Query Optimization

How to design a Meta-Algorithm that designs algorithms

Mahmoud Abo Khamis

University of Oxford

Sept 2026

Why Query Optimization TODAY?

- DB systems nowadays are expected to be comprehensive **BI** platforms
  - learn from data, make business decisions
- Modern queries are getting **incredibly complex!**
  - Auto-generated by AI Agents or higher-level languages
- Query Optimizers are now the **bottleneck..**

Lessons I learned from Industry (RelationalAI - LogicBlox)

- Producing a query plan with the **right complexity** is *necessary!*
  - No amount of engineering can make for an $O(N^2)$-plan

Lessons I learned from Industry (RelationalAI - LogicBlox)

- Producing a query plan with the **right complexity** is *necessary!*
  - No amount of engineering can make for an $O(N^2)$-plan
- Worst-case scenarios DO happen!
  - A single workload consists of thousands of queries
  - **99%** success rate on query plans $\Rightarrow$ **0%** completed workloads!

Lessons I learned from Industry (RelationalAI - LogicBlox)

- Producing a query plan with the **right complexity** is *necessary!*
  - No amount of engineering can make for an $O(N^2)$-plan
- Worst-case scenarios DO happen!
  - A single workload consists of thousands of queries
  - **99%** success rate on query plans $\Rightarrow$ **0%** completed workloads!
- Real-world Query Optimization requires **real-time accountability** to customers
  - *"Explain this!" "Can you fix it by the morning?!"*
  - *"Can you **guarantee** it won't happen again?"*

Lessons I learned from Industry (RelationalAI - LogicBlox)

- Producing a query plan with the **right complexity** is *necessary!*
  - No amount of engineering can make for an $O(N^2)$-plan
- Worst-case scenarios DO happen!
  - A single workload consists of thousands of queries
  - **99%** success rate on query plans $\Rightarrow$ **0%** completed workloads!
- Real-world Query Optimization requires **real-time accountability** to customers
  - *"Explain this!" "Can you fix it by the morning?!"*
  - *"Can you **guarantee** it won't happen again?"*
- Only a **Rigorous Theory** can give you that
  - LLMs can't (yet)
    - They can still be *part of* the pipeline but *cannot replace* it

(Futuristic) Terminology

Query	ANY Question/Problem about the input data
Query Plan	ANY Algorithm
Statistics	ANY Prior Knowledge about the input data

(Futuristic) Terminology

Query	ANY Question/Problem about the input data
Query Plan	ANY Algorithm
Statistics	ANY Prior Knowledge about the input data
Query Optimizer	A Meta-Algorithm, i.e., a Universal Algorithm Designer

(Futuristic) Terminology

Query	ANY Question/Problem about the input data
Query Plan	ANY Algorithm
Statistics	ANY Prior Knowledge about the input data
Query Optimizer	A Meta-Algorithm, i.e., a Universal Algorithm Designer **Input:** Query / Problem (+ Statistics) $\Rightarrow$ **Output:** Query Plan / Algorithm

Examples:

Input:		**Output:**
Query / Problem (+ Statistics)	$\Rightarrow$	Query Plan / Algorithm

Examples:

Input:		**Output:**
Query / Problem (+ Statistics)	$\Rightarrow$	Query Plan / Algorithm
Shortest-path problem	$\Rightarrow$	Floyd-Warshall algorithm

Examples:

Input:		**Output:**
Query / Problem (+ Statistics)	$\Rightarrow$	Query Plan / Algorithm
Shortest-path problem	$\Rightarrow$	Floyd-Warshall algorithm
A matrix product $A \times B \times C \times D$ & dimensions	$\Rightarrow$	$A \times ((B \times C) \times D)$

Examples:

Input:		**Output:**
Query / Problem (+ Statistics)	$\Rightarrow$	Query Plan / Algorithm
Shortest-path problem	$\Rightarrow$	Floyd-Warshall algorithm
A matrix product $A \times B \times C \times D$ & dimensions	$\Rightarrow$	$A \times ((B \times C) \times D)$
“Given a graph of size N, how many 6-cycles are there?”	$\Rightarrow$	An algorithm matching the state-of-the-art *human-designed* algorithm

Challenges

1. Accurate ***cost estimation*** of query plans
   - *How does the optimizer know the complexity of Floyd-Warshall?*
2. ***Vastness of the space*** of query plans
   - *How can the optimizer stumble upon Floyd-Warshall out of all possible algorithms?*

Challenges

1. Accurate ***cost estimation*** of query plans
    - *How does the optimizer know the complexity of Floyd-Warshall?*
2. ***Vastness of the space*** of query plans
    - *How can the optimizer stumble upon Floyd-Warshall out of all possible algorithms?*
3. Emerging applications leading to ***new query classes***
    - *"Great job about shortest paths!"*
    - *"Now when can your optimizer do my tax return?!"*

Outline

- Start with a (relatively) simple class of queries
  - **Research Thread 1:** Using **Information Theory** for
    - Provably Accurate Cost Estimation
    - Designing Optimal Query Plans
- Expanding the class of queries
  - **Research Thread 2:** Using **Algebra** for
    - Amplifying the Power of Query Languages
    - Re-using Algorithms in Different Applications

Research Thread 1

An **Information-Theoretic** Framework for Query Optimization

Conjunctive Queries (CQs)

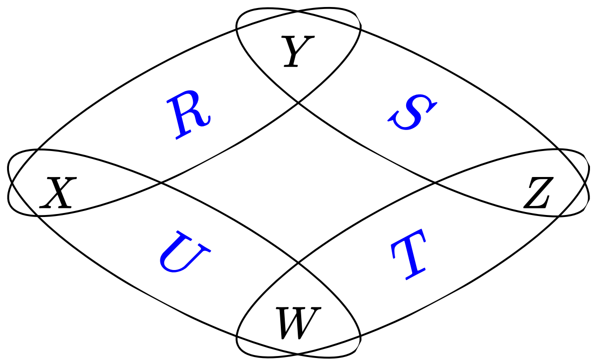
Conjunctive Queries (CQs)

- **Input**
  - A set of *variables* $\boldsymbol{X}$
  - A set of *relations* $R_1(\boldsymbol{X}_1), \ldots, R_k(\boldsymbol{X}_k)$
  - A *query* $Q(\boldsymbol{X}) = R_1(\boldsymbol{X}_1) \wedge \ldots \wedge R_k(\boldsymbol{X}_k)$

Conjunctive Queries (CQs)

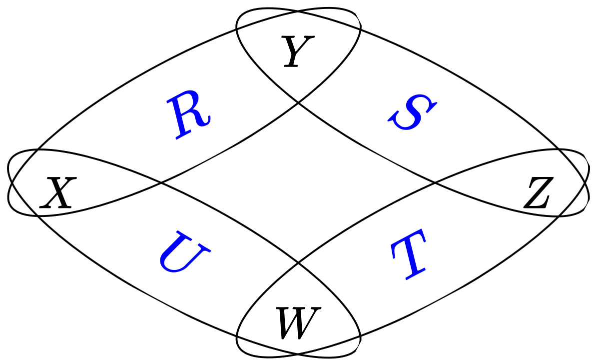
- **Input**
  - A set of *variables* $\boldsymbol{X}$
  - A set of *relations* $R_1(\boldsymbol{X}_1), \ldots, R_k(\boldsymbol{X}_k)$
  - A *query* $Q(\boldsymbol{X}) = R_1(\boldsymbol{X}_1) \wedge \ldots \wedge R_k(\boldsymbol{X}_k)$
- **Target**
  - Compute $Q(\boldsymbol{X})$

Conjunctive Queries (CQs) - Example



$$Q(X,Y,Z,W) = R(X,Y) \wedge S(Y,Z) \wedge T(Z,W) \wedge U(W,X)$$

Conjunctive Queries (CQs) - Example



$$Q(X,Y,Z,W) = R(X,Y) \wedge S(Y,Z) \wedge T(Z,W) \wedge U(W,X)$$

R

X	Y
1	p
1	q
2	p

S

Y	Z
p	3
q	4
q	5

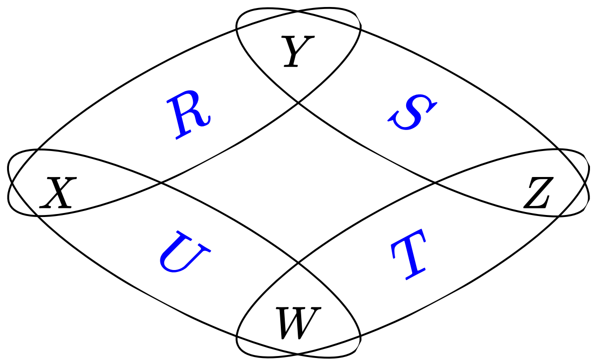
T

Z	W
3	i
5	i
5	j

U

W	X
i	1
j	1
k	2

Conjunctive Queries (CQs) - Example



$$Q(X,Y,Z,W) = R(X,Y) \wedge S(Y,Z) \wedge T(Z,W) \wedge U(W,X)$$

R

X	Y
1	p
1	q
2	p

S

Y	Z
p	3
q	4
q	5

T

Z	W
3	i
5	i
5	j

U

W	X
i	1
j	1
k	2

Q

X	Y	Z	W
1	p	3	i
1	q	5	i
1	q	5	j

Special Case: Graph Pattern Matching (**GPM**)

Special Case: Graph Pattern Matching (GPM)

- **Input**
  - A (directed) graph G of size N

Special Case: Graph Pattern Matching (GPM)

- **Input**
  - A (directed) graph G of size N
- **Target**
  - Search for a specific pattern P in G
    - P could be a triangle, a k-cycle, a k-clique, ...

Special Case: Graph Pattern Matching (GPM)

- **Input**
  - A (directed) graph G of size N

- **Target**
  - Search for a specific pattern P in G
    - P could be a triangle, a k-cycle, a k-clique, ...

- **Variants**
  - *Detection:* Does G contain P?
  - *Counting:* How many occurrences of P does G contain?
  - *Listing:* List all occurrences
  - ...

