RelationalAI

CS294-248 Special Topics in Database Theory

Variable Elimination and Tree Decomposition Functional Aggregate Queries

Hung Q. Ngo

Outline

Warm-up Examples

Sum-product Queries

Variable Elimination and Tree Decomposition

Functional Aggregate Queries

What is the Best Way to Answer This Query?

```
def Q = count[x, y, z:  R(x) and S(y) and T(z)]
```

$O(N^3)$ typically

What is the Best Way to Answer This Query?

```
def Q = count[x, y, z:  R(x) and S(y) and T(z)]
```

$O(N^3)$ typically

Optimized query plan:

```
def t1 = count[x :  R(x)]

def t2 = count[y :  S(y)]

def t3 = count[z :  T(z)]

def Q = t1 * t2 * t3
```

What is the Best Way to Answer This Query?

```
def Q = count[x, y, z:  R(x) and S(y) and T(z) and x<y and y<z]
```

What is the Best Way to Answer This Query?

```
def Q = count[x, y, z:  R(x) and S(y) and T(z) and x<y and y<z]
```

Optimized query plan:

```
def T1[y] = count[x :  R(x) and S(y) and x<y]
def T2[y] = count[z :  T(z) and S(y) and y<z]
def Q = sum[y :  T1[y] * T2[y]]
```

What is the Best Way to Answer This Query?

```
def Q[x] = max[y, z, u:  R(x,y) and S(y,z) and T(z, u)]
```

typically $O(N^2)$

What is the Best Way to Answer This Query?

```
def Q[x] = max[y, z, u:  R(x,y) and S(y,z) and T(z, u)]
```

typically $O(N^2)$

Optimized query plan:

```
def W1[z] = max[u :  T(z, u)]

def W2[y) = max[z, a :  S(y, z) and W1(z, a)]

def Q[x] = max[y, b :  R(x, y) and W2(y, b)]
```

What is the Best Way to Answer This Query?

```
def Q = count[a,b,c,d,e:
          E(a,b) and E(a,c) and E(b,c) and E(c,d) and E(c,e)
        ]
```

Assuming E is the edge relation of a graph.

What is the Best Way to Answer This Query?

```
def Q = count[a,b,c,d,e:
          E(a,b) and E(a,c) and E(b,c) and E(c,d) and E(c,e)
        ]
```

Assuming E is the edge relation of a graph.

Optimized query plan:

```
def T[c] = count[d :  E(c, d)]

def Q = sum[a,b,c,z:  E(a,b) and E(a,c) and E(b,c) and z = T[c]*T[c]]
```

What is the Best Way to Answer This Query?

```
def Q = count[a,b,c,d,e,f:
          R(a,b) and S(a,c) and T(b,c,d,e) and W(e,f) and V(d,f)
        ]
```

Assuming all relations have the same size N.

Abstractions We Need

- Algebraic abstraction to model these problems
- Algorithmic abstraction to solve all of them
- Query optimization abstraction to solve them efficiently

Outline

Given a semiring $(D, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, a Sum-Product-query is

$$\varphi(\boldsymbol{X}_F) := \sum_{\boldsymbol{X}_{V-F}} \prod_{S \in \mathcal{E}} \psi_S(\boldsymbol{X}_S)$$

Given a semiring $(D, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, a Sum-Product-query is

$$\varphi(\boldsymbol{X}_F) := \sum_{\boldsymbol{X}_{V-F}} \prod_{S \in \mathcal{E}} \psi_S(\boldsymbol{X}_S)$$

Example 1: $(\{\text{true}, \text{false}\}, \vee, \wedge, \text{false}, \text{true})$ is the Boolean semiring.

$$Q() = \bigvee_{a,b,c} E(a,b) \wedge E(a,c) \wedge E(b,c)$$

Given a semiring $(D, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, a Sum-Product-query is

$$\varphi(\boldsymbol{X}_F) := \sum_{\boldsymbol{X}_{V-F}} \prod_{S \in \mathcal{E}} \psi_S(\boldsymbol{X}_S)$$

Example 1: $(\{\text{true}, \text{false}\}, \vee, \wedge, \text{false}, \text{true})$ is the Boolean semiring.

