} \psi_{27}\, \psi_{37} \sum_{x_5} \psi_{135} \right) \left(\max_{x_6} \psi_{246} \right)$$

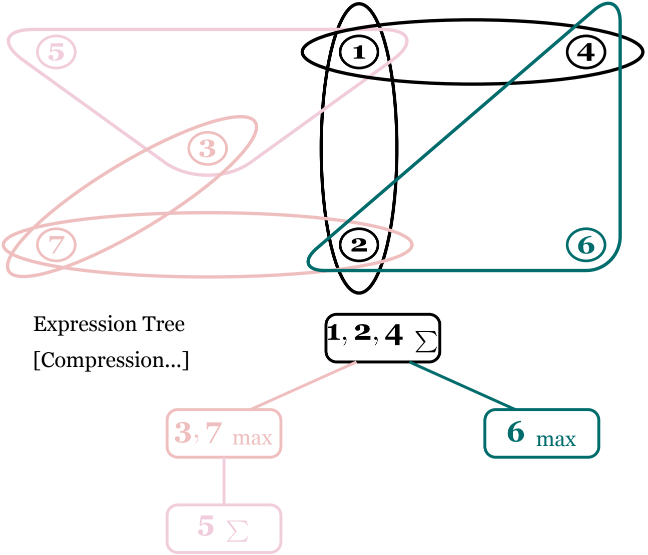


Expression Tree

[Compression...]



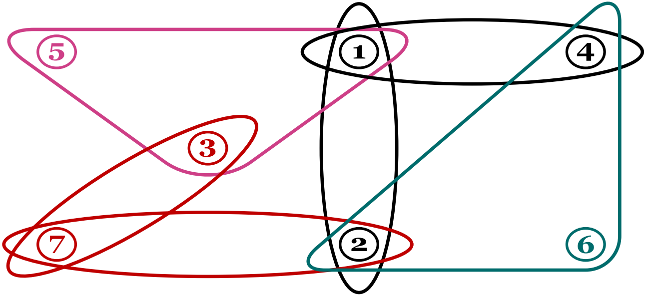
$$\varphi = \sum_{x_1} \sum_{x_2} \sum_{x_4} \psi_{12}\,\psi_{14} \left(\max_{x_3} \max_{x_7} \psi_{27}\,\psi_{37} \sum_{x_5} \psi_{135} \right) \left(\max_{x_6} \psi_{246} \right)$$



Expression Tree

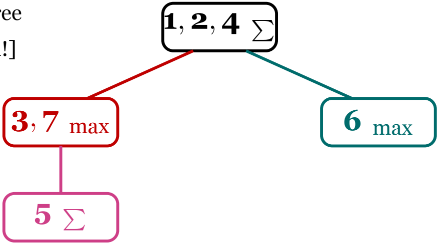
[Compression...]

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



Expression Tree

[Compression!]



Soundness and Completeness

Soundness

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

Soundness and Completeness

Soundness

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

Completeness

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

Soundness and Completeness

Soundness

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

Completeness

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

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

Soundness and Completeness

Soundness

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

Completeness

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

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

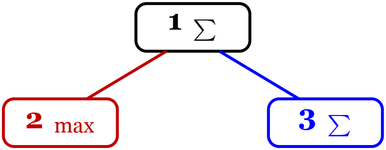
Soundness and Completeness

Soundness

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

Completeness

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



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

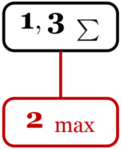
Soundness and Completeness

Soundness

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

Completeness

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



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

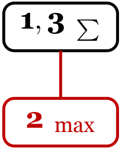
Soundness and Completeness

Soundness

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

Completeness

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



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

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

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

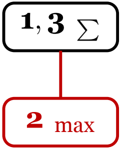
Soundness and Completeness

Soundness

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

Completeness

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



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

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

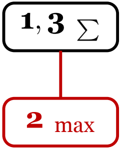
Soundness and Completeness

Soundness

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

Completeness

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



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

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

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

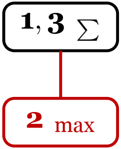
Soundness and Completeness

Soundness

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

Completeness

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



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

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

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

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

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

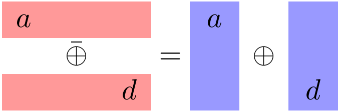
Completeness Background

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



Completeness Background

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

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

Completeness Background

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

$$\overset{a}{} \;\bar{\oplus}\; \underset{d}{} = \overset{a}{} \;\oplus\; \underset{d}{}$$

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

Completeness Background

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

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

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

Two semantically different aggregates may be identical by accident, e.g. min and $\times$ under $\{0, 1\}$

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

- The algorithm is a generalization of the two blocks case.