CQs are a *Vast* Generalization of GPM

$$Q(\boldsymbol{X}) = R_1(\boldsymbol{X}_1) \wedge \ldots \wedge R_k(\boldsymbol{X}_k)$$

CQs are a *Vast* Generalization of GPM

$$Q(\boldsymbol{X}) = R_1(\boldsymbol{X}_1) \wedge \ldots \wedge R_k(\boldsymbol{X}_k)$$

- $R_1, \ldots, R_k$ can be different
  - *"Colored" graphs*

CQs are a *Vast* Generalization of GPM

$$Q(\boldsymbol{X}) = R_1(\boldsymbol{X}_1) \wedge \ldots \wedge R_k(\boldsymbol{X}_k)$$

- $R_1, \ldots, R_k$ can be different
  - *"Colored" graphs*
- $R_1, \ldots, R_k$ are not necessarily binary relations
  - *Hypergraphs*

CQs are a *Vast* Generalization of GPM

$$Q(\boldsymbol{X}) = R_1(\boldsymbol{X}_1) \wedge \ldots \wedge R_k(\boldsymbol{X}_k)$$

- $R_1, \ldots, R_k$ can be different
  - *"Colored" graphs*
- $R_1, \ldots, R_k$ are not necessarily binary relations
  - *Hypergraphs*
- We could have many *statistics* besides N
  - Functional dependencies, e.g. city → country in R(city, country, company)
  - Degree statistics, e.g. average or max number of cities per country
  - Correlation among data columns
  - ...

Prior results

- Tailored to a specific query

Prior results

- Tailored to a specific query
- **Combinatorial:**
  - Triangle $\Rightarrow O(N^{\frac{3}{2}})$ [Itai & Rodeh-**STOC'77**]
  - k-cycle$\Rightarrow O(N^{2-\frac{1}{\lceil k/2 \rceil}})$ [Alon, Yuster & Zwick-**Algorithmica'79**]
  - ...

Prior results

- Tailored to a specific query
- **Combinatorial:**
  - Triangle $\Rightarrow O(N^{\frac{3}{2}})$ [Itai & Rodeh-**STOC'77**]
  - k-cycle$\Rightarrow O(N^{2-\frac{1}{\lceil k/2 \rceil}})$ [Alon, Yuster & Zwick-**Algorithmica'79**]
  - ...
- **Non-combinatorial:**

ω is the Matrix Multiplication exponent

  - Triangle $\Rightarrow O(N^{\frac{2\omega}{\omega+1}})$ [Alon, Yuster & Zwick-**Algorithmica'79**]
  - 4-cycle $\Rightarrow \tilde{O}(N^{\frac{4\omega-1}{2\omega+1}})$ [Yuster & Zwick-**SODA'04**] [Dalirrooyfard et al-**STOC'19**]
  - 5-clique $\Rightarrow O(N^{\frac{\omega}{2}+1})$ [Dalirrooyfard et al-**STOC'24**]
  - ...

Our Results: A Unified Algorithm, **PANDA**

- subsumes *all known* results
- discovers *new* ones
- remains the *state-of-the-art* for ANY Q
- handles arbitrary *statistics*

Our Results: A Unified Algorithm, PANDA

- subsumes *all known* results
- discovers *new* ones
- remains the *state-of-the-art* for ANY Q
- handles arbitrary *statistics*

Combinatorial version

[M. Abo Khamis, Hung Ngo, Dan Suciu - PODS'17]

- *Any* $Q \Rightarrow O(N^{\text{subw}(Q)})$
  - Triangle $\Rightarrow O(N^{\frac{3}{2}})$ [IR-STOC'77]
  - k-cycle$\Rightarrow O(N^{2-\frac{1}{\lceil k/2 \rceil}})$ [AYZ-Algorithmica'79]
  - ...

Our Results: A Unified Algorithm, PANDA

- subsumes *all known* results
- discovers *new* ones
- remains the *state-of-the-art* for ANY Q
- handles arbitrary *statistics*

Combinatorial version

[M. Abo Khamis, Hung Ngo, Dan Suciu - PODS'17]

- *Any* $Q \Rightarrow O(N^{\text{subw}(Q)})$
  - Triangle $\Rightarrow O(N^{\frac{3}{2}})$ [IR-STOC'77]
  - k-cycle$\Rightarrow O(N^{2-\frac{1}{\lceil k/2 \rceil}})$ [AYZ-Algorithmica'79]
  - ...

Non-Combinatorial version

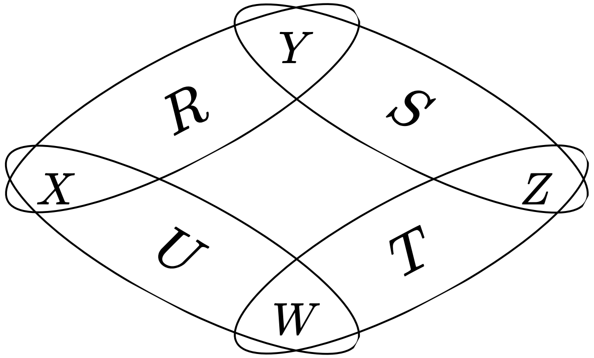
[M. Abo Khamis, Xiao Hu, Dan Suciu - PODS'25]

- *Any* $Q \Rightarrow O(N^{\omega\text{-subw}(Q)})$
  - Triangle $\Rightarrow O(N^{\frac{2\omega}{\omega+1}})$ [AYZ-Algorithmica'79]
  - 4-cycle $\Rightarrow \tilde{O}(N^{\frac{4\omega-1}{2\omega+1}})$ [YZ-SODA'04] [DVW-STOC'19]
  - 5-clique $\Rightarrow O(N^{\frac{\omega}{2}+1})$ [DMWX-STOC'24]
  - ...

PANDA is a Meta-Algorithm

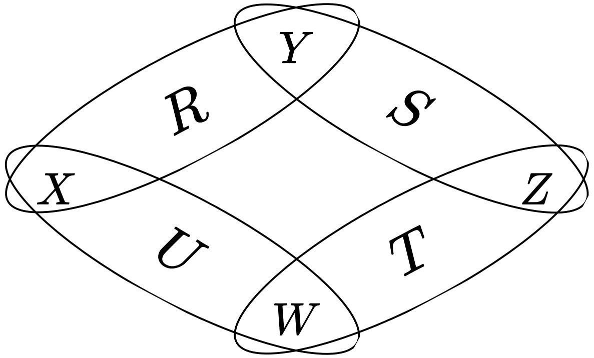
- **Input:**

PANDA is a Meta-Algorithm



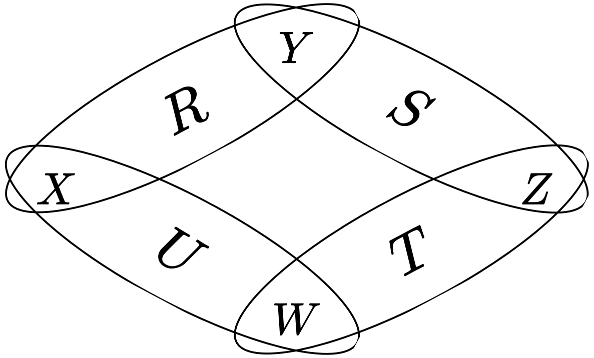
- **Input:**
  - Query, e.g. $Q(X, Y, Z, W) = R(X, Y) \wedge S(Y, Z) \wedge T(Z, W) \wedge U(W, X)$

PANDA is a Meta-Algorithm



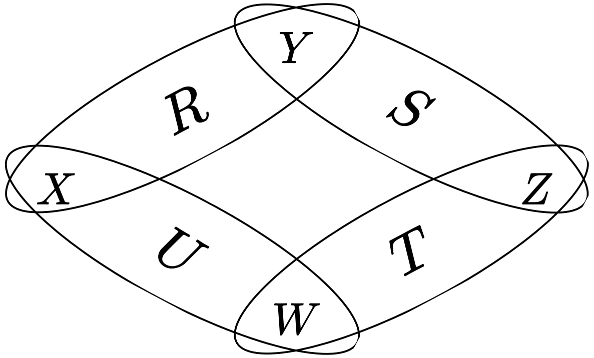
- **Input:**
  - Query, e.g. $Q(X,Y,Z,W) = R(X,Y) \wedge S(Y,Z) \wedge T(Z,W) \wedge U(W,X)$
  - Statistics, e.g.
    - $|R|, |S|, |T|, |U| \leq N$
    - Functional dependency: $W \to X$
    - Degree Constraint: $\mathrm{maxdeg}(W|X) \leq C$
    - ...

PANDA is a Meta-Algorithm



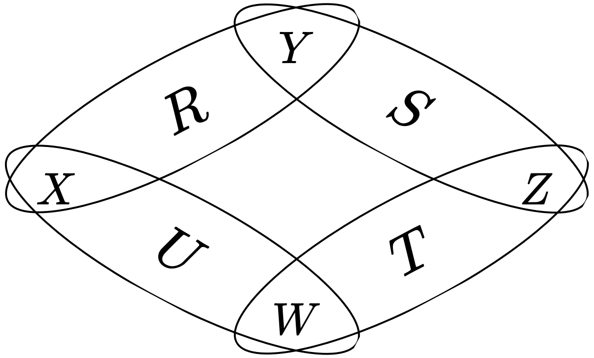
- **Input:**
  - Query, e.g. $Q(X,Y,Z,W) = R(X,Y) \wedge S(Y,Z) \wedge T(Z,W) \wedge U(W,X)$
  - Statistics, e.g.
    - $|R|, |S|, |T|, |U| \leq N$
    - Functional dependency: $W \to X$
    - Degree Constraint: $\text{maxdeg}(W|X) \leq C$
    - ...
- **Output:**
  - An $O(N^{3/2} \cdot \sqrt{C})$-time algorithm
  - An upper bound, e.g., $|Q| \leq N^{3/2} \cdot \sqrt{C}$

Upper Bound



[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Upper Bound

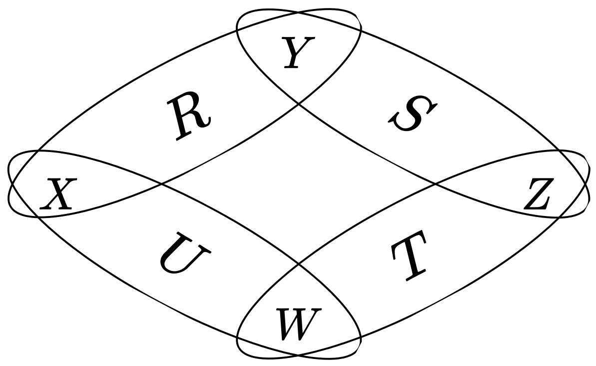


Database World:

- *maximize* $|Q|$
- *over all* **databases** R, S, T, U
- *subject to* statistics:

[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Upper Bound

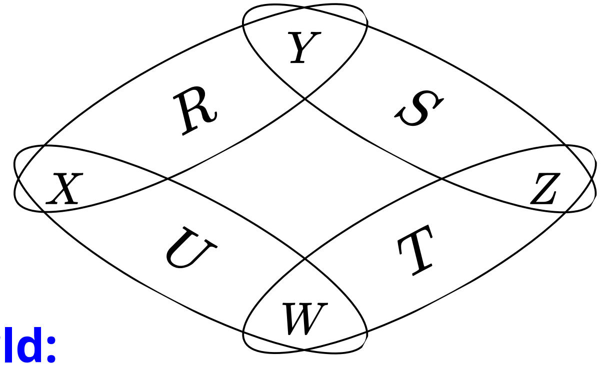


Database World:

- *maximize* $|Q|$
- *over all* **databases** R, S, T, U
- *subject to* statistics:
  - $|R|, |S|, |T|, |U| \leq N$
  - Functional dependency: $W \to X$
  - Degree Constraint: $\mathrm{maxdeg}(W|X) \leq C$
  - ...