$$Q() = \bigvee_{a,b,c} E(a,b) \wedge E(a,c) \wedge E(b,c)$$

Example 2: $(\mathbb{R}, +, \times, 0, 1)$ is the sum-product semiring.

$$Q(a) = \sum_{b,c,d} E(a,b) \times E(b,c) \times E(c,d) \times \mathbf{1}_{a \neq d}$$

Given a semiring $(D, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, a Sum-Product-query is

$$\varphi(\boldsymbol{X}_F) := \sum_{\boldsymbol{X}_{V-F}} \prod_{S \in \mathcal{E}} \psi_S(\boldsymbol{X}_S)$$

Example 1: $(\{\text{true}, \text{false}\}, \vee, \wedge, \text{false}, \text{true})$ is the Boolean semiring.

$$Q() = \bigvee_{a,b,c} E(a,b) \wedge E(a,c) \wedge E(b,c)$$

Example 2: $(\mathbb{R}, +, \times, 0, 1)$ is the sum-product semiring.

$$Q(a) = \sum_{b,c,d} E(a,b) \times E(b,c) \times E(c,d) \times \mathbf{1}_{a \neq d}$$

They are a great IR!

Mapping to Sum-Product **Representation**

```
def Q = count[x, y, z:  R(x) and S(y) and T(z)]
```

Mapping to Sum-Product Representation

```
def Q = count[x, y, z:  R(x) and S(y) and T(z)]
```

Abusing notation, define

$$R(x) := \mathbf{1}_{x \in R}$$
$$S(y) := \mathbf{1}_{y \in S}$$
$$T(z) := \mathbf{1}_{z \in T}$$

Mapping to Sum-Product Representation

```
def Q = count[x, y, z:  R(x) and S(y) and T(z)]
```

Abusing notation, define

$$R(x) := \mathbf{1}_{x \in R}$$
$$S(y) := \mathbf{1}_{y \in S}$$
$$T(z) := \mathbf{1}_{z \in T}$$

The correpsonding Sum-Product-query is over the $(+, \times)$-semiring

$$Q = \sum_x \sum_y \sum_z R(x) \times S(y) \times T(z)$$

Mapping to Sum-Product Representation

```
def Q = count[x, y, z:  R(x) and S(y) and T(z) and x<y and y<z]
```

Mapping to Sum-Product **Representation**

```
def Q = count[x, y, z:  R(x) and S(y) and T(z) and x<y and y<z]
```

The correpsonding Sum-Product-query is over the $(+, \times)$-semiring

$$Q = \sum_x \sum_y \sum_z R(x) \times S(y) \times T(z) \times \mathbf{1}_{x<y} \times \mathbf{1}_{y<z}$$

Mapping to Sum-Product **Representation**

```
def Q = sum[x, y, z:  R(x) and S(y) and T(z) and x<y and y<z]
```

Mapping to Sum-Product **Representation**

```
def Q = sum[x, y, z:  R(x) and S(y) and T(z) and x<y and y<z]
```

Abusing notation, define

$$R(x) := \mathbf{1}_{x \in R}$$

$$S(y) := \mathbf{1}_{y \in S}$$

$$T(z) := z \times \mathbf{1}_{z \in T}$$

Mapping to Sum-Product Representation

```
def Q = sum[x, y, z:  R(x) and S(y) and T(z) and x<y and y<z]
```

Abusing notation, define

$$R(x) := \mathbf{1}_{x \in R}$$
$$S(y) := \mathbf{1}_{y \in S}$$
$$T(z) := z \times \mathbf{1}_{z \in T}$$

The correpsonding Sum-Product-query is over the $(+, \times)$-semiring

$$Q = \sum_x \sum_y \sum_z R(x) \times S(y) \times T(z) \times \mathbf{1}_{x<y} \times \mathbf{1}_{y<z}$$

Mapping to Sum-Product Representation

```
def Q[x] = max[y, z, u:  R(x,y) and S(y,z) and T(z, u)]
```

Mapping to Sum-Product Representation

```
def Q[x] = max[y, z, u:  R(x,y) and S(y,z) and T(z, u)]
```