Approximating faqw(φ)

- The algorithm is a generalization of the two blocks case.
- Achieves the same approximation guarantee as the two blocks case!

Approximating faqw(φ)

- The algorithm is a generalization of the two blocks case.
- Achieves the same approximation guarantee as the two blocks case!
  - The number of blocks and the depth of the expression tree are irrelevant.

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

- The algorithm is a generalization of the two blocks case.
- Achieves the same approximation guarantee as the two blocks case!
  - The number of blocks and the depth of the expression tree are irrelevant.

Theorem

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

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

we have an approximation for $\mathsf{faqw}(\varphi)$ *within abound of*

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

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

- The algorithm is a generalization of the two blocks case.
- Achieves the same approximation guarantee as the two blocks case!
  - The number of blocks and the depth of the expression tree are irrelevant.

Theorem

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

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

we have an approximation for $\mathsf{faqw}(\varphi)$ *within abound of*

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

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 $\otimes$.

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

FAQ with *idempotent* product aggregates

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

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

FAQ with *idempotent* product aggregates

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

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

FAQ with *idempotent* product aggregates

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

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

FAQ with *idempotent* product aggregates

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

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

FAQ with *idempotent* product aggregates

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

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

Examples: QCQ,

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

FAQ with *idempotent* product aggregates

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

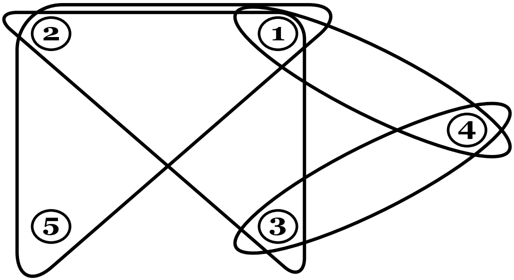
$$\varphi(\boldsymbol{x}_{[f]}) = \underbrace{\bigoplus_{x_{f+1}}^{(f+1)} \cdots \bigoplus_{x_{f+\ell}}^{(f+\ell)}}_{\text{Semiring aggregates}} \underbrace{\bigoplus_{x_{f+\ell+1}}^{(f+\ell+1)} \cdots \bigoplus_{x_n}^{(n)}}_{\text{Products or semirings closed under } \mathbf{D}_I} \underbrace{\bigotimes_{S\in\mathcal{E}}}_{\text{Closed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

Examples: QCQ, #QCQ

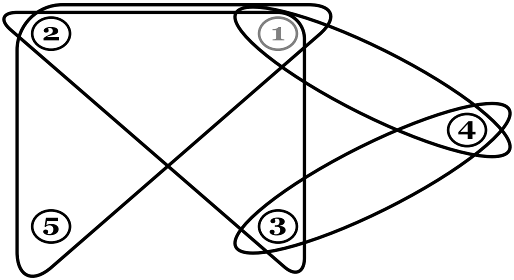
$$\varphi(\boldsymbol{x}_{[f]}) = \bigoplus_{x_{f+1}}^{(f+1)} \cdots \bigoplus_{x_n}^{(n)} \prod_{S\in\mathcal{E}} \psi_S(\boldsymbol{x}_S).$$

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

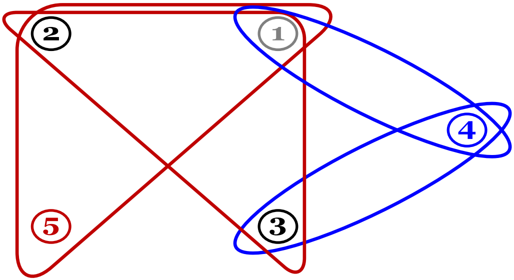
$$\varphi = \sum_{x_1} \max_{x_2} \prod_{x_3} \max_{x_4} \max_{x_5} \psi_{123}\, \psi_{125}\, \psi_{14}\, \psi_{34}$$