[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Upper Bound



Database World:

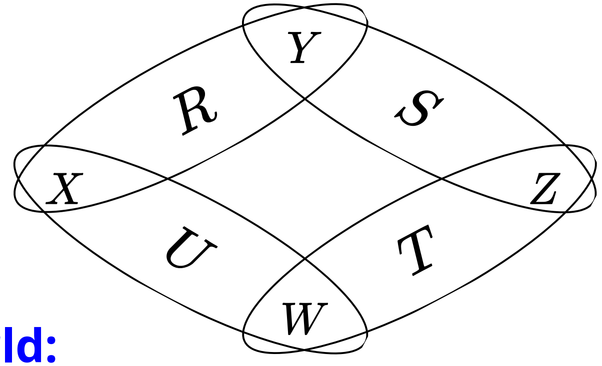
- *maximize* $|Q|$
- *over all* **databases** R, S, T, U
- *subject to* statistics:
  - $|R|, |S|, |T|, |U| \leq N$
  - Functional dependency: $W \rightarrow X$
  - Degree Constraint: $\mathrm{maxdeg}(W|X) \leq C$
  - ...

Information-Theory World:

- *maximize* $h(XYZW)$
- *over all* **"entropies"** h
- *subject to* statistics:

[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Upper Bound



Database World:

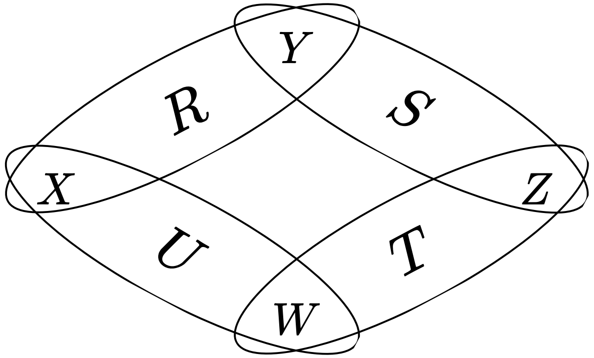
- *maximize* $|Q|$
- *over all* **databases** R, S, T, U
- *subject to* statistics:
  - $|R|, |S|, |T|, |U| \leq N$
  - Functional dependency: $W \to X$
  - Degree Constraint: $\mathrm{maxdeg}(W|X) \leq C$
  - ...

Information-Theory World:

- *maximize* $h(XYZW)$
- *over all* **"entropies"** h
- *subject to* statistics:
  - $h(XY), h(YZ), h(ZW), h(WX) \leq \log N$
  - $h(X|W) = 0$
  - $h(W|X) \leq \log C$
  - ...

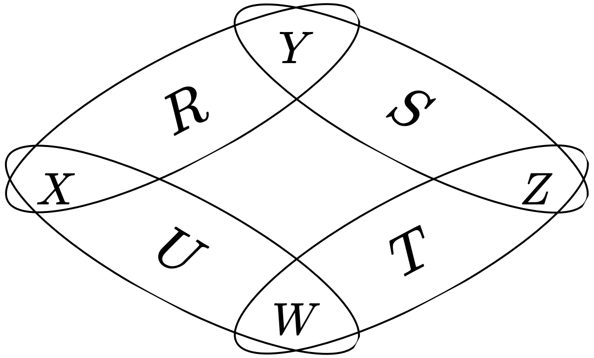
[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Why does the upper bound work?



[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Why does the upper bound work?



R

X	Y
1	p
1	q
2	p

S

Y	Z
p	3
q	4
q	5

T

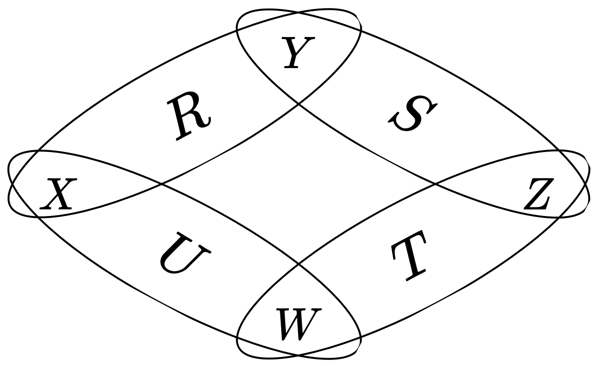
Z	W
3	i
5	i
5	j

U

W	X
i	1
j	1
k	2

[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Why does the upper bound work?



R

X	Y
1	p
1	q
2	p

S

Y	Z
p	3
q	4
q	5

T

Z	W
3	i
5	i
5	j

U

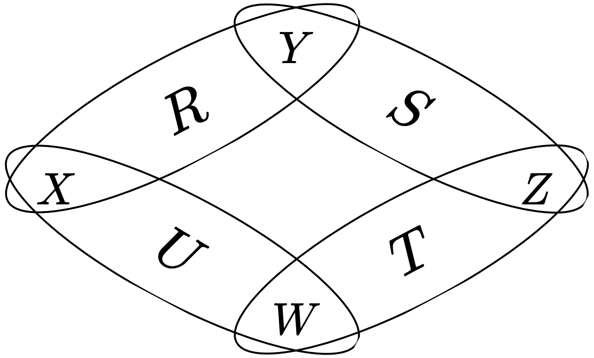
W	X
i	1
j	1
k	2

Q

X	Y	Z	W
1	p	3	i
1	q	5	i
1	q	5	j

[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Why does the upper bound work?



R

X	Y
1	p
1	q
2	p

S

Y	Z
p	3
q	4
q	5

T

Z	W
3	i
5	i
5	j

U

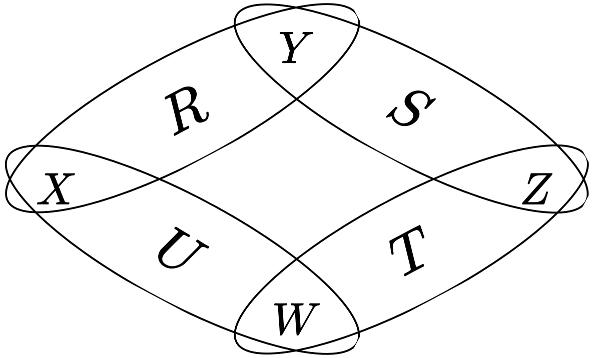
W	X
i	1
j	1
k	2

Q

X	Y	Z	W	
1	p	3	i	1/3
1	q	5	i	1/3
1	q	5	j	1/3

[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Why does the upper bound work?



R

X	Y	
1	p	1/3
1	q	2/3
2	p	0

S

Y	Z	
p	3	1/3
q	4	0
q	5	2/3

T

Z	W	
3	i	1/3
5	i	1/3
5	j	1/3

U

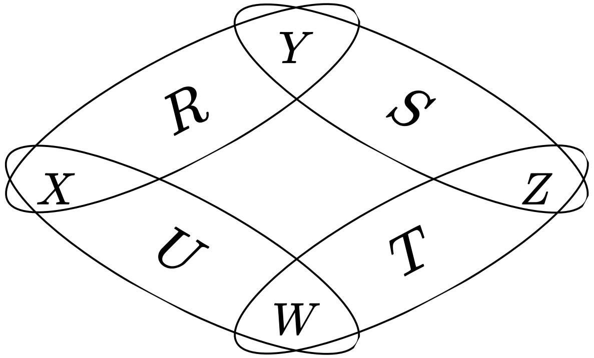
W	X	
i	1	2/3
j	1	1/3
k	2	0

Q

X	Y	Z	W	
1	p	3	i	1/3
1	q	5	i	1/3
1	q	5	j	1/3

[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Why does the upper bound work?



R

X	Y	
1	p	1/3
1	q	2/3
2	p	0

S

Y	Z	
p	3	1/3
q	4	0
q	5	2/3

T

Z	W	
3	i	1/3
5	i	1/3
5	j	1/3

U

W	X	
i	1	2/3
j	1	1/3
k	2	0

Q

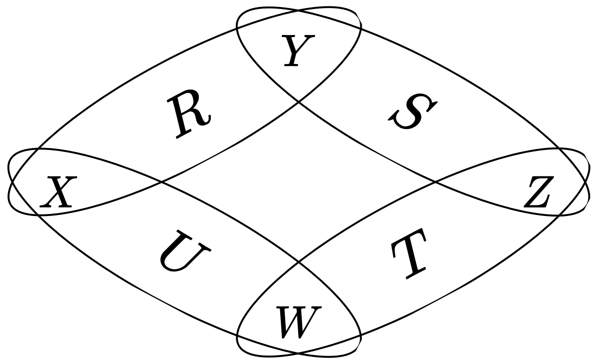
X	Y	Z	W	
1	p	3	i	1/3
1	q	5	i	1/3
1	q	5	j	1/3

Database World:

- $|Q|$
- $|R| \leq N$
- $W \rightarrow X$
- $\text{maxdeg}(W|X) \leq C$
- ...

[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

Why does the upper bound work?



R

X	Y	
1	p	1/3
1	q	2/3
2	p	0

S

Y	Z	
p	3	1/3
q	4	0
q	5	2/3

T

Z	W	
3	i	1/3
5	i	1/3
5	j	1/3

U

W	X	
i	1	2/3
j	1	1/3
k	2	0

Q

X	Y	Z	W	
1	p	3	i	1/3
1	q	5	i	1/3
1	q	5	j	1/3

Database World:

- $|Q|$
- $|R| \leq N$
- $W \to X$
- $\mathrm{maxdeg}(W|X) \leq C$
- ...

Information-Theory World:

- $h(XYZY) = \log|Q|$
- $h(XY) \leq \log N$
- $h(X|W) = 0$
- $h(W|X) \leq \log C$
- ...

[..., AGM08, GLVV'12, ANS'17, JRR'21, ANOS'24, ...]

