DATALAB
@Northeastern

Any-k: Ranked Enumeration for Dynamic Programming

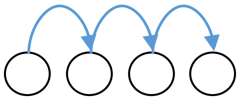
Nikolaos Tziavelis

Based on joint work with: Wolfgang Gatterbauer, Mirek Riedewald

Northeastern University, Boston

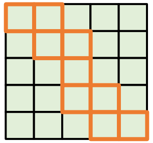
Website: https://northeastern-datalab.github.io/anyk/

Ranked Enumeration for Combinatorial Problems



Enumeration

Optimization



Dynamic Programming
(& semirings)

In order
of importance

2nd best, 3rd best, …?

Ranked Enumeration

"Any-k"

Anytime algorithms + Top-k

Outline

- Ranked Enumeration & Dynamic Programming
  - DP as a DAG
  - Semirings
  - Any-k Algorithms
- Ranked Enumeration for (Full) Conjunctive Queries
  - Mapping CQs to DP
  - Ranking Function & Query Structure
- Conclusion

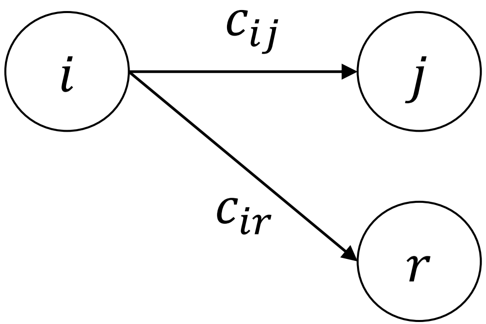
Outline

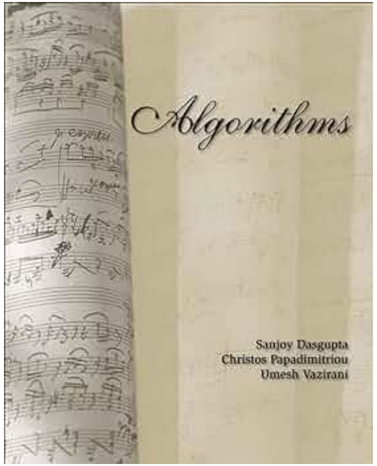
- Ranked Enumeration & Dynamic Programming
  - DP as a DAG
  - Semirings
  - Any-k Algorithms
- Ranked Enumeration for (Full) Conjunctive Queries
  - Mapping CQs to DP
  - Ranking Function & Query Structure
- Conclusion

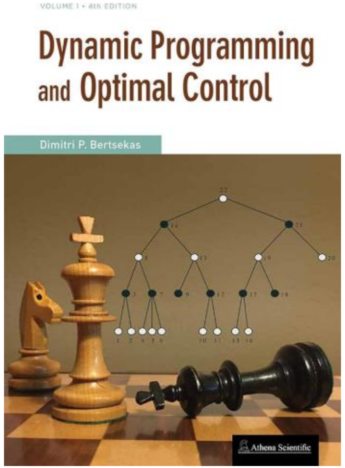
Dynamic Programming

View of DP as a Directed Acyclic Graph (DAG)

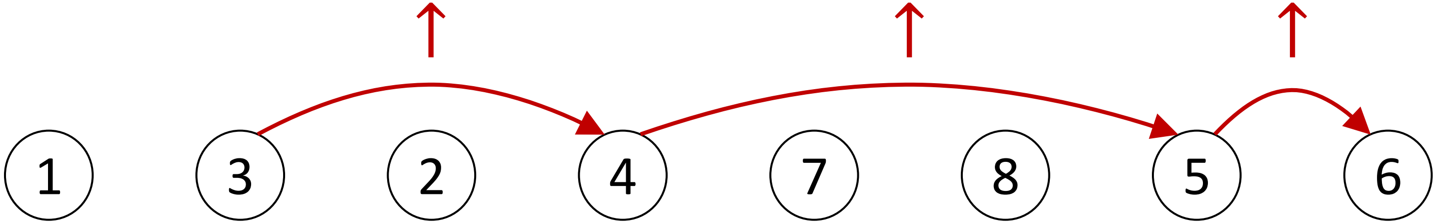
$$f(i) = \min\{c_{ij} + f(j), c_{ir} + f(r)\}$$