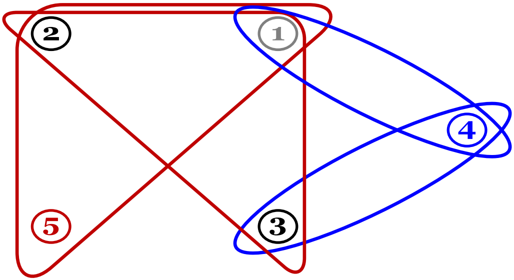
$$\varphi = \sum_{x_1} \max_{x_2} \prod_{x_3} \max_{x_4} \max_{x_5} \psi_{123}\, \psi_{125}\, \psi_{14}\, \psi_{34}$$



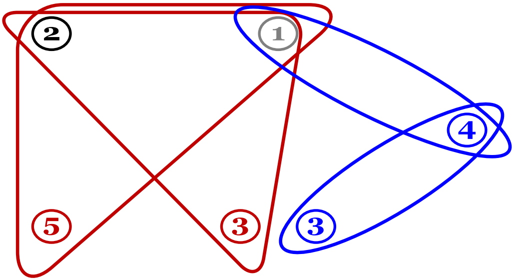
$$\varphi = \sum_{x_1} \max_{x_2} \prod_{x_3} \max_{x_4} \max_{x_5} \psi_{123}\, \psi_{125}\, \psi_{14}\, \psi_{34}$$



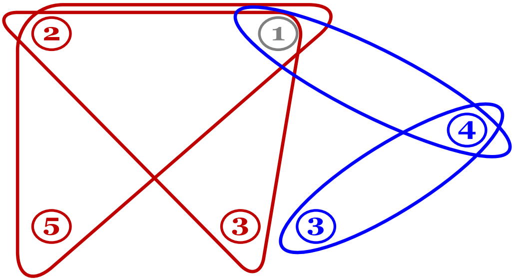
$$\varphi = \sum_{x_1} \max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123}\, \psi_{125} \max_{x_4} \psi_{14}\, \psi_{34}$$



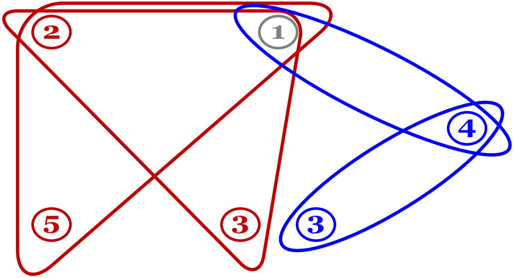
$$\varphi = \sum_{x_1} \max_{x_2} \left(\prod_{x_3} \max_{x_5} \psi_{123}\ \psi_{125} \right) \left(\prod_{x_3'} \max_{x_4} \psi_{14}\ \psi_{34} \right)$$



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

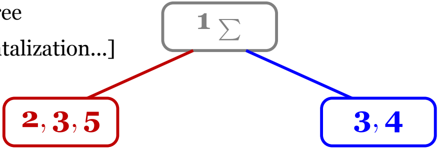


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123}\ \psi_{125} \right) \left(\prod_{x_3'} \max_{x_4} \psi_{14}\ \psi_{34} \right)$$

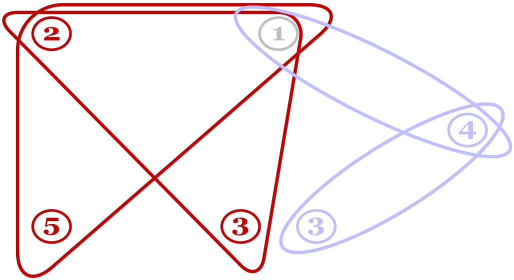


Expression Tree

[Compartmentalization...]

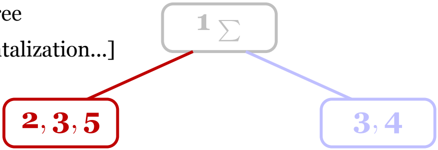


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123}\ \psi_{125} \right) \left(\prod_{x_3'} \max_{x_4} \psi_{14}\ \psi_{34} \right)$$

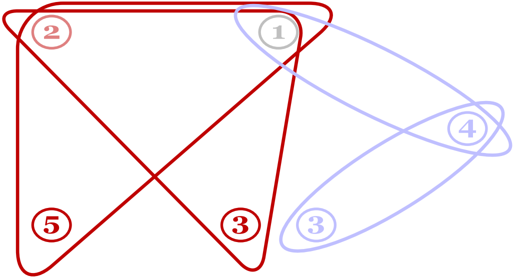


Expression Tree

[Compartmentalization...]