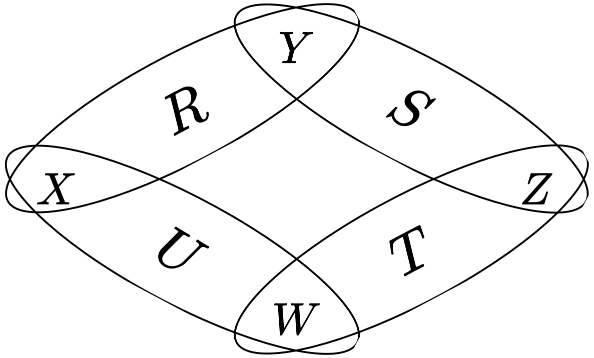
Framework Overview

Maximization Problem → Upper Bound

↓

Algorithm ← Proof

Framework Overview



- *maximize* $h(XYZW)$
- *over all* **"entropies"** h
- *subject to* statistics:
  - $h(XY), \ldots \leq \log N$
  - $h(X|W) = 0$
  - $h(W|X) \leq \log C$

$$|Q| \quad \leq \quad N^{3/2} \cdot \sqrt{C}$$

$$\underbrace{2h(XYZW)}_{=\log |Q|} \leq \underbrace{h(XY)}_{\leq \log N} + \underbrace{h(YZ)}_{\leq \log N} + \underbrace{h(ZW)}_{\leq \log N} + \underbrace{h(X|W)}_{=0} + \underbrace{h(W|X)}_{\leq \log C}$$

- $h(YZ) \to h(Y) + h(Z|Y)$
- $h(XY) + h(Z|Y) \to h(XYZ)$
- $h(XYZ) + h(W|X) \to h(XYZW)$
- $h(Y) + h(ZW) \to h(YZW)$
- $h(YZW) + h(X|W) \to h(XYZW)$

- $S(Y,Z) \to S^h(Y) \cup S^\ell(Y,Z)$ $\quad // \sqrt{N/C}$
- $R(X,Y) \bowtie S^\ell(Y,Z) \to P^\ell(X,Y,Z)$
- $P^\ell(X,Y,Z) \bowtie U(W,X) \to Q^\ell(X,Y,Z,W)$
- $S^h(Y) \bowtie T(Z,W) \to P^h(Y,Z,W)$
- $P^h(Y,Z,W) \bowtie U(W,X) \to Q^h(X,Y,Z,W)$

Interest within Different Communities

- **Theory & Algorithms**
  - Invited *Plenary Talk* at **STOC 2018**
  - Invited to JACM
  - Invited to TheoretiCS
- **Information Theory**
  - AAIT (Algorithmic Aspects of Information Theory) Seminar
  - Dagstuhl Seminar 22301
- **DB Systems**
  - **ACM SIGMOD 2025 Best Paper Award**
  - ACM SIGMOD 2024 *Research Highlight Award*
- **DB Theory**
  - ACM PODS 2024 & 2025 *Distinguished Paper Awards*

Practical Impact

[Haozhe Zhang, Christoph Mayer, Mahmoud Abo Khamis, Dan Olteanu, Dan Suciu - **SIGMOD 25 Best Paper Award**]

- **Q1:** How accurate is our framework in predicting true output sizes?
  - ***Competitors:*** a wide range of size estimators
    - Open-source & commercial, ML-based, etc..
  - **Benchmarks:** IMDB JOB, STATS, Subgraph Matching (DBLP)
  - **Accuracy:** Our framework can be orders of magnitude more accurate
  - **Resources:** Comparable low time and space

Practical Impact

[Haozhe Zhang, Christoph Mayer, Mahmoud Abo Khamis, Dan Olteanu, Dan Suciu - **SIGMOD 25 Best Paper Award**]

- **Q1:** How accurate is our framework in predicting true output sizes?
  - ***Competitors:*** a wide range of size estimators
    - Open-source & commercial, ML-based, etc..
  - **Benchmarks:** IMDB JOB, STATS, Subgraph Matching (DBLP)
  - **Accuracy:** Our framework can be orders of magnitude more accurate
  - **Resources:** Comparable low time and space
- **Q2:** How much does a typical optimizer benefit from our size estimator?
  - Postgres produces query plans as good as those based on true sizes

Key Takeaways So Far…

- PANDA is a Meta-Algorithm that automatically generates algorithms for CQs
- Recovers or improves upon known hand-written algorithms
- Rooted in information theory
- Has applications in DB systems

Research Thread 2

Using **Algebra** to Amplify the Power of Query Languages & Algorithms

Motivation

Math	PL
Algebraic Abstraction **T**	Template **T**
Parameterized Algorithm by **T**	Templated Library
$\mathrm{Algo}_1 + \mathrm{Algo}_2 + \ldots \Rightarrow \mathrm{Algo}<\mathbf{T}>$	$\mathrm{fun}_1 + \mathrm{fun}_2 + \ldots \Rightarrow \mathrm{fun}<\mathbf{T}>$
Generalize proofs	Reuse code

Functional Aggregate Queries (FAQ)

[M. Abo Khamis, Hung Ngo, Atri Rudra - PODS'16 Best Paper - PODS'26 ToT]

Functional Aggregate Queries (FAQ)

[M. Abo Khamis, Hung Ngo, Atri Rudra - PODS'16 Best Paper - PODS'26 ToT]

- A *unified query language* that captures many problems across different domains
  - Databases
    - CQs, #CQs, QCQs
  - Linear Algebra
    - Matrix Chain Multiplication
    - Tensor Computations
  - Probabilistic Graphical Models (PGMs)
    - MAP queries
  - CSP
  - FFT

Functional Aggregate Queries (**FAQ**)

[M. Abo Khamis, Hung Ngo, Atri Rudra - PODS'16 Best Paper - PODS'26 ToT]

- A *unified query language* that captures many problems across different domains
  - Databases
    - CQs, #CQs, QCQs
  - Linear Algebra
    - Matrix Chain Multiplication
    - Tensor Computations
  - Probabilistic Graphical Models (PGMs)
    - MAP queries
  - CSP
  - FFT
- A *unified algorithm* for FAQs, called **InsideOut**