Abusing notation, define

$$R(x, y) := \mathbf{1}_{(x,y)\in R}$$

$$S(y, z) := \mathbf{1}_{(y,z)\in S}$$

$$T(z, u) := u \times \mathbf{1}_{(z,u)\in T}$$

Mapping to Sum-Product Representation

```
def Q[x] = max[y, z, u:  R(x,y) and S(y,z) and T(z, u)]
```

Abusing notation, define

$$R(x, y) := \mathbf{1}_{(x,y)\in R}$$
$$S(y, z) := \mathbf{1}_{(y,z)\in S}$$
$$T(z, u) := u\times\mathbf{1}_{(z,u)\in T}$$

The correpsonding Sum-Product-query is over the $(\max, \times)$-semiring

$$Q(x) = \max_y \max_z R(x, y) \times S(y, z) \times T(z, y)$$

Outline

Sum-Product **Representation and Variable Elimination**

Over the $(+, \times)$-semiring

$$
\begin{aligned}
Q &= \sum_x \sum_y \sum_z R(x) \times S(y) \times T(z) \\
&= \sum_x \sum_y R(x) \times S(y) \times \sum_z T(z) \\
&= \sum_x \sum_y R(x) \times S(y) \times t_1 \\
&= t_1 \times \sum_x R(x) \times \sum_y S(y) \\
&= t_1 \times t_2 \times t_3
\end{aligned}
$$

The exact same query plans works over $(\max, \times)$, $(\min, +)$, $(\vee, \wedge)$, etc.

Sum-Product **Representation and Variable Elimination**

The correpsonding Sum-Product-query is over the $(+, \times)$-semiring

$$\begin{aligned} Q &= \sum_x \sum_y \sum_z R(x) \times S(y) \times T(z) \times \mathbf{1}_{x<y} \times \mathbf{1}_{y<z} \\ &= \sum_x \sum_y R(x) \times \mathbf{1}_{x<y} \times \sum_z S(y) \times T(z) \times \mathbf{1}_{y<z} \\ &= \sum_x \sum_y R(x) \times \mathbf{1}_{x<y} \times M_1(y) \end{aligned}$$

Variable Elimination

$Q = \textbf{count}[a, b, c, d : R(a, b) \wedge S(b, c) \wedge T(c, d)]$

Variable elimination [ZP 94]

Works for any semi-ring!

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

Variable elimination [ZP 94]

Works for any semi-ring!

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

Variable elimination [ZP 94]

Works for any semi-ring!

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

Variable elimination [ZP 94]

Works for any semi-ring!

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

Variable elimination [ZP 94]

Works for any semi-ring!

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

Variable Elimination

$$Q = \textbf{count}[a, b, c, d : R(a, b) \wedge S(b, c) \wedge T(c, d)]$$

Variable elimination [ZP 94]

Works for any semi-ring!

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

Database jargon: (aggregation | projection | predicate) pushdowns

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

Indicator Projection, and Fractional Hypertree Width [GM 06] [ANR 16]

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

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

Indicator Projection, and Fractional Hypertree Width

[GM 06] [ANR 16]

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

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

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

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

Indicator Projection, and Fractional Hypertree Width [GM 06] [ANR 16]

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

Indicator Projection, and Fractional Hypertree Width

[GM 06] [ANR 16]

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

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

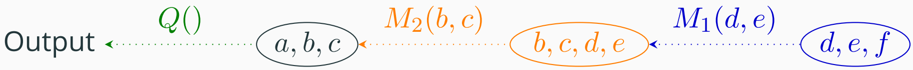
$$= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e)M_1(d,e)$$

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

$$= \sum_{a,b,c} R(a,b)S(a,c)M_2(b,c) \qquad = \underbrace{\sum_{a,b,c} R(a,b)S(a,c)M_2(b,c)}_{\tilde{O}(N^{3/2}),\ \text{WCOJ}}$$

Variable Elimination, Message Passing, Belief Propagation, Yannakakis

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



- Yannakakis algorithm [Yannakakis 81]
- Belief propagation [Pearl 82]