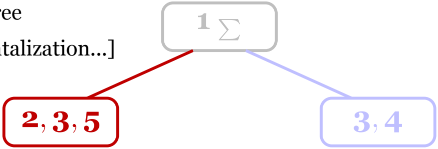


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123}\, \psi_{125} \right) \left(\prod_{x_3'} \max_{x_4} \psi_{14}\, \psi_{34} \right)$$

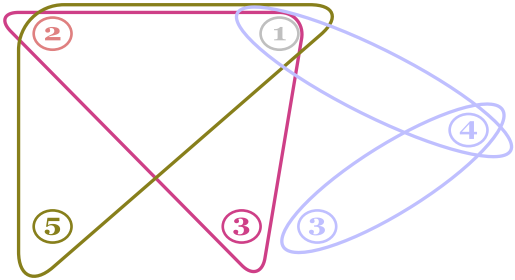


Expression Tree

[Compartmentalization...]

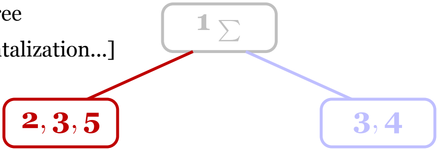


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \max_{x_5} \psi_{123}\, \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14}\, \psi_{34} \right)$$

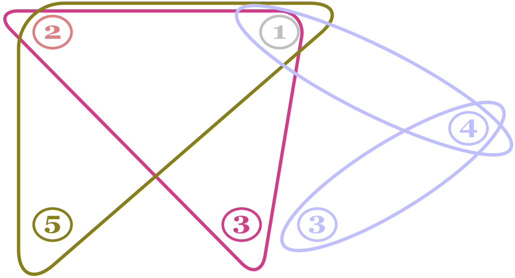


Expression Tree

[Compartmentalization...]

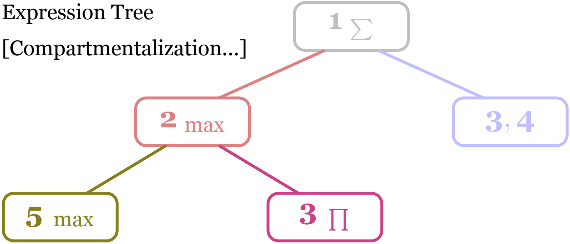


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x_3'} \max_{x_4} \psi_{14} \, \psi_{34} \right)$$

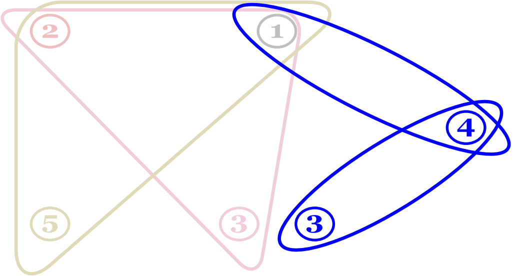


Expression Tree

[Compartmentalization...]

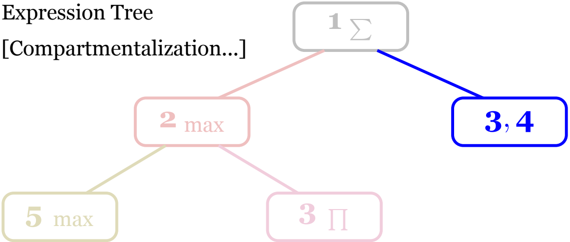


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x_3'} \max_{x_4} \psi_{14}\, \psi_{34} \right)$$

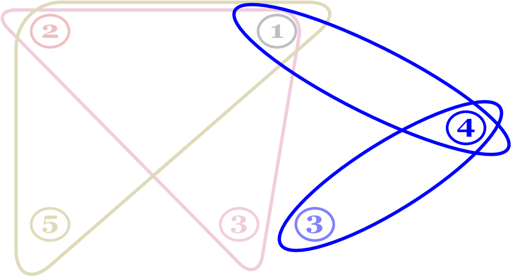


Expression Tree

[Compartmentalization...]