FAQ: Before and After

FAQ: Before and After

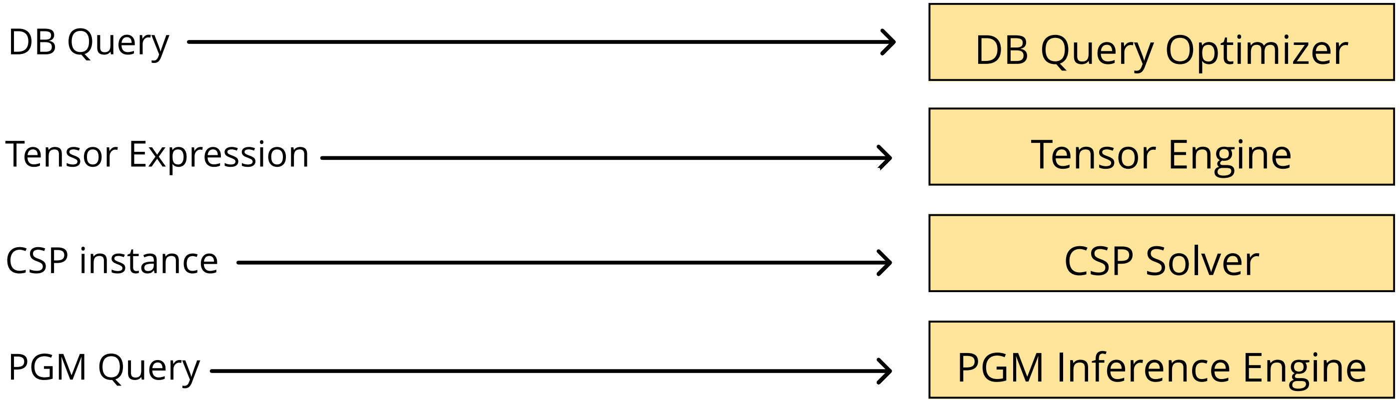
DB Query

Tensor Expression

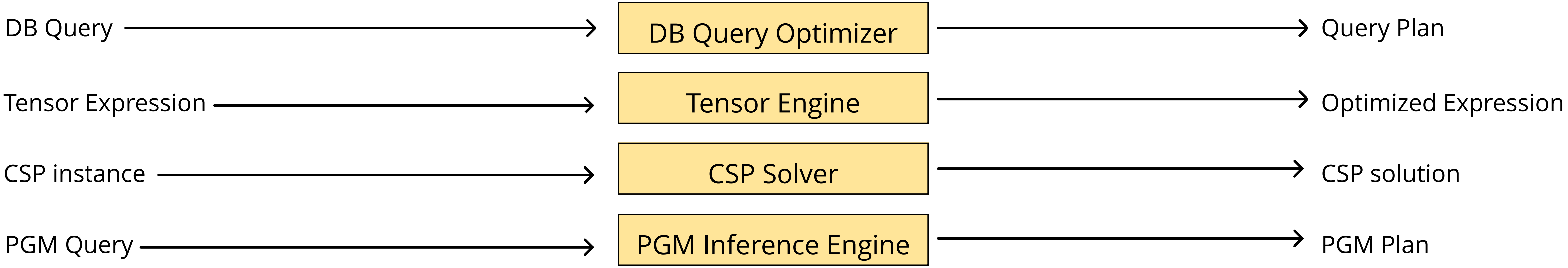
CSP instance

PGM Query

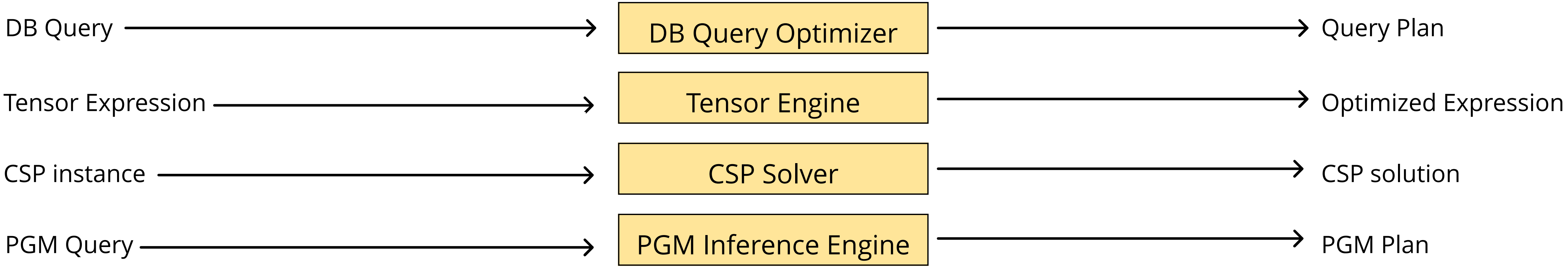
FAQ: Before and After



FAQ: Before and After



FAQ: Before and After



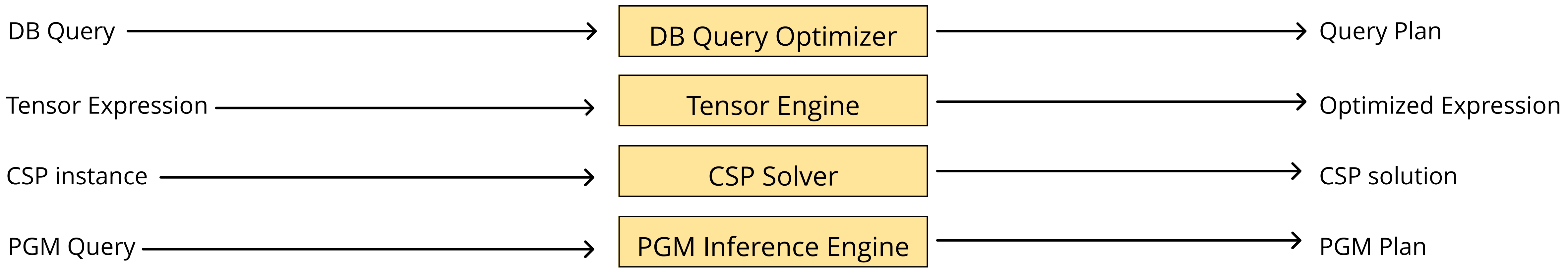
DB Query

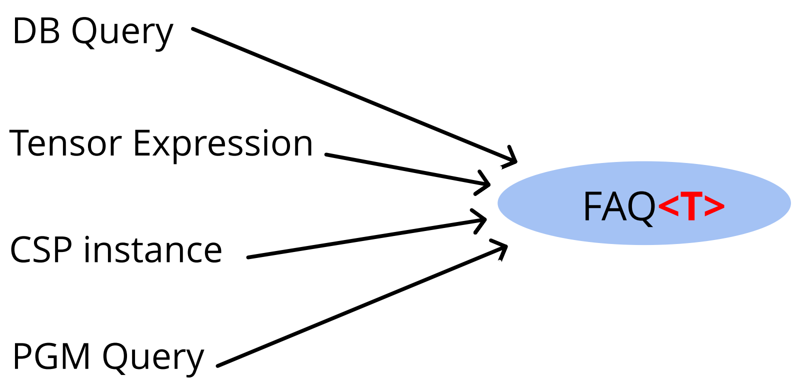
Tensor Expression

CSP instance

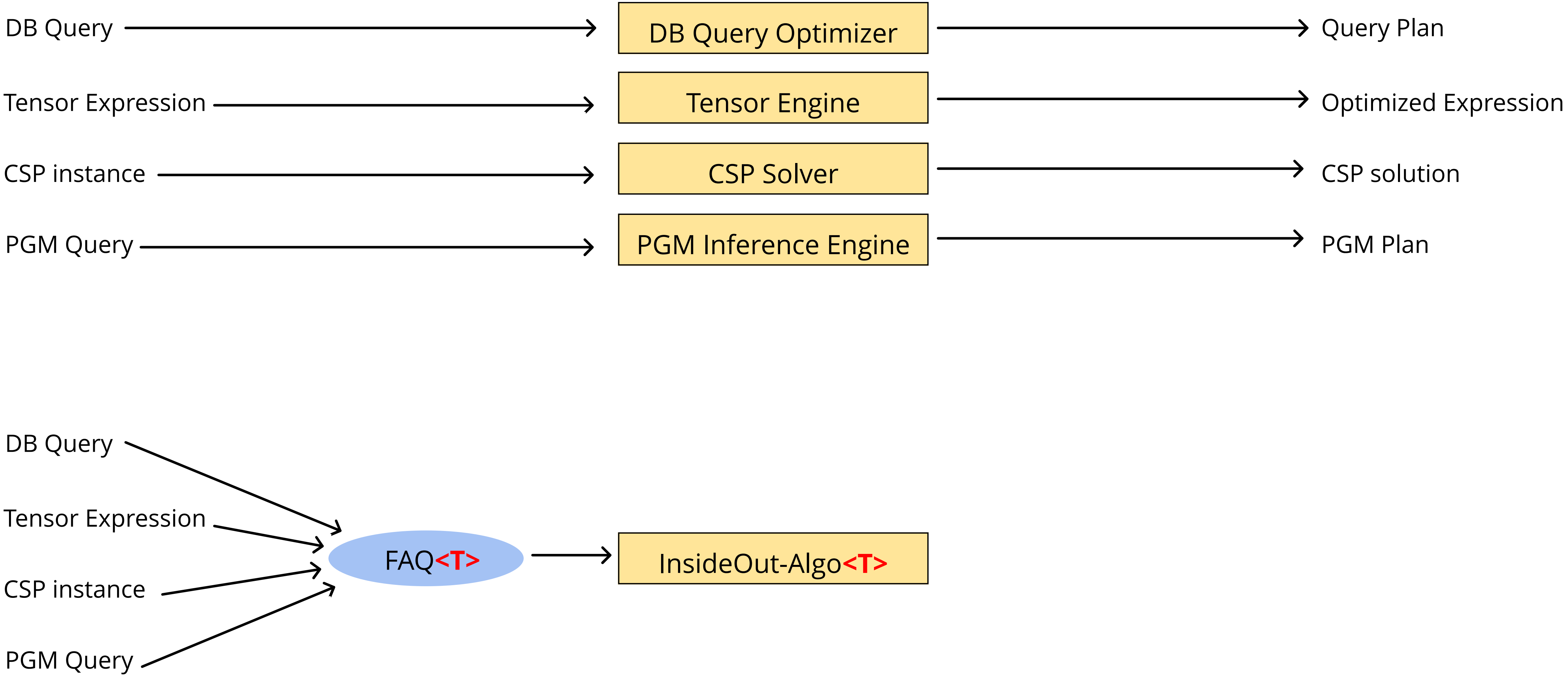
PGM Query

FAQ: Before and After

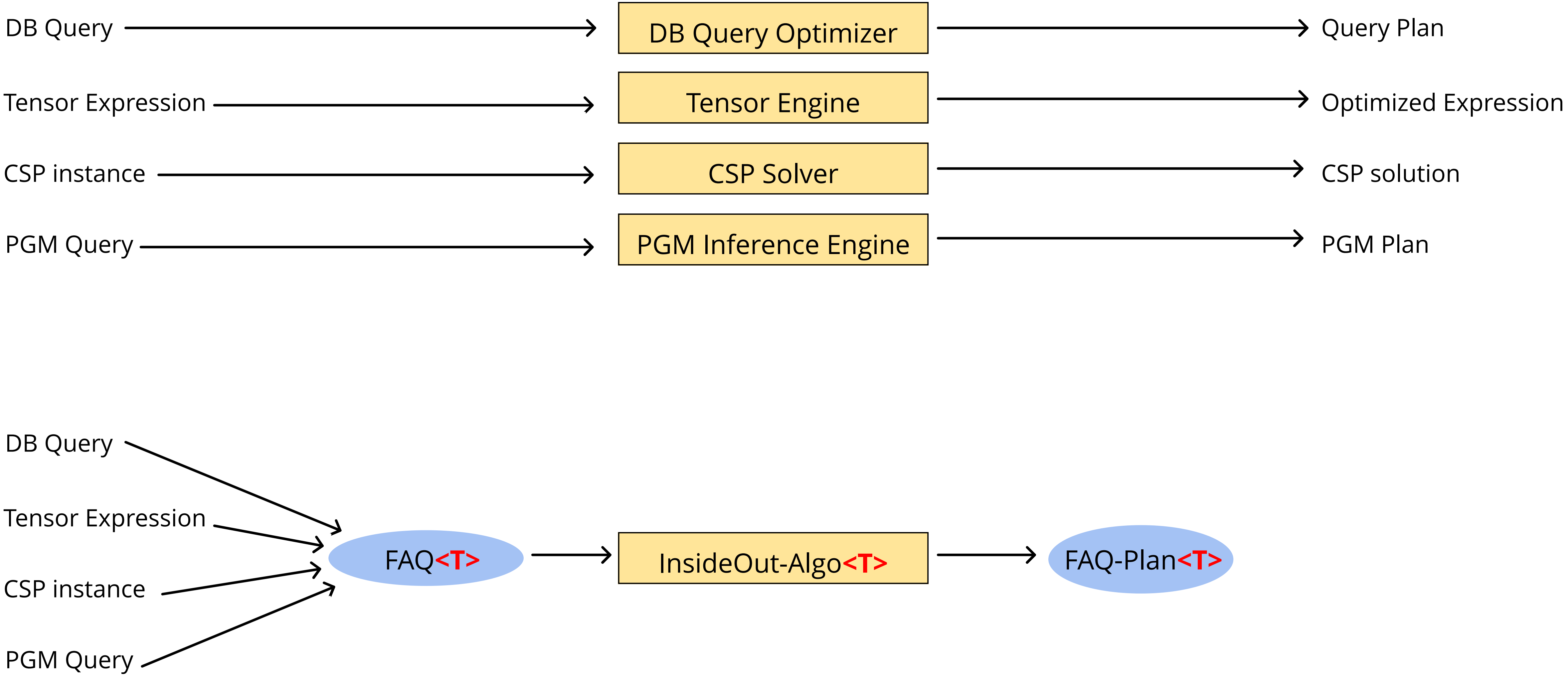




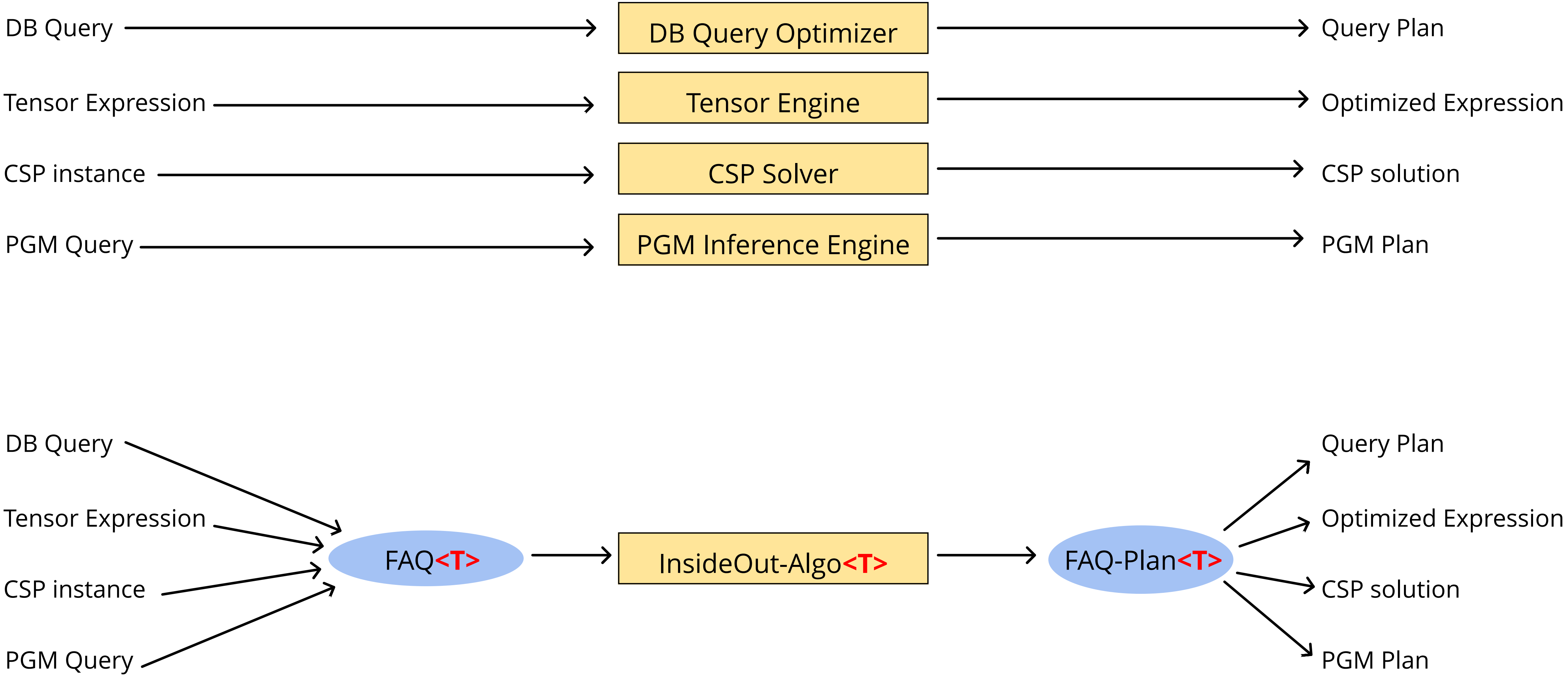
FAQ: Before and After



FAQ: Before and After



FAQ: Before and After



What is an FAQ?

- FAQ**<T>** is a *sum of a product* of functions under a semiring **T** = $(\boldsymbol{D}, \oplus, \otimes)$

What is an FAQ?

- FAQ**<T>** is a *sum of a product* of functions under a semiring **T** = $(\boldsymbol{D}, \oplus, \otimes)$
- Example:
  - $\varphi(x,t) = \bigoplus_{y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(z,t)$

What is an FAQ?

- FAQ<T> is a *sum of a product* of functions under a semiring **T** = $(\boldsymbol{D}, \oplus, \otimes)$
- Example:
  - $\varphi(x,t) = \bigoplus_{y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(z,t)$
- Pairs (x,t) where there is a 3-path?
  - $Q_1(x,t) = \bigvee_{y,z} E(x,y) \wedge E(y,z) \wedge E(z,t)$ **T** = $(\mathbb{B}, \vee, \wedge)$

What is an FAQ?

- FAQ**<T>** is a *sum of a product* of functions under a semiring **T** = $(\boldsymbol{D}, \oplus, \otimes)$
- Example:
  - $\varphi(x, t) = \bigoplus_{y,z} \psi_1(x, y) \otimes \psi_2(y, z) \otimes \psi_3(z, t)$
- Pairs (x, t) where there is a 3-path?
  - $Q_1(x, t) = \bigvee_{y,z} E(x, y) \wedge E(y, z) \wedge E(z, t)$ **T** = $(\mathbb{B}, \vee, \wedge)$
- How many 3-paths?
  - $Q_2(x, t) = \sum_{y,z} 1_{E(x,y)} \times 1_{E(y,z)} \times 1_{E(z,t)}$ **T** = $(\mathbb{N}, +, \times)$

What is an FAQ?

- FAQ**<T>** is a *sum of a product* of functions under a semiring **T** = $(\boldsymbol{D}, \oplus, \otimes)$
- Example:
  - $\varphi(x,t) = \bigoplus_{y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(z,t)$
- Pairs (x,t) where there is a 3-path?
  - $Q_1(x,t) = \bigvee_{y,z} E(x,y) \wedge E(y,z) \wedge E(z,t)$ **T** = $(\mathbb{B}, \vee, \wedge)$