Indicator Projection, and Fractional Hypertree Width

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

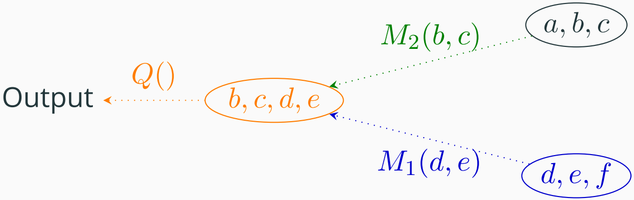
$$= \sum_{a,b,c,d,e} R(a,b)S(a,c)T(b,c,d,e) \underbrace{\sum_{f} \pi_{de}T(d,e) W(e,f)V(d,f)}_{M_1(d,e) \text{ in } \tilde{O}(N^{3/2}), \text{ WCOJ}}$$

$$= \sum_{b,c,d,e} T(b,c,d,e)M_1(d,e) \underbrace{\sum_{a} \pi_{bc}T(b,c)R(a,b)S(a,c)}_{M_2(b,c) \text{ in } O(N^{3/2}), \text{ WCOJ}}$$

$$= \sum_{b,c,d,e} M_1(d,e)T(b,c,d,e)M_2(b,c)$$

Variable Elimination, Message Passing, Belief Propagation, Yannakakis

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



Outline

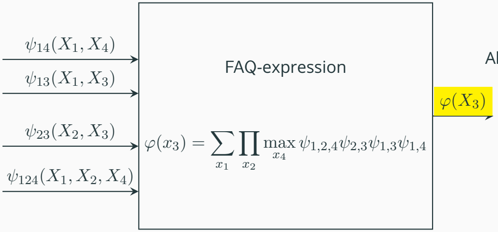
Warm-up Examples

Sum-product Queries

Variable Elimination and Tree Decomposition

Functional Aggregate Queries

Functional Aggregate Query



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

Output $\varphi : \prod_{i \in F} \mathsf{Dom}(X_h) \to \mathbf{D}$

- Compute $\varphi(\boldsymbol{x}_{[f]}) = \bigoplus^{(f+1)}_{x_{f+1} \in \mathsf{Dom}(X_{f+1})} \cdots \bigoplus^{(n-1)}_{x_{n-1} \in \mathsf{Dom}(X_{n-1})} \bigoplus^{(n)}_{x_n \in \mathsf{Dom}(X_n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\boldsymbol{x}_S)$
- Every aggregate $\bigoplus^{(i)}$
  - either $\left(\mathbf{D}, \bigoplus^{(i)}, \bigotimes\right)$ is a *commutative semiring* semiring agg
  - or $\bigoplus^{(i)} = \bigotimes$ product agg

FAQ **example:** CSP **a.k.a. Boolean Conjunctive Query**

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

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

- Constraint $\psi_S : \prod_{i \in S} \text{Dom}(X_h) \to \{\text{true}, \text{false}\}$ with *support* S

FAQ **example:** #CSP **= Join cardinality query**

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

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

- Constraints $\psi_S : \prod_{i \in S} \text{Dom}(X_h) \to \{0, 1\}$, where

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

FAQ **example: Conjunctive Query Evaluation –** CQE

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

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

FAQ **example:** DB Aggregate Query – **count**

```
SELECT R.a, 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 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

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

FAQ **example:** DB Aggregate Query **– sum**

```
SELECT R.a, 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 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

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

FAQ **example: Matrix Chain Multiplication**

Let $\boldsymbol{A}_h = (a^{(i)}_{x,y})$ 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 instance:
  - Variables: $\mathrm{Dom}(X_h) = [p_h]$, $i \in \{0, \ldots, k\}$
  - Factors: $\psi_{i,i+1}(x_h, x_{i+1}) = a^{(i)}_{x_h, x_{i+1}}$
  - Compute new function $\varphi : [p_0] \times [p_k] \to \mathbf{D}$