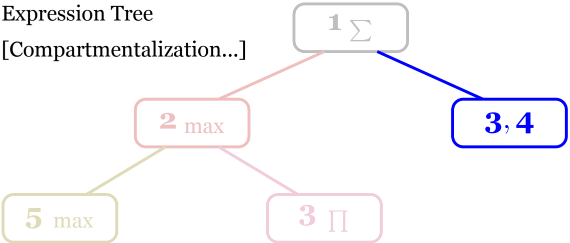


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \psi_{34} \right)$$

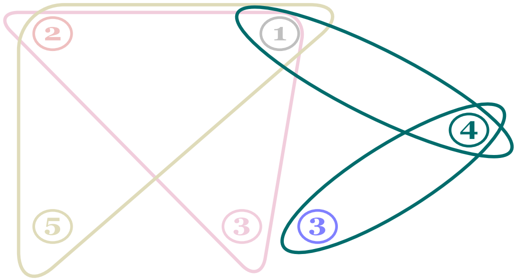


Expression Tree

[Compartmentalization...]

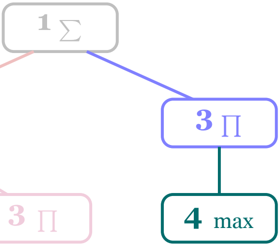


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x_3'} \max_{x_4} \psi_{14} \psi_{34} \right)$$

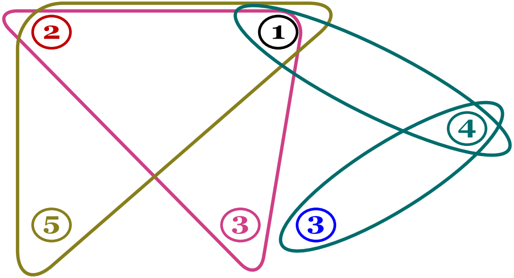


Expression Tree

[Compartmentalization...]

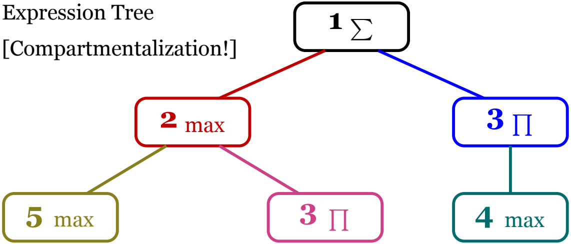


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \, \psi_{34} \right)$$

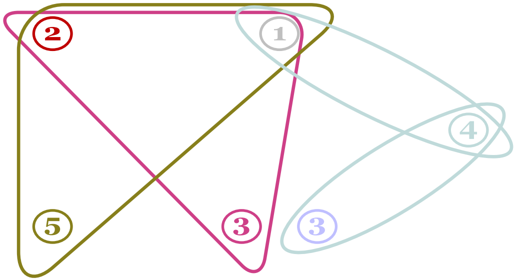


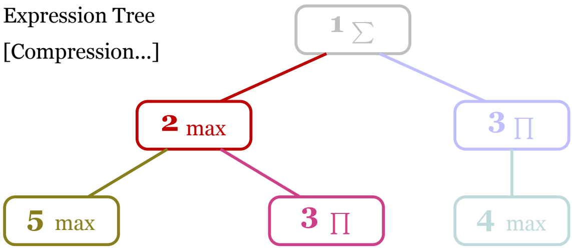
Expression Tree

[Compartmentalization!]

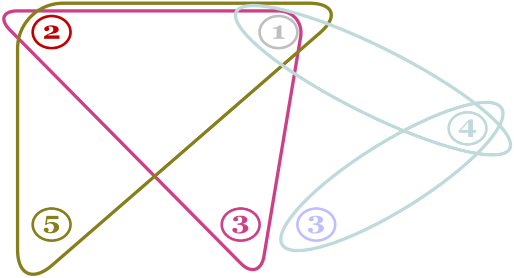


$$\varphi = \sum_{x_1} \left(\max_{x_2} \prod_{x_3} \psi_{123} \max_{x_5} \psi_{125} \right) \left(\prod_{x_3'} \max_{x_4} \psi_{14}\, \psi_{34} \right)$$



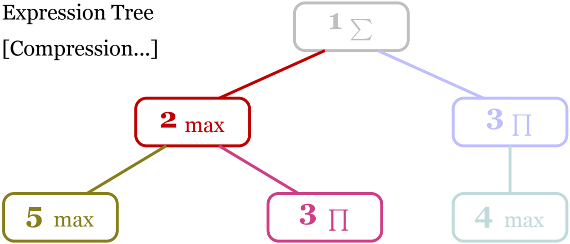


$$\varphi = \sum_{x_1} \left(\max_{x_2} \max_{x_5} \psi_{125} \prod_{x_3} \psi_{123} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \, \psi_{34} \right)$$

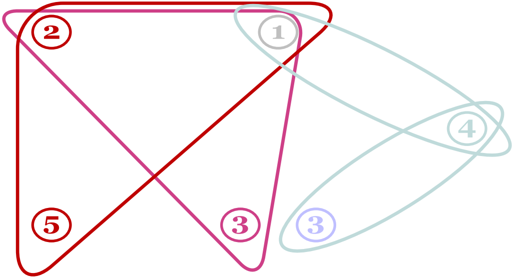


Expression Tree

[Compression...]