- How many 3-paths?
  - $Q_2(x,t) = \sum_{y,z} 1_{E(x,y)} \times 1_{E(y,z)} \times 1_{E(z,t)}$ **T** = $(\mathbb{N}, +, \times)$
- Shortest 3-path?
  - $Q_3(x,t) = \min_{y,z} W(x,y) + W(y,z) + W(z,t)$ **T** = $(\mathbb{R}, \min, +)$

What is an FAQ?

- FAQ**<T>** is a *sum of a product* of functions under a semiring **T** = $(\boldsymbol{D}, \oplus, \otimes)$
- Example:
  - $\varphi(x,t) = \bigoplus_{y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(z,t)$
- Pairs (x,t) where there is a 3-path?
  - $Q_1(x,t) = \bigvee_{y,z} E(x,y) \wedge E(y,z) \wedge E(z,t)$ **T** = $(\mathbb{B}, \vee, \wedge)$
- How many 3-paths?
  - $Q_2(x,t) = \sum_{y,z} 1_{E(x,y)} \times 1_{E(y,z)} \times 1_{E(z,t)}$ **T** = $(\mathbb{N}, +, \times)$
- Shortest 3-path?
  - $Q_3(x,t) = \min_{y,z} W(x,y) + W(y,z) + W(z,t)$ **T** = $(\mathbb{R}, \min, +)$
- Matrix Chain Multiplication
  - $M[x,t] = \sum_{y,z} M_1[x,y] \times M_2[y,z] \times M_3[z,t]$ **T** = $(\mathbb{R}, +, \times)$

InsideOut: Variable Elimination on Steroids

$$\varphi(x,t) = \bigoplus_{y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(z,t)$$

[Dechter'99, Aji&McEliese'00, Kohlas&Wilson'08, ...]

InsideOut: Variable Elimination on Steroids

$$\varphi(x,t) = \bigoplus_{y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(z,t)$$

- Eliminate z
  - $\varphi(x,t) = \bigoplus_y \psi_1(x,y) \otimes \underbrace{\bigoplus_z \psi_2(y,z) \otimes \psi_3(z,t)}_{\psi_z(y,t)}$
  - $\varphi(x,t) = \bigoplus_y \psi_1(x,y) \otimes \psi_z(y,t)$

[Dechter'99, Aji&McEliese'00, Kohlas&Wilson'08, ...]

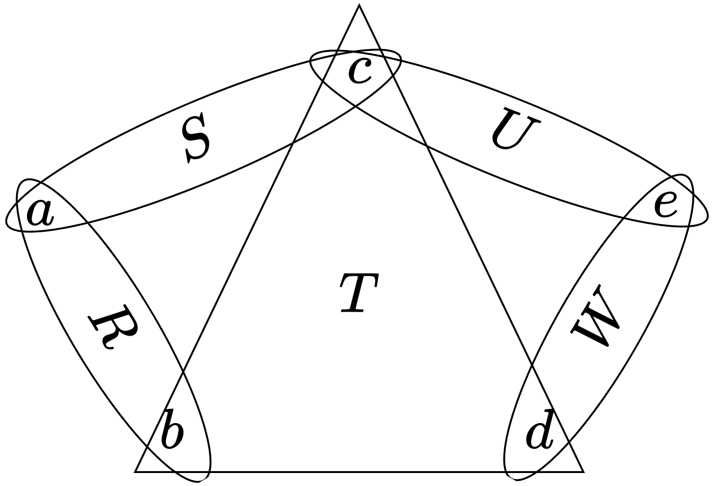
InsideOut: Variable Elimination on Steroids

$$\varphi(x,t) = \bigoplus_{y,z} \psi_1(x,y) \otimes \psi_2(y,z) \otimes \psi_3(z,t)$$

- Eliminate z
  - $\varphi(x,t) = \bigoplus_y \psi_1(x,y) \otimes \underbrace{\bigoplus_{z} \psi_2(y,z) \otimes \psi_3(z,t)}_{\psi_z(y,t)}$
  - $\varphi(x,t) = \bigoplus_y \psi_1(x,y) \otimes \psi_z(y,t)$
- Eliminate y
  - $\varphi(x,t) = \underbrace{\bigoplus_{y} \psi_1(x,y) \otimes \psi_z(y,t)}_{\psi_y(x,t)}$
  - $\varphi(x,t) = \psi_y(x,t)$

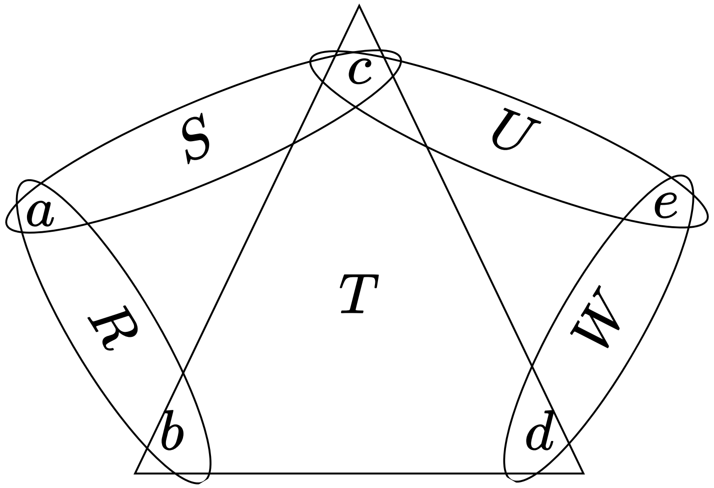
[Dechter'99, Aji&McEliese'00, Kohlas&Wilson'08, ...]