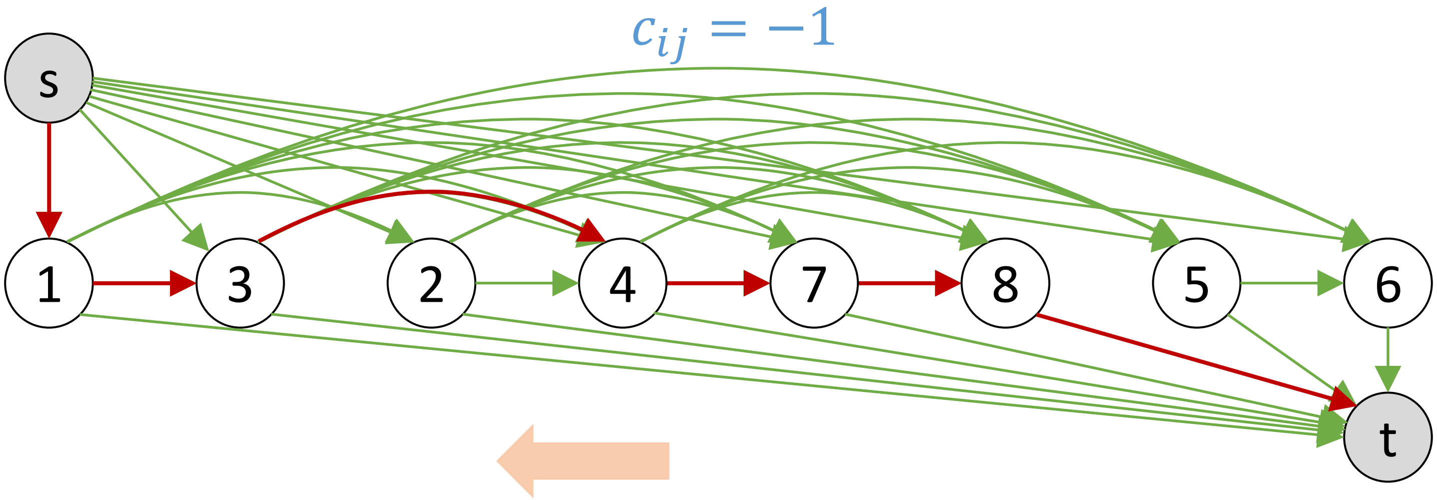




Example: Longest Increasing Subsequence



Longest Increasing Subsequence with DP



Dynamic Programming

```
for $i = n, \ldots, 2, 1$:
  $f(i) = \max\{f(j) + 1 \mid j > i, v(j) > v(i)\}$
return $\max\{f(i) \mid i \in [n]\}$
```

Shortest path in DAG

```
for $i = n, \ldots, 2, 1, s$:
  $d(i) = \min\{d(j) + c_{ij} \mid (i,j) \in E\}$
return $-d(s)$
```

Outline

- Ranked Enumeration & Dynamic Programming
  - DP as a DAG
  - Semirings
  - Any-k Algorithms
- Ranked Enumeration for (Full) Conjunctive Queries
  - Mapping CQs to DP
  - Ranking Function & Query Structure
- Conclusion

DP & Semirings

Previous example:

$$f(i) = \max\{f(j) + 1 | j \in \cdots\}$$

$$f(i) = \bigoplus_j \{f(j) \otimes c_{ij}\}$$

Commutative Semiring $(W, \oplus, \otimes, 0, 1)$

1. $(W, \oplus, 0)$ is a commutative monoid
2. $(W, \otimes, 1)$ is a commutative monoid
3. $\otimes$ distributes over $\oplus$: $(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$
4. 0 annihilates $\otimes$: $0 \otimes x = 0$

Mohri. Semiring frameworks and algorithms for shortest-distance problems. JALC'02 http://www.jalc.de/issues/2002/issue_7_3/abs-321.pdf

Total Order & Selectivity

- Additional property for total order: Selectivity
  - $\forall xy\colon\ (x \oplus y = x) \vee (x \oplus y = y)$
  - Semiring with Selectivity = Selective Dioid
- "Natural" total order: $x \leq y$ iff $x \oplus y = x$
- Examples:
  - Tropical semiring $(\mathbb{R}^{\infty}, \min, +, \infty, 0)$ ✓
  - Viterbi semiring $([0,1], \max, \times, 0,1)$ ✓
  - Boolean semiring $(\{0,1\}, \vee, \wedge, 0,1)$ ✓
  - Natural numbers semiring $(\mathbb{N}, +, \times, 0,1)$ ✗
    - Can count #paths / #solutions
    - What would the 2nd best solution be here?