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

FAQ **Examples: Inference in Probabilistic Graphical Models**

- In PGMs, factors are also called *potential functions*
- Typical inference/learning tasks on PGMs
  - DENS $\text{Prob}(\boldsymbol{x})$
  - MAR (marginal) $\text{Prob}(\boldsymbol{x}_A)$
  - COND (conditional) $\text{Prob}(\boldsymbol{x}_A \mid \boldsymbol{x}_B)$
    - A special case is $\text{Prob}(\boldsymbol{x}_A \mid \boldsymbol{x}_{[n]-A})$
    - likelihood ($\text{Prob}(\boldsymbol{D} \mid \boldsymbol{\theta})$) posterior ($\text{Prob}(\boldsymbol{\theta} \mid \boldsymbol{D})$)
  - MPE (most probable explanation, mode) $\arg\max_{\boldsymbol{x}_A} \text{Prob}(\boldsymbol{x}_A \mid \boldsymbol{x}_{[n]-A})$
  - MAP (maximum a posteriori, also a mode) $\arg\max_{\boldsymbol{x}_A} \text{Prob}(\boldsymbol{x}_A \mid \boldsymbol{x}_B)$
- These problems are all FAQ

FAQ **example: Quantified Conjunctive Queries**

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

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

- FAQ instance on $(\{0,1\}, \{\max, \times\}, \times)$
- Compute the function $\varphi(x_1, \cdots, x_f) = \bigoplus^{(f+1)}_{x_{f+1} \in \{0,1\}} \cdots \bigoplus^{(n)}_{x_n \in \{0,1\}} \prod_{S \in \mathcal{E}} \psi_S(\boldsymbol{x}_S)$, where

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

FAQ example: # Quantified Conjunctive Queries

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

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

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

- FAQ instance on $(\{0,1\} \cup \mathbb{R}_+, \{\max, \times, +\}, \times)$
- Compute the constant $\varphi = \sum_{x_1} \cdots \sum_{x_f} \bigoplus^{(f+1)}_{x_{f+1} \in \{0,1\}} \cdots \bigoplus^{(n)}_{x_n \in \{0,1\}} \prod_{S \in \mathcal{E}} \psi_S(\boldsymbol{x}_S)$ where

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

Many more FAQ examples

• Boolean conjunctive query evaluation	Boolean semiring
• SAT	Boolean semiring
• Quantifier-free conjunctive query evaluation	Set semiring
• k-colorability	Boolean semiring
• Permanent	Sum-Product semiring
• Partition function	Sum-Product semiring

Yet many more FAQ examples

- Discrete Fourier Transform
- Hollant Problem (as in Holographic algorithms)
- Graph Homomorphism Problem
- Weighted CSP
- List recoverable codes
- LDPC codes
- etc.

InsideOut = VE + indicator-projections

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

InsideOut = VE + indicator-projections

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

InsideOut = VE + indicator-projections

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

InsideOut = VE + indicator-projections

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

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

InsideOut = VE + indicator-projections

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

Theorem (Runtime of InsideOut**)**

For Sum-Product, InsideOut *runs in time* $\tilde{O}$ *of*

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

InsideOut = VE + indicator-projections

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

Theorem (Runtime of InsideOut**)**

For Sum-Product, InsideOut *runs in time* $\tilde{O}$ *of*

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

Corollary (New (textbook?) result)

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

InsideOut = VE + indicator-projections

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

Theorem (Runtime of InsideOut**)**

For Sum-Product, InsideOut *runs in time* $\tilde{O}$ *of*

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

Corollary (Grohe-Marx (SODA'06))

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

InsideOut = VE + indicator-projections

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

Theorem (Runtime of InsideOut**)**

For Sum-Product, InsideOut *runs in time* $\tilde{O}$ *of*

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

Corollary (Grohe-Marx (SODA'06))

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

InsideOut = VE + indicator-projections

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

Theorem (Runtime of InsideOut**)**

For Sum-Product, InsideOut *runs in time* $\tilde{O}$ *of*

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

Corollary (New?)