InsideOut: Variable Elimination on Steroids



$$\varphi(a) = \bigoplus_{b,c,d,e} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes U(c,e) \otimes W(d,e)$$

InsideOut: Variable Elimination on Steroids

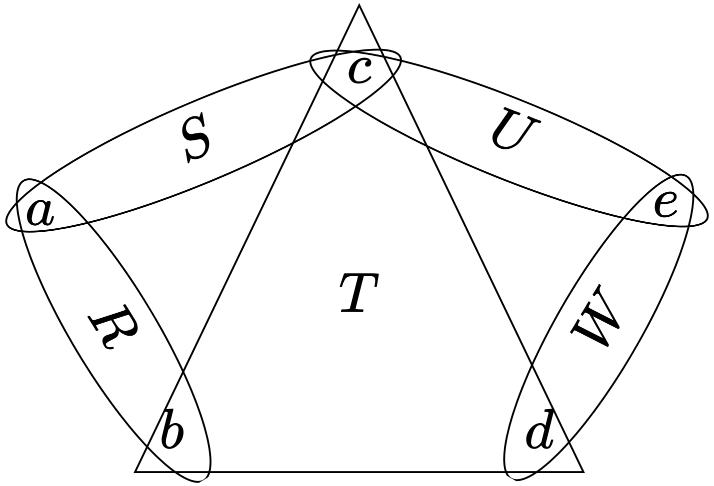


$$\varphi(a) = \bigoplus_{b,c,d,e} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes U(c,e) \otimes W(d,e)$$

- Eliminate d

$$= \bigoplus_{b,c,e} R(a,b) \otimes S(a,c) \otimes U(c,e) \otimes \underbrace{\bigoplus_{d} T(b,c,d) \otimes W(d,e)}_{\substack{\psi_d(b,c,e) \\ O(N^2)}}$$

InsideOut: Variable Elimination on Steroids

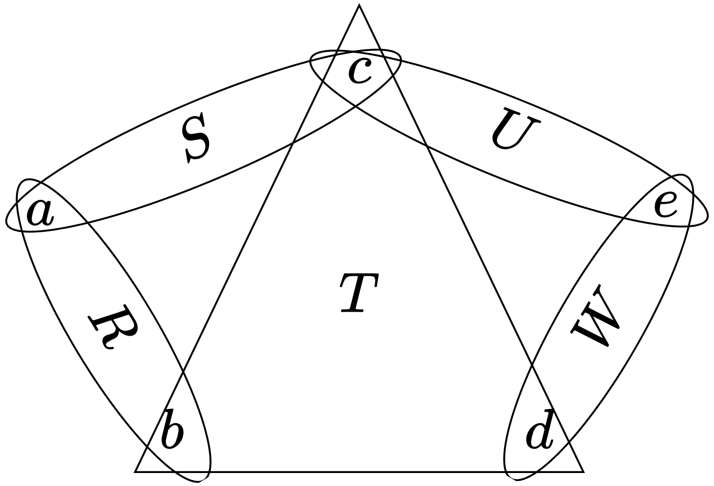


$$\varphi(a) = \bigoplus_{b,c,d,e} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes U(c,e) \otimes W(d,e)$$

- Eliminate e

$$= \bigoplus_{b,c,d} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes \underbrace{\bigoplus_{e} U(c,e) \otimes W(d,e)}_{\substack{\psi_e(c,d) \\ O(N^2)}}$$

InsideOut: Variable Elimination on Steroids



$$\varphi(a) = \bigoplus_{b,c,d,e} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes U(c,e) \otimes W(d,e)$$

- Eliminate e smarter

$$= \bigoplus_{b,c,d} R(a,b) \otimes S(a,c) \otimes T(b,c,d) \otimes \underbrace{\bigoplus_{e} U(c,e) \otimes W(d,e) \otimes T'(c,d)}_{\substack{\psi'_e(c,d) \\ O(N^{1.5})}}$$

$$T'(c,d) := 1_{\exists b \,:\, T(b,c,d) \neq \mathbf{0}}$$

Practical Impact

- Implemented and **used in production** at
  - LogicBlox
  - RelationalAI
- An *essential* part the query optimizer
  - Many real workloads **don't terminate** without it!
  - Breaks down a large query into smaller ones

FAQs are not enough: Enter Recursion

Many real-world problems are **circular!**

- Price $\Rightarrow$ Sales $\Rightarrow$ Price...
- ML training: Model1 $\Rightarrow$ Error1 $\Rightarrow$ Model2 $\Rightarrow$ Error2 ...
- LLMs (auto-regression): Token1 $\Rightarrow$ Token2 $\Rightarrow$ Token 3 ...
- Graphs (Knowledge Graphs, Social Networks, Semantic Web...)
  - Reachability
  - Shortest Path
  - Pagerank
  - Betweenness Centrality

$\mathrm{Datalog}^{o}$: FAQs with Recursion

[M. Abo Khamis, Hung Ngo, Reinhard Pichler, Dan Suciu, Remy Wang - **PODS'22 Best Paper - JACM'24**]

- Abstract $\mathrm{Datalog}^{o}$ program parameterized by a semiring
  - $R(X,Y) \quad = \quad E(X,Y) \quad \oplus \quad \bigoplus_Z E(X,Z) \otimes R(Z,Y)$ $\qquad (\boldsymbol{D}, \oplus, \otimes)$
- Transitive Closure
  - $R(X,Y) \quad = \quad E(X,Y) \quad \vee \quad E(X,Z) \wedge R(Z,Y)$ $\qquad (\mathbb{B}, \vee, \wedge)$
- Shortest Paths
  - $R(X,Y) \quad = \quad \min(E(X,Y), \quad \min_Z(E(X,Z) + R(Z,Y)))$ $\qquad (\mathbb{R}, \min, +)$
- Number of Paths
  - $R(X,Y) \quad = \quad E(X,Y) \quad + \quad \sum_Z E(X,Z) \times R(Z,Y)$ $\qquad (\mathbb{N}, +, \times)$

Recognition of FAQ and $\mathrm{Datalog}^{o}$

- **Test-of-Time Award** at ACM PODS 2026 (2nd year in a row)
- **Two Best Paper Awards** at ACM PODS 2016 and 2022
- Invitation for a *Plenary Talk* at **STOC 2017** (and again in STOC 2018)
- Two invitations to the **Journal of the ACM** (only one published)
- Two ACM SIGMOD Research Highlight Awards
- Best Dissertation Award at UB 2016

Key Takeaways So Far...

- FAQs are "templated" queries
  - InsideOut is a "templated"-algorithm
    - unifies and generalizes several algorithms in different domains
- $\mathrm{Datalog}^o$ are FAQs with Recursion
  - Very expressive
  - But very hard to optimize

Future Directions

Recursion Optimization!

- Recursion is inevitable when modeling problems in the real world
- Yet, we are struggling to automatically generate algorithms for these problems
- because we are struggling to automatically analyze the runtimes of these algorithms

Change is good...   if you can keep up!

- Real data is constantly changing, now faster than ever!
- Real-world query evaluation is like shooting a moving target
- Algorithms Community:
  - **Dynamic** Algorithms
- DB Community:
  - **Incremental View Maintenance (IVM)**

Change is good... if you can keep up!

- State-of-the-art is scattered!
  - Triangle $\Rightarrow O(N^{1/2})$-updates [KNNOZ'20]
  - 4-path $\Rightarrow O(N^{1/2})$-updates [HHH'22]
  - 4-cycle
    - Combinatorial: $O(N^{2/3})$-updates [HHH'22]
    - Non-combinatorial: $O(N^{2/3-\epsilon})$ [AS'25]
    - ...
- Is there a **dynamic** Meta-algorithm?

Conclusion

Conclusion

- A Query Optimizer is a <u>Meta-Algorithm</u>: Any problem $\Rightarrow$ Algorithm

- Perfect opportunities for cross-community collaboration
  - **Algorithms/Theory**
  - **DB Systems**
  - **DB Theory**

Conclusion

- A Query Optimizer is a Meta-Algorithm: Any problem $\Rightarrow$ Algorithm
- Perfect opportunities for cross-community collaboration
  - **Algorithms/Theory**
    - Brings the solution techniques, lacks the generality
  - **DB Systems**
  - **DB Theory**

Conclusion

- A Query Optimizer is a Meta-Algorithm: Any problem $\Rightarrow$ Algorithm

- Perfect opportunities for cross-community collaboration
  - **Algorithms/Theory**
    - Brings the solution techniques, lacks the generality
  - **DB Systems**
    - Brings the problems, engineering, lacks rigorous solutions
  - **DB Theory**

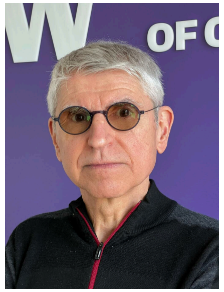
Conclusion

- A Query Optimizer is a <u>Meta-Algorithm</u>: Any problem $\Rightarrow$ Algorithm

- Perfect opportunities for cross-community collaboration
  - **Algorithms/Theory**
    - Brings the solution techniques, lacks the generality
  - **DB Systems**
    - Brings the problems, engineering, lacks rigorous solutions
  - **DB Theory**
    - Brings the bridge, but needs to keep up!

Thank You All!

Hung Ngo



Dan Suciu

Xiao Hu

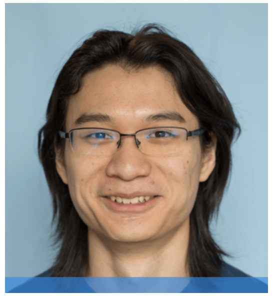
Dan Olteanu

Chris Mayor

Haozhe Zhang

Atri Rudra

Reinhard Pichler



Remy Wang

Molham Aref

and many more!

Appendix

Appendix A: PANDA

Appendix B: FAQ

FAQ<T>: The underlying structure T

FAQ<T>: The underlying structure T

- **T** is a commutative semiring (or multiple semirings)
  - **T** = $(\boldsymbol{D}, \oplus, \otimes)$
    - Distributive Property:
      - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

FAQ<T>: The underlying structure T

- **T** is a commutative semiring (or multiple semirings)
  - **T** = $(\boldsymbol{D}, \oplus, \otimes)$
    - Distributive Property:
      - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
- Examples
  - Boolean: $(\mathbb{B}, \vee, \wedge)$
  - Tropical: $(\mathbb{R} \cup \{\infty\}, \min, +)$
  - Numeric: $(\mathbb{R}, +, \times)$

Appendix C:

$\mathrm{Datalog}^{o}$

$\mathrm{Datalog}^o$: Datalog with Semirings

[M. Abo Khamis, Hung Ngo, Reinhard Pichler, Dan Suciu, Remy Wang - **PODS'22 Best Paper - JACM'24**]

- Abstract $\mathrm{Datalog}^o$ program parameterized by a semiring
  - $R(X,Y) \quad = \quad E(X,Y) \quad \oplus \quad \bigoplus_Z E(X,Z) \otimes R(Z,Y)$ $\qquad (\boldsymbol{D}, \oplus, \otimes)$
- Transitive Closure
  - $R(X,Y) \quad = \quad E(X,Y) \quad \vee \quad E(X,Z) \wedge R(Z,Y)$ $\qquad (\mathbb{B}, \vee, \wedge)$
- Shortest Paths
  - $R(X,Y) \quad = \quad \min(E(X,Y), \quad \min_Z(E(X,Z) + R(Z,Y)))$ $\qquad (\mathbb{R}, \min, +)$
- Number of Paths
  - $R(X,Y) \quad = \quad E(X,Y) \quad + \quad \sum_Z E(X,Z) \times R(Z,Y)$ $\qquad (\mathbb{N}, +, \times)$

Datalog^{o}: Datalog with Semirings

[M. Abo Khamis, Hung Ngo, Reinhard Pichler, Dan Suciu, Remy Wang - **PODS'22 Best Paper - JACM'24**]

- Abstract Datalog^{o} program parameterized by a semiring
  - $R(X,Y) \quad = \quad E(X,Y) \quad \oplus \quad \bigoplus_Z E(X,Z) \otimes R(Z,Y)$ $\qquad (\boldsymbol{D}, \oplus, \otimes)$
- Transitive Closure
  - $R(X,Y) \quad = \quad E(X,Y) \quad \vee \quad E(X,Z) \wedge R(Z,Y)$ $\qquad (\mathbb{B}, \vee, \wedge)$

Terminates *always*!