Distributivity → Monotonicity

Monotonicity in Selective Dioids: $\mathrm{x} \leq \mathrm{y} \Rightarrow x \otimes z \leq y \otimes z$

Proof

$\mathrm{x} \leq \mathrm{y}$

$x \oplus y = x$

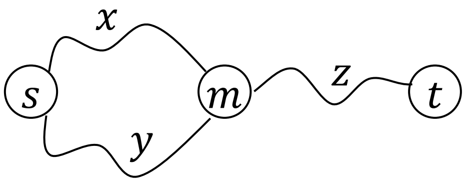
$(x \oplus y) \otimes z = x \otimes z$

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

Distributivity

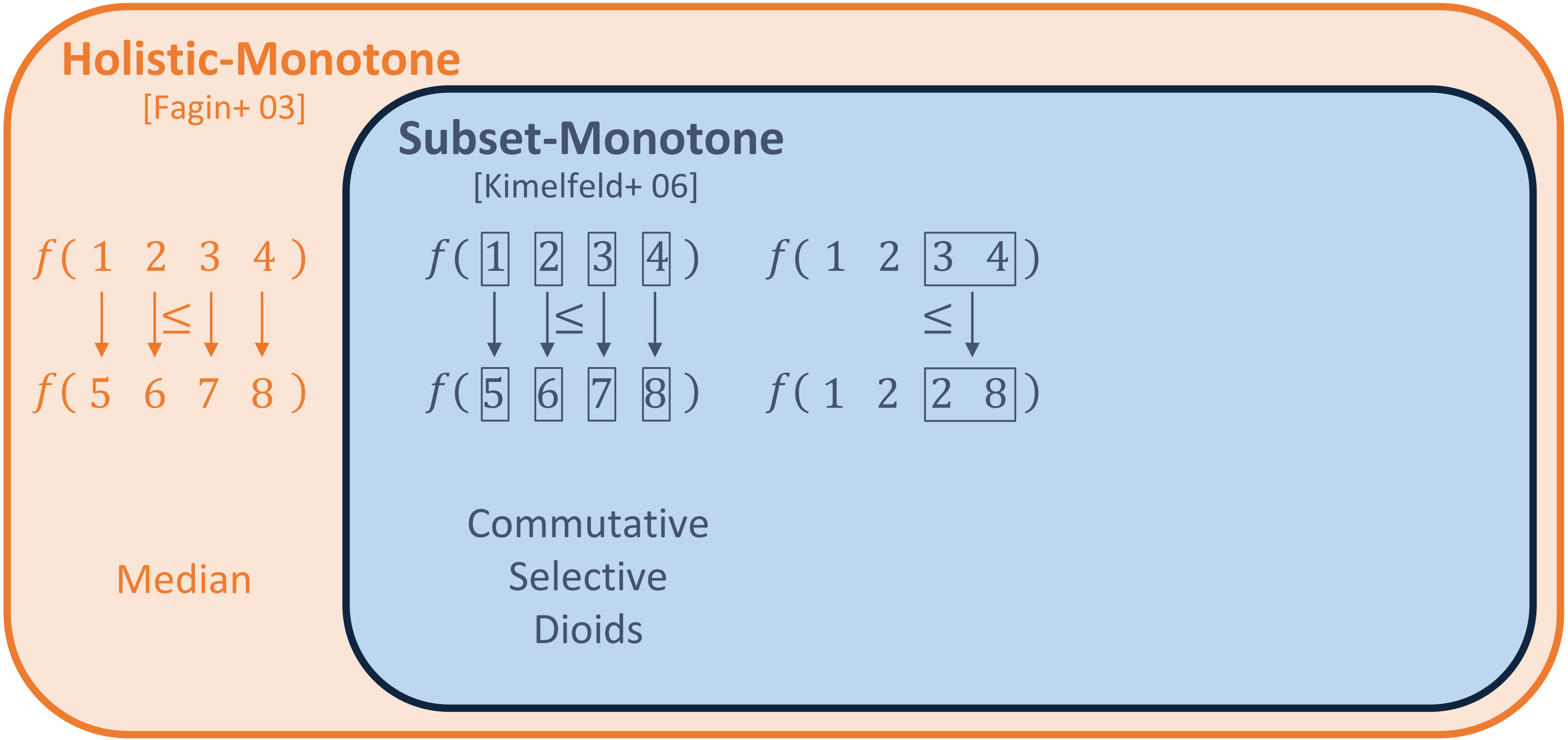
$(x \otimes z) \leq y \otimes z$

Equivalent to "optimal substructure“ property in DP



$x \leq y \quad \otimes \qquad z$

Monotonicity Classes for Ranking Functions



Holistic-Monotone

[Fagin+ 03]

$$\begin{tikzcd} f(1 \arrow[d] & 2 \arrow[d, "\leq"] & 3 \arrow[d] & 4) \arrow[d] \\ f(5 & 6 & 7 & 8) \end{tikzcd}$$