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

InsideOut = VE + indicator-projections

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

Theorem (Runtime of InsideOut**)**

For Sum-Product, InsideOut *runs in time* $\tilde{O}$ *of*

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

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

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

InsideOut for FAQ Queries

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

InsideOut **for** FAQ **Queries**

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

InsideOut for FAQ Queries

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

InsideOut for FAQ Queries

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

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

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

InsideOut for FAQ Queries

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

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

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** FAQ **Queries**

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

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

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

InsideOut for FAQ Queries

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

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

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** FAQ **Queries**

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

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

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** FAQ **Queries**

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

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

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

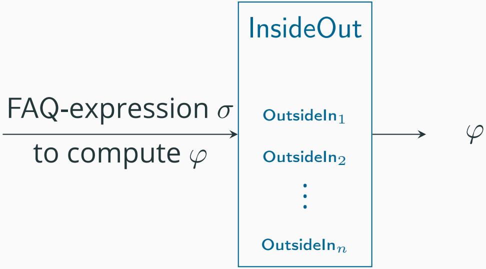
$$\varphi = \cdots \bigoplus_{x_{n-1}}^{(n-1)} \left(\bigotimes_{S \in \mathcal{E}} \bigoplus_{x_n}^{(n)} \psi_S(\mathbf{x}_S) \right)$$

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

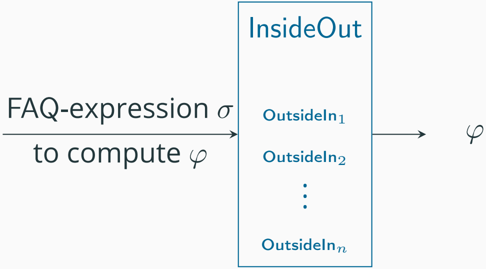
Some Corollaries

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

The FAQ Algorithmic Framework

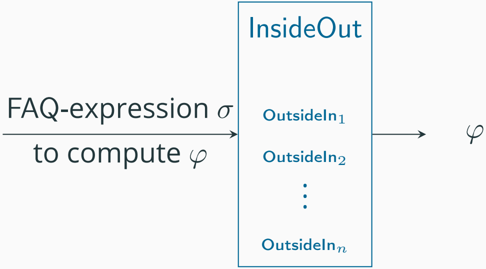


The FAQ Algorithmic Framework



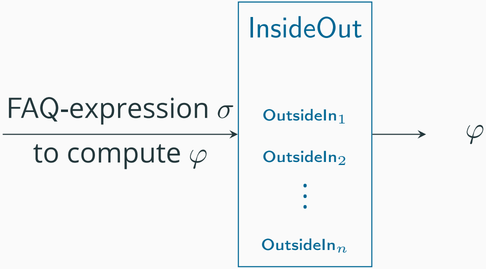
- InsideOut represents dynamic programming

The FAQ Algorithmic Framework



- InsideOut represents dynamic programming
  - is variable elimination ([ZP94, Dechter95]) with new twists

The FAQ Algorithmic Framework



- InsideOut represents dynamic programming
  - is variable elimination ([ZP94, Dechter95]) with new twists
- OutsideIn represents backtracking search

WCOJ

The FAQ Algorithmic Framework

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

FAQ-expression σ to compute φ $\longrightarrow$ **InsideOut** [OutsideIn_1, OutsideIn_2, $\vdots$, OutsideIn_n] $\longrightarrow$ φ

- InsideOut represents dynamic programming
  - is variable elimination ([ZP94, Dechter95]) with new twists
- OutsideIn represents backtracking search

WCOJ

The FAQ Algorithmic Framework

FAQ-expression σ

to compute φ

InsideOut

OutsideIn$_1$

OutsideIn$_2$

$\vdots$

OutsideIn$_n$

φ

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

faqw(σ) is the "width" of σ

FAQ-analog of

fractional hypertree width

- InsideOut represents dynamic programming
  - is variable elimination ([ZP94, Dechter95]) with new twists
- OutsideIn represents backtracking search

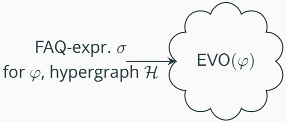
WCOJ

Optimizing the FAQ Query Plans

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

Optimizing the FAQ Query Plans

EVO(φ) $=$ set of expressions
"semantically equivalent" to φ



Optimizing the FAQ **Query Plans**

EVO(φ) = set of expressions
"semantically equivalent" to φ

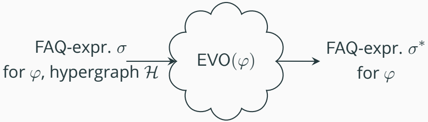
FAQ-expr. σ
for φ, hypergraph $\mathcal{H}$ $\rightarrow$ EVO(φ)