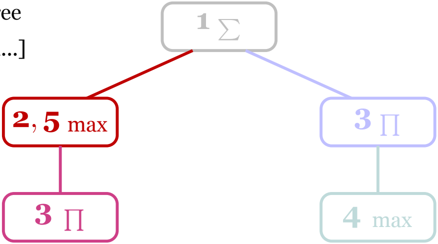


$$\varphi = \sum_{x_1} \left(\max_{x_2} \max_{x_5} \psi_{125} \prod_{x_3} \psi_{123} \right) \left(\prod_{x'_3} \max_{x_4} \psi_{14} \, \psi_{34} \right)$$

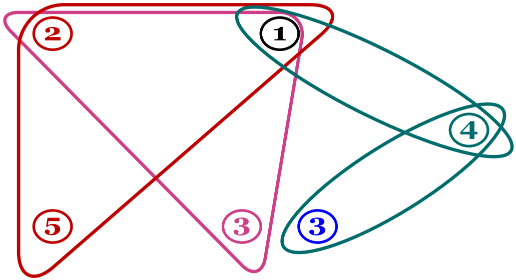


Expression Tree

[Compression...]

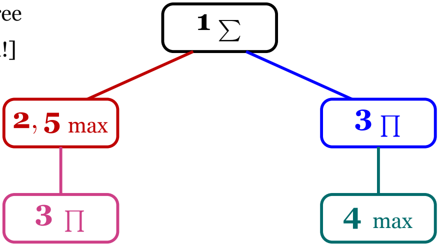


$$\varphi = \sum_{x_1} \left(\max_{x_2} \max_{x_5} \psi_{125} \prod_{x_3} \psi_{123} \right) \left(\prod_{x_3'} \max_{x_4} \psi_{14} \, \psi_{34} \right)$$



Expression Tree

[Compression!]



Corollaries

- Completeness proof and approximation algorithm can be extended to this case.

Corollaries

- Completeness proof and approximation algorithm can be extended to this case.
- Strictly stronger than Chen-Dalmau (LICS'12).

Corollaries

- Completeness proof and approximation algorithm can be extended to this case.
- Strictly stronger than Chen-Dalmau (LICS'12).

Corollary (QCQ tractability)

Corollaries

- Completeness proof and approximation algorithm can be extended to this case.
- Strictly stronger than Chen-Dalmau (LICS'12).

Corollary (QCQ tractability)

- QCQ *is tractable when* faqw *is bounded.*

Corollaries

- Completeness proof and approximation algorithm can be extended to this case.
- Strictly stronger than Chen-Dalmau (LICS'12).

Corollary (QCQ tractability)

- QCQ *is tractable when* faqw *is bounded.*
- *CD-width can be unbounded while* faqw *is bounded.*

Corollaries

- Completeness proof and approximation algorithm can be extended to this case.
- Strictly stronger than Chen-Dalmau (LICS'12).

Corollary (QCQ tractability)

- QCQ *is tractable when* faqw *is bounded.*
- *CD-width can be unbounded while* faqw *is bounded.*

- Answers an open question by Durant-Mengel (ICDT'13)

Corollaries

- Completeness proof and approximation algorithm can be extended to this case.
- Strictly stronger than Chen-Dalmau (LICS'12).

Corollary (QCQ tractability)

- QCQ *is tractable when* faqw *is bounded.*
- *CD-width can be unbounded while* faqw *is bounded.*

- Answers an open question by Durant-Mengel (ICDT'13)

Corollary (#QCQ tractability)

Corollaries

- Completeness proof and approximation algorithm can be extended to this case.
- Strictly stronger than Chen-Dalmau (LICS'12).

Corollary (QCQ tractability)

- QCQ *is tractable when* faqw *is bounded.*
- *CD-width can be unbounded while* faqw *is bounded.*

- Answers an open question by Durant-Mengel (ICDT'13)

Corollary (#QCQ tractability)

- #QCQ *is tractable when* faqw *is bounded.*

Many Thanks!
Any FAQ**?**