Median

Subset-Monotone

[Kimelfeld+ 06]

$$\begin{tikzcd} f(\boxed{1} \arrow[d] & \boxed{2} \arrow[d, "\leq"] & \boxed{3} \arrow[d] & \boxed{4}) \arrow[d] \\ f(\boxed{5} & \boxed{6} & \boxed{7} & \boxed{8}) \end{tikzcd}$$

$$\begin{tikzcd} f(1 \; 2 \; \boxed{3 \;\; 4}) \arrow[d, "\leq"'] \\ f(1 \; 2 \; \boxed{2 \;\; 8}) \end{tikzcd}$$

Commutative
Selective
Dioids

[Fagin+03] Fagin, Lotem, Naor. Optimal aggregation algorithms for middleware. JCSS 2003.https://doi.org/10.1145/375551.375567

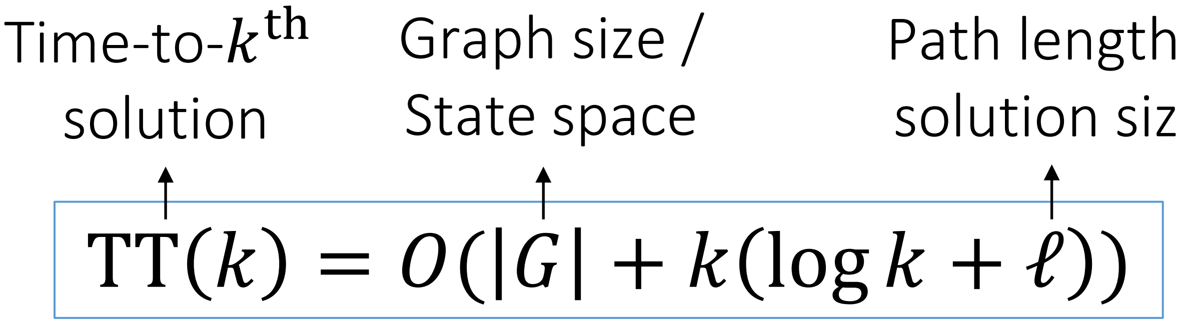
[Kimelfeld+06] Kimelfeld, Sagiv. Incrementally Computing Ordered Answers of Acyclic Conjunctive Queries. NGITS’06 https://doi.org/10.1007/11780991_13

Outline

- Ranked Enumeration & Dynamic Programming
  - DP as a DAG
  - Semirings
  - Any-k Algorithms
- Ranked Enumeration for (Full) Conjunctive Queries
  - Mapping CQs to DP
  - Ranking Function & Query Structure
- Conclusion

Any-k Algorithms

- Best answer (DP) → shortest path in DAG
- k^{th} best answer → k^{th} shortest path in DAG
- Best we know for subset-monotone ranking functions:



Time-to-k^{th} solution

Graph size / State space

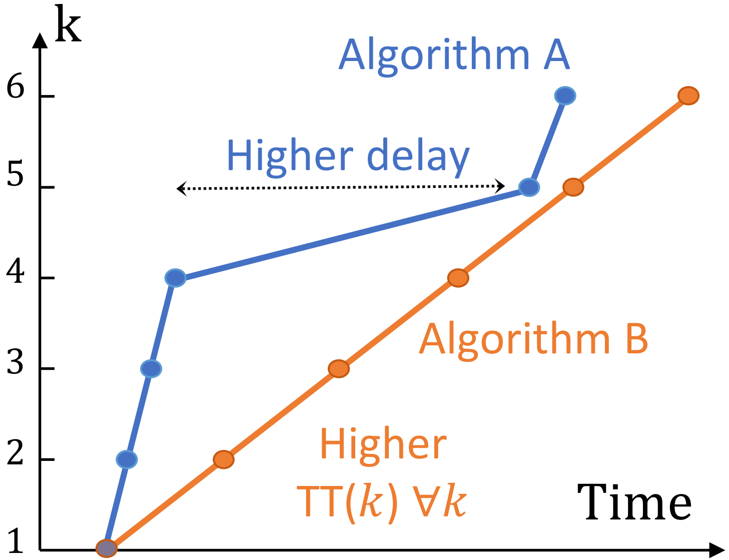
Path length / solution size

$$\boxed{\text{TT}(k) = O(|G| + k(\log k + \ell))}$$

Measures of Enumeration: $\mathrm{TT}(k)$ vs Delay

- Fastest answers: $\mathrm{TT}(k)$
- What about delay?

- We can upper bound $\mathrm{TT}(k)$ with delay:
  - $\text{delay} \leq c \Rightarrow \mathrm{TT}(k) \leq |\text{Prep}| + ck$
- Improving the delay of an algorithm with "good $\mathrm{TT}(k)$ "can slow it down
- Lower bounds are more general if stated for $\mathrm{TT}(k)$



$\mathrm{TT}(k)$ vs Delay Gap

Is the gap “real”?