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

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

Optimizing the FAQ Query Plans

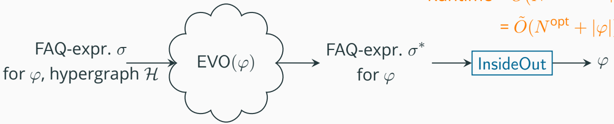
$$\sigma^* = \arg\min_{\tau \in \mathsf{EVO}(\varphi)} \mathsf{faqw}(\tau)$$



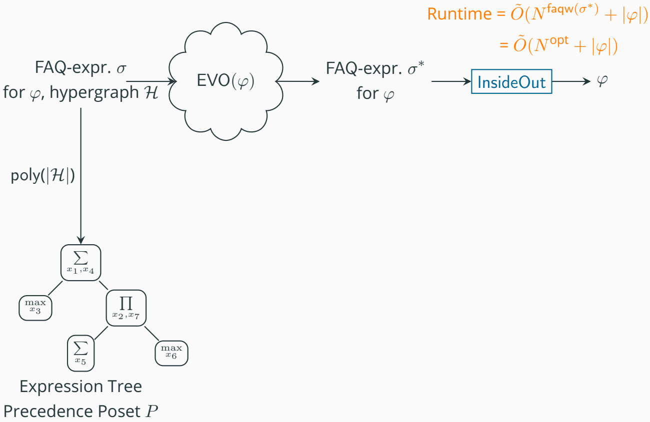
Optimizing the FAQ Query Plans

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

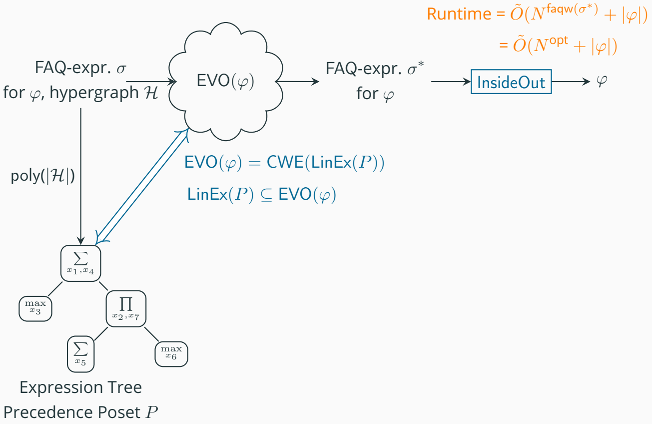
$$\text{Runtime} = \tilde{O}(N^{\mathsf{faqw}(\sigma^*)} + |\varphi|)$$
$$= \tilde{O}(N^{\mathsf{opt}} + |\varphi|)$$