- Shortest Paths
  - $R(X,Y) \quad = \quad \min(E(X,Y), \quad \min_Z(E(X,Z) + R(Z,Y)))$ $\qquad (\mathbb{R}, \min, +)$
- Number of Paths
  - $R(X,Y) \quad = \quad E(X,Y) \quad + \quad \sum_Z E(X,Z) \times R(Z,Y)$ $\qquad (\mathbb{N}, +, \times)$

Datalog^{o}: Datalog with Semirings

[M. Abo Khamis, Hung Ngo, Reinhard Pichler, Dan Suciu, Remy Wang - **PODS'22 Best Paper - JACM'24**]

- Abstract Datalog^{o} program parameterized by a semiring
  - $R(X,Y) \quad = \quad E(X,Y) \quad \oplus \quad \bigoplus_Z E(X,Z) \otimes R(Z,Y)$ $\qquad (\boldsymbol{D}, \oplus, \otimes)$
- Transitive Closure
  - $R(X,Y) \quad = \quad E(X,Y) \quad \vee \quad E(X,Z) \wedge R(Z,Y)$ $\qquad$ Terminates *always*! $(\mathbb{B}, \vee, \wedge)$
- Shortest Paths
  - $R(X,Y) \quad = \quad \min(E(X,Y), \quad \min_Z(E(X,Z) + R(Z,Y)))$ $\qquad$ Terminates *only* on $\mathbb{R}_+$ $(\mathbb{R}, \min, +)$
- Number of Paths
  - $R(X,Y) \quad = \quad E(X,Y) \quad + \quad \sum_Z E(X,Z) \times R(Z,Y)$ $\qquad (\mathbb{N}, +, \times)$

Datalog^{o}: Datalog with Semirings

[M. Abo Khamis, Hung Ngo, Reinhard Pichler, Dan Suciu, Remy Wang - **PODS'22 Best Paper - JACM'24**]

- Abstract Datalog^{o} program parameterized by a semiring
  - $R(X,Y) \quad = \quad E(X,Y) \quad \oplus \quad \bigoplus_Z E(X,Z) \otimes R(Z,Y)$ $\qquad (\boldsymbol{D}, \oplus, \otimes)$
- Transitive Closure
  - $R(X,Y) \quad = \quad E(X,Y) \quad \vee \quad E(X,Z) \wedge R(Z,Y)$ $\qquad$ Terminates *always*! $(\mathbb{B}, \vee, \wedge)$
- Shortest Paths
  - $R(X,Y) \quad = \quad \min(E(X,Y), \quad \min_Z(E(X,Z) + R(Z,Y)))$ $\qquad$ Terminates *only* on $\mathbb{R}_+$ $(\mathbb{R}, \min, +)$
- Number of Paths
  - $R(X,Y) \quad = \quad E(X,Y) \quad + \quad \sum_Z E(X,Z) \times R(Z,Y)$ $\qquad$ *No* termination! (cycles) $(\mathbb{N}, +, \times)$

Not all semirings are created equal!

- **Q1:** How do we explain variations in termination/speed?
- **Q2:** How do we "fix" non-terminating programs?

Not all semirings are created equal!

- **Q1:** How do we explain variations in termination/speed?
- **Q2:** How do we "fix" non-terminating programs?
- **Hint:** $\boldsymbol{p}$**-stability** of the semiring $(\boldsymbol{D}, \oplus, \otimes)$
  - $(\boldsymbol{D}, \oplus, \otimes)$ is $\boldsymbol{p}$**-stable**:
    - $\mathbf{1} \oplus a^1 \oplus a^2 \oplus \cdots \oplus a^p \quad = \quad \mathbf{1} \oplus a^1 \oplus a^2 \oplus \cdots \oplus a^{p+1}$

Not all semirings are created equal!

- **Q1:** How do we explain variations in termination/speed?
- **Q2:** How do we "fix" non-terminating programs?
- **Hint:** $\boldsymbol{p}$**-stability** of the semiring $(\boldsymbol{D}, \oplus, \otimes)$
  - $(\boldsymbol{D}, \oplus, \otimes)$ is $\boldsymbol{p}$**-stable**:
    - $\mathbf{1} \oplus a^1 \oplus a^2 \oplus \cdots \oplus a^p \quad = \quad \mathbf{1} \oplus a^1 \oplus a^2 \oplus \cdots \oplus a^{p+1}$
  - Examples
    - $(\mathbb{B}, \vee, \wedge)$ is 0-stable: $\quad \textsf{true} = \textsf{true} \vee a$
    - $(\mathbb{R}_+, \min, +)$ is 0-stable: $\quad 0 = \min(0, a) \qquad \forall a \in \mathbb{R}_+$
    - $(\mathbb{R}, \min, +)$ is NOT p-stable $\quad 0 \neq \min(0, a) \qquad \forall a \in \mathbb{R}$
    - $(\mathbb{N}, +, \times)$ is NOT p-stable $\quad 1 \neq 1 + a \qquad \forall a \in \mathbb{N}$
      - *"Lifted"* $(\mathbb{N} \cup \{\bot\}, +, \times)$ is 0-stable $\quad \bot + a = \bot, \qquad \bot \times a = \bot$

Termination from the Lens of p-stability

- Transitive Closure

  Terminates *always*!

  - $R(X,Y) \quad = \quad E(X,Y) \quad \vee \quad E(X,Z) \wedge R(Z,Y)$

- Shortest Paths

  Terminates *only* on $\mathbb{R}_+$

  - $R(X,Y) \quad = \quad \min(E(X,Y), \quad \min_Z(E(X,Z) + R(Z,Y)))$

- Number of Paths

  No termination! (cycles)

  - $R(X,Y) \quad = \quad E(X,Y) \quad + \quad \sum_Z E(X,Z) \times R(Z,Y)$

Termination from the Lens of p-stability

- Transitive Closure
  - $R(X,Y) \quad = \quad E(X,Y) \quad \vee \quad E(X,Z) \wedge R(Z,Y)$

Terminates *always*!

$(\mathbb{B}, \vee, \wedge)$ is 0-stable

- Shortest Paths
  - $R(X,Y) \quad = \quad \min(E(X,Y), \quad \min_Z(E(X,Z) + R(Z,Y)))$

Terminates *only* on $\mathbb{R}_+$

- Number of Paths
  - $R(X,Y) \quad = \quad E(X,Y) \quad + \quad \sum_Z E(X,Z) \times R(Z,Y)$

No termination! (cycles)

Termination from the Lens of p-stability

- Transitive Closure
  - $R(X,Y) \quad = \quad E(X,Y) \quad \vee \quad E(X,Z) \wedge R(Z,Y)$

Terminates *always*!

$(\mathbb{B}, \vee, \wedge)$ is 0-stable

- Shortest Paths
  - $R(X,Y) \quad = \quad \min(E(X,Y), \quad \min_Z(E(X,Z) + R(Z,Y)))$

Terminates *only* on $\mathbb{R}_+$

$(\mathbb{R}_+, \min, +)$ is 0-stable

$(\mathbb{R}, \min, +)$ is NOT p-stable

- Number of Paths
  - $R(X,Y) \quad = \quad E(X,Y) \quad + \quad \sum_Z E(X,Z) \times R(Z,Y)$

No termination! (cycles)

Termination from the Lens of p-stability

- Transitive Closure
  - $R(X,Y) \quad = \quad E(X,Y) \quad \vee \quad E(X,Z) \wedge R(Z,Y)$

Terminates *always*!

$(\mathbb{B}, \vee, \wedge)$ is 0-stable

- Shortest Paths
  - $R(X,Y) \quad = \quad \min(E(X,Y), \quad \min_Z(E(X,Z) + R(Z,Y)))$

Terminates *only* on $\mathbb{R}_+$

$(\mathbb{R}_+, \min, +)$ is 0-stable

$(\mathbb{R}, \min, +)$ is NOT p-stable

- Number of Paths
  - $R(X,Y) \quad = \quad E(X,Y) \quad + \quad \sum_Z E(X,Z) \times R(Z,Y)$

No termination! (cycles)

$(\mathbb{N}, +, \times)$ is NOT p-stable

$(\mathbb{N} \cup \{\perp\}, +, \times)$ is 0-stable

(cycles produce $\perp$)

Appendix D:
Future Directions

Output-Sensitive Meta-Algorithm?

- Runtime in terms of N *and* output size OUT

Output-Sensitive Meta-Algorithm?

- Runtime in terms of N *and* output size OUT

- State-of-the-art is *very* scattered!
  - 2-path: $O(N \cdot \mathrm{OUT}^{0.3454} + \mathrm{OUT} + N^{0.8} \cdot \mathrm{OUT}^{0.5436})$ [ABFK'24]
  - Triangle: $O(N^{1.408} + N^{1.222} \cdot \mathrm{OUT}^{0.186})$ [BPWZ'14]
  - Acyclic: $O(N^{0.372} \cdot \mathrm{OUT} + N \cdot \mathrm{OUT}^{0.871})$ [H'24]
  - ...

$\mathrm{Datalog}^o$ Analogues of $\mathrm{Datalog}$ Problems

- **Data Repair**
  - $\mathrm{Datalog}$:
    - Add a minimum number of edges to a graph to ensure **reachability**
  - $\mathrm{Datalog}^o$:
    - Change the weights of edges of a graph to meet certain constraints on **shortest paths**

- **Analyze #Iterations**