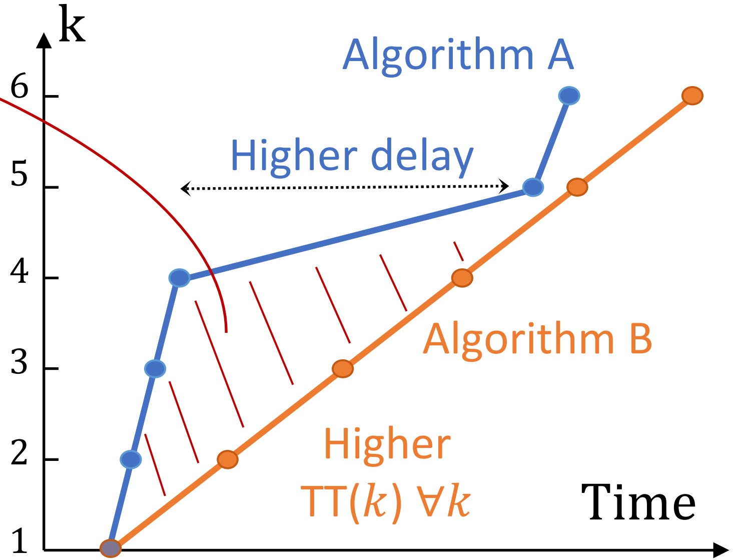
Answer #1 [CS23]

The delay term in $\mathrm{TT}(k)$

Incremental delay => delay at the cost of a log factor <u>for unranked enumeration</u>

Answer #2

For ranked enumeration, we can get a better algorithm for $\mathrm{TT}(k)$



[CS23] Capelli, Strozecki. Geometric Amortization of Enumeration Algorithms. STACS 2023 https://doi.org/10.4230/LIPIcs.STACS.2023.18