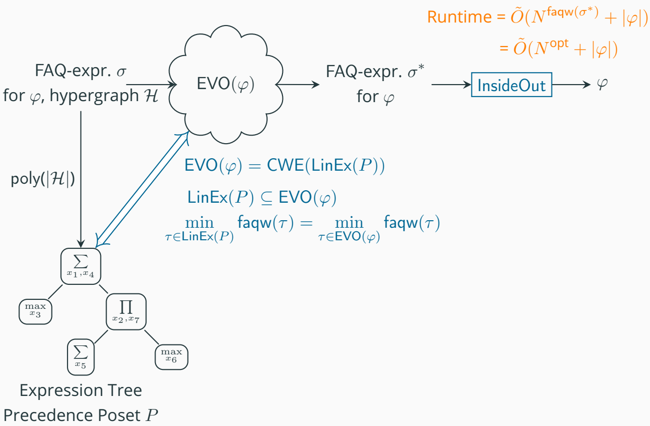
Optimizing the FAQ Query Plans



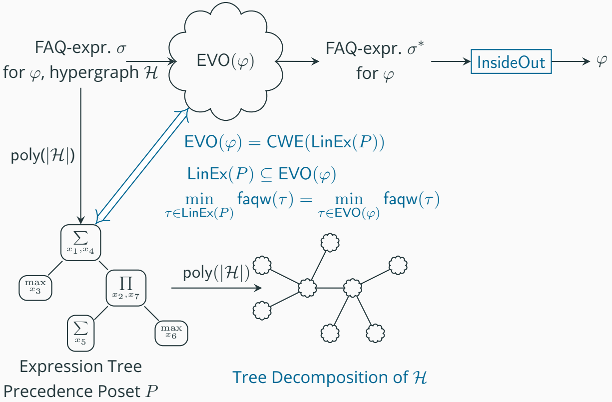
Optimizing the FAQ Query Plans



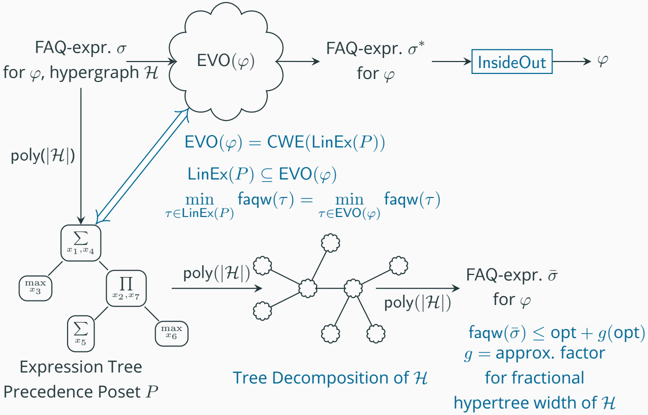
Optimizing the FAQ Query Plans



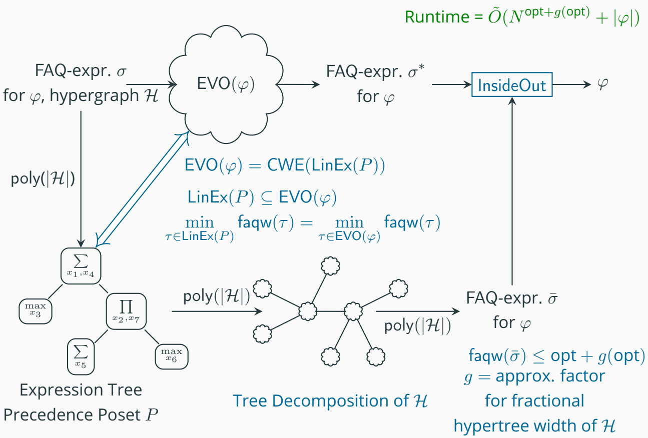
Optimizing the FAQ Query Plans



Optimizing the FAQ Query Plans



Optimizing the FAQ Query Plans



Some References

Yannakakis 81 Mihalis Yannakakis: Algorithms for Acyclic Database Schemes. VLDB 1981: 82-94

Pearl 82 Pearl, Judea. “Reverend Bayes on inference engines: A distributed hierarchical approach”). AAAI-82

ZP 1994 Zhang, N.L., Poole, D.:A Simple Approach to Bayesian Network Computations.In: 7th Canadian Conference on Artificial Intelligence,pp. 171–178

Dechter 1995 Rina Dechter: Bucket Elimination: A Unifying Framework for Probabilistic Inference. Learning in Graphical Models 1998: 75-104

GM 06 Martin Grohe, Dániel Marx: Constraint solving via fractional edge covers. SODA 2006: 289-298

GGLS 16 Georg Gottlob, Gianluigi Greco, Nicola Leone, Francesco Scarcello: Hypertree Decompositions: Questions and Answers. PODS 2016: 57-74

CD’16 Hubie Chen, Victor Dalmau: Decomposing Quantified Conjunctive (or Disjunctive) Formulas. SIAM J. Comput. 45(6): 2066-2086 (2016)

ANR 16 Mahmoud Abo Khamis, Hung Q. Ngo, Atri Rudra: FAQ: Questions Asked Frequently. PODS 2016: 13-28

ANOS 19 Mahmoud Abo Khamis, Hung Q. Ngo, Dan Olteanu, Dan Suciu: Boolean Tensor Decomposition for Conjunctive Queries with Negation. ICDT 2019: 21:1-21:19

- Mahmoud Abo Khamis, Ryan R. Curtin, Benjamin Moseley, Hung Q. Ngo, XuanLong Nguyen, Dan Olteanu, Maximilian Schleich: On Functional Aggregate Queries with Additive Inequalities. PODS 2019: 414-431

Many Thanks!