The Anyk-Part+ Algorithm

#nodes

path length

$$O(|G| + k(\log k + \ell)) \longrightarrow O(|G| + k(\log N + \ell))$$

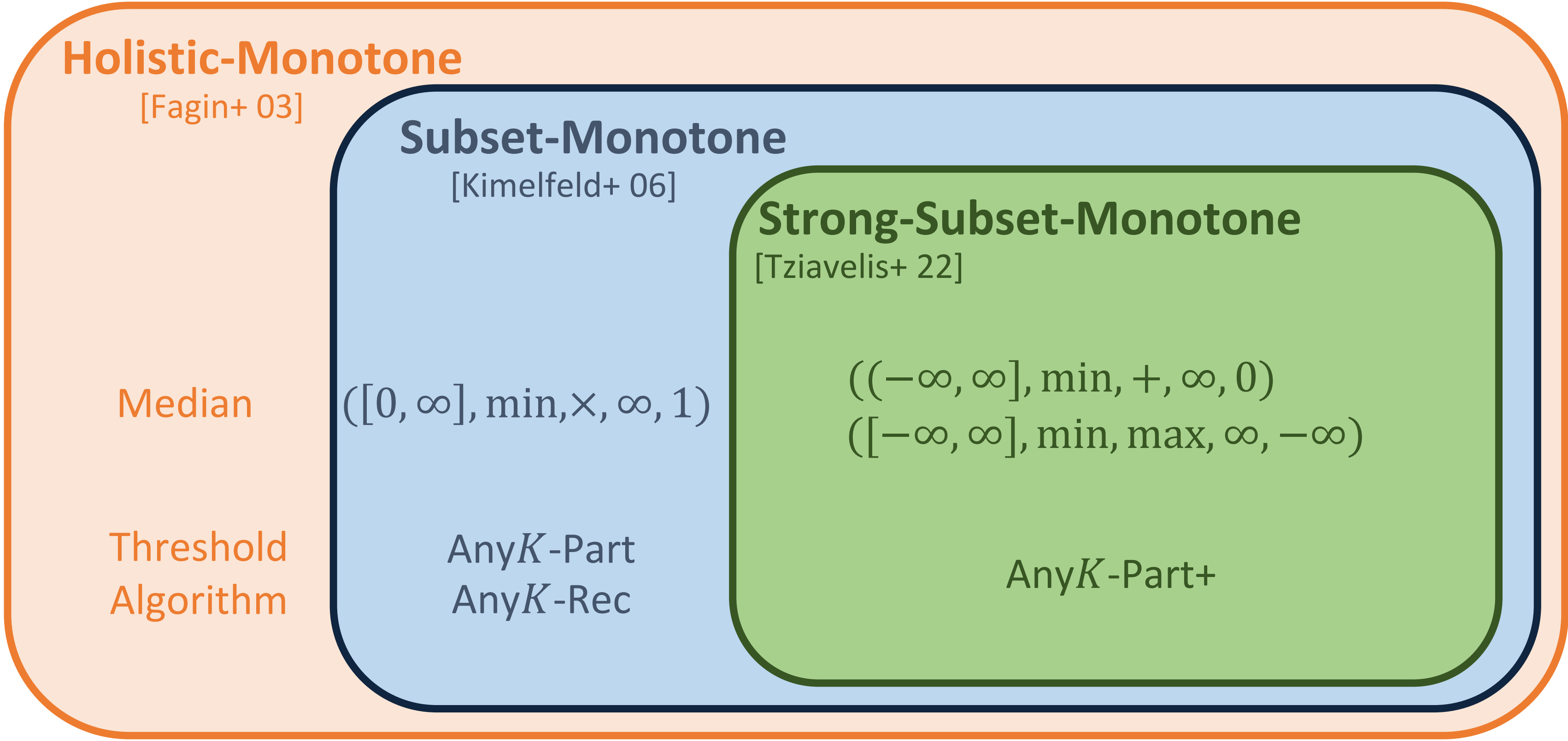
Previously best known

Small k: $O(|G|)$ dominates, same

Large k: Better

Faster than sorting for entire sorted output

Monotonicity Classes



	Holistic-Monotone [Fagin+ 03]	Subset-Monotone [Kimelfeld+ 06]	Strong-Subset-Monotone [Tziavelis+ 22]
Examples	Median	$([0,\infty],\min,\times,\infty,1)$	$((-\infty,\infty],\min,+,\infty,0)$ $([-\infty,\infty],\min,\max,\infty,-\infty)$
Algorithms	Threshold Algorithm	Any$K$-Part Any$K$-Rec	Any$K$-Part+

[Fagin+03] Fagin, Lotem, Naor. Optimal aggregation algorithms for middleware. JCSS 2003.https://doi.org/10.1145/375551.375567

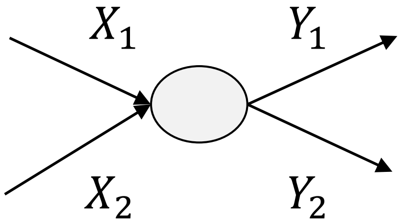
[Kimelfeld+06] Kimelfeld, Sagiv. Incrementally Computing Ordered Answers of Acyclic Conjunctive Queries. NGITS'06 https://doi.org/10.1007/11780991_13

[Tziavelis+22] Tziavelis, Gatterbauer, Riedewald. Any-k Algorithms for Enumerating Ranked Answers to Conjunctive Queries. arXiv'22 https://arxiv.org/abs/2205.05649

Strong-Subset-Monotonicity

Strong-Subset-Monotonicity

$$f(X_1, Y_1) \leq f(X_1, Y_2) \wedge f(X_1) \leq f(X_2) \Rightarrow f(X_2, Y_1) \leq f(X_2, Y_2)$$



$((-\infty, \infty], \min, +, \infty, 0)$ ✓

1) $(1, 1, 1)$
2) $(1, 1, 2)$
3) $(1, 4, 1)$

$f(2{,}1{,}1) < f(2{,}1{,}2)$
$f(2{,}1{,}2) < f(2{,}4{,}1)$

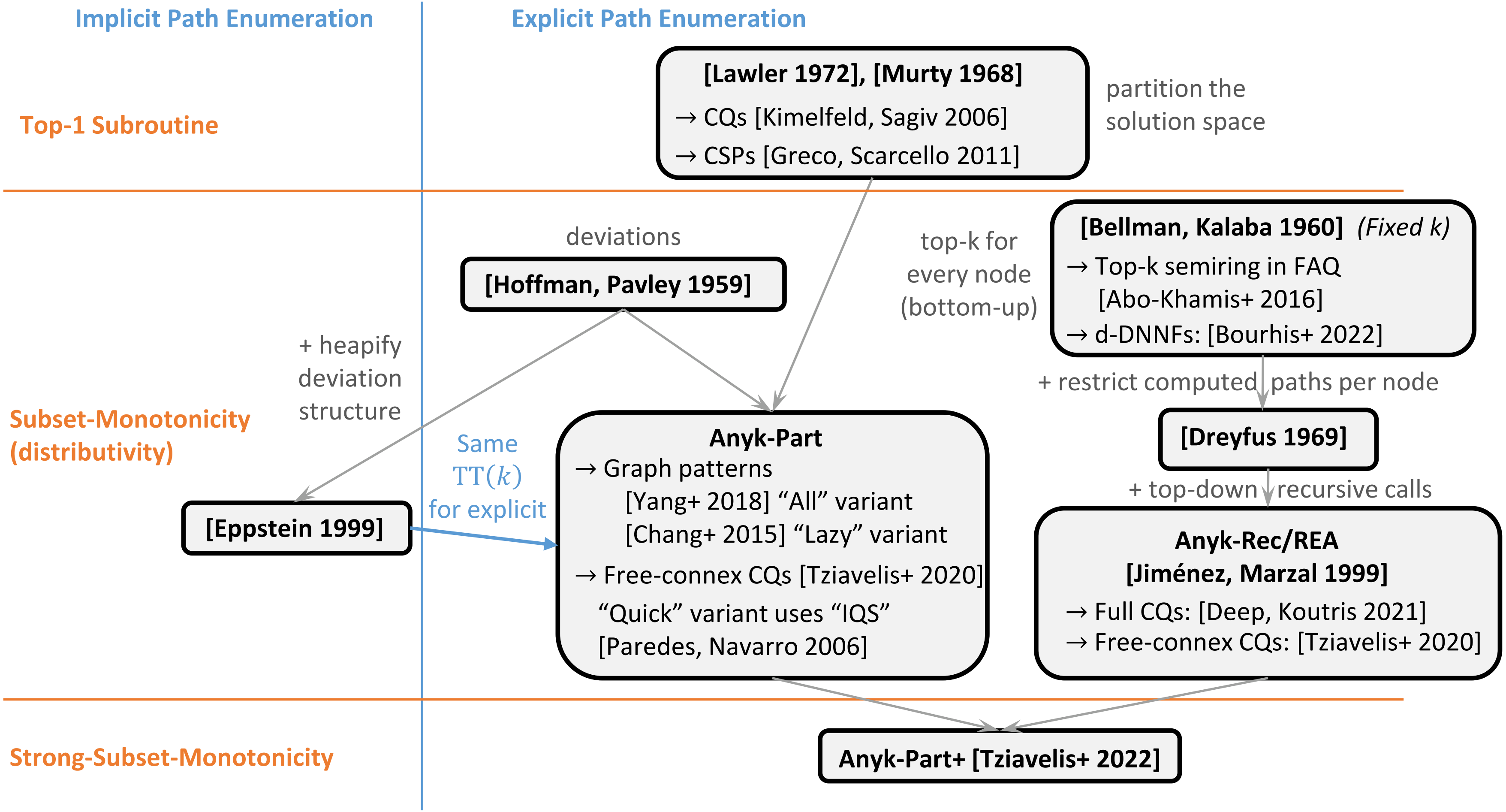
$([0, \infty], \min, \times, \infty, 1)$ ✗

1) $(0, 9, 9)$
2) $(0, 1, 1)$
3) $(0, 2, 2)$

✗

$f(1{,}9{,}9) < f(1{,}1{,}1)$

History of Algorithms for k-Shortest Paths on a DAG



Implicit Path Enumeration

Explicit Path Enumeration

Top-1 Subroutine

[Lawler 1972], [Murty 1968]
→ CQs [Kimelfeld, Sagiv 2006]
→ CSPs [Greco, Scarcello 2011]

partition the solution space

Subset-Monotonicity (distributivity)

deviations

[Hoffman, Pavley 1959]

+ heapify deviation structure

[Eppstein 1999]

Same $\mathrm{TT}(k)$ for explicit

Anyk-Part
→ Graph patterns
[Yang+ 2018] "All" variant
[Chang+ 2015] "Lazy" variant
→ Free-connex CQs [Tziavelis+ 2020]
"Quick" variant uses "IQS"
[Paredes, Navarro 2006]

top-k for every node (bottom-up)

[Bellman, Kalaba 1960] *(Fixed k)*
→ Top-k semiring in FAQ
[Abo-Khamis+ 2016]
→ d-DNNFs: [Bourhis+ 2022]

+ restrict computed paths per node

[Dreyfus 1969]

+ top-down recursive calls

Anyk-Rec/REA
[Jiménez, Marzal 1999]
→ Full CQs: [Deep, Koutris 2021]
→ Free-connex CQs: [Tziavelis+ 2020]

Strong-Subset-Monotonicity

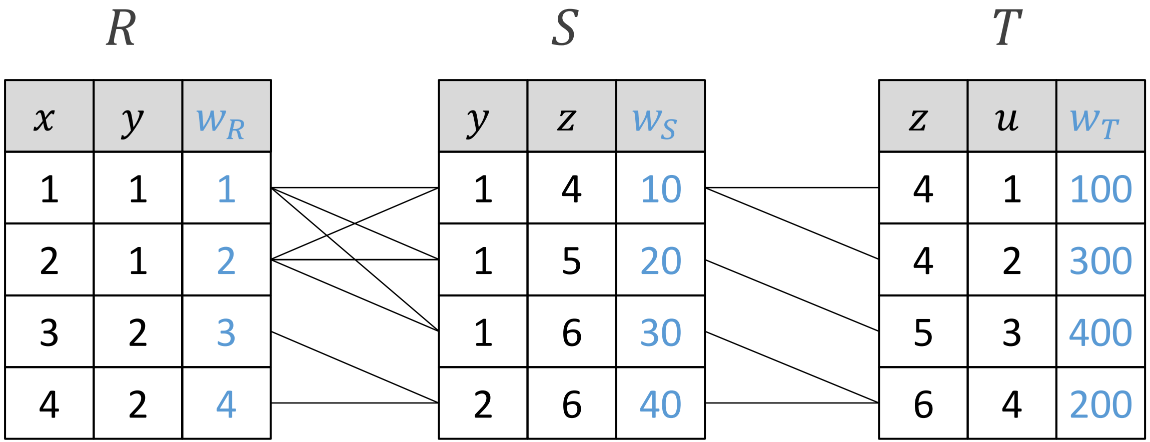
Anyk-Part+ [Tziavelis+ 2022]

Outline

- Ranked Enumeration & Dynamic Programming
  - DP as a DAG
  - Semirings
  - Any-k Algorithms
- Ranked Enumeration for (Full) Conjunctive Queries
  - Mapping CQs to DP
  - Ranking Function & Query Structure
- Conclusion

Ranked Enumeration for Conjunctive Queries

$Q(x, y, z, u) :- R(x, y), S(y, z), T(z, u)$ $\quad w_R + w_S + w_T$



R

x	y	w_R
1	1	1
2	1	2
3	2	3
4	2	4

S

y	z	w_S
1	4	10
1	5	20
1	6	30
2	6	40

T

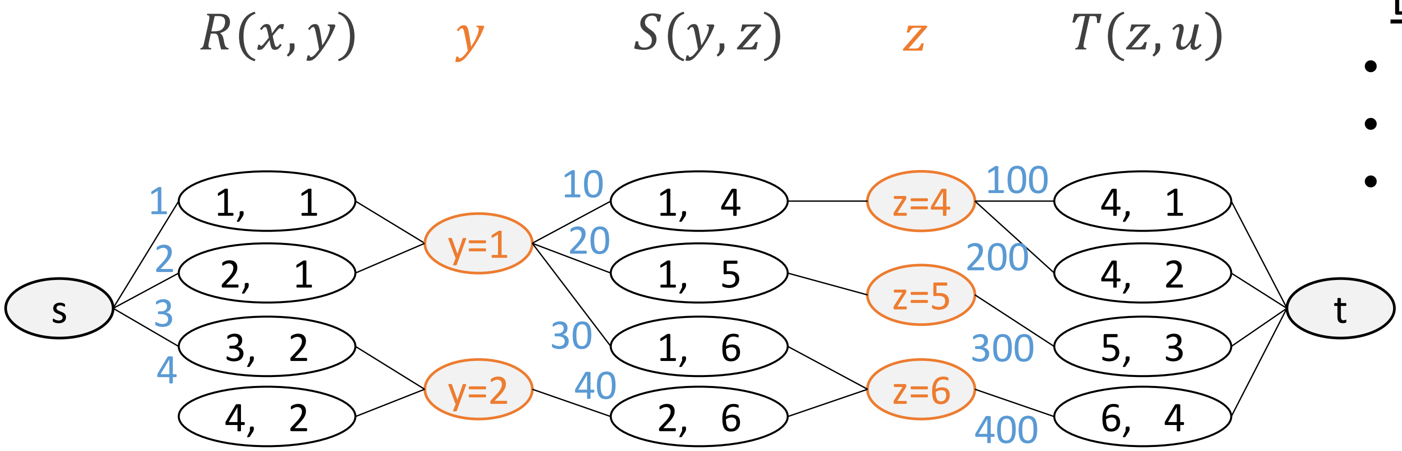
z	u	w_T
4	1	100
4	2	300
5	3	400
6	4	200

Increasing
sum of weights

(1, 1, 4, 1, 111) → (2, 1, 4, 1, 112) → (1, 1, 6, 4, 231) → ...

Path CQ → DP

$Q(x,y,z,u) :- R(x,y), S(y,z), T(z,u)$ $\quad w_R + w_S + w_T$



DAG

- Nodes = Tuples
- Edges = Joining pairs
- Paths = Join answers

Factorization

$O(n^2) \to O(n)$

Also: $O(n \text{ polylog } n)$ for inequality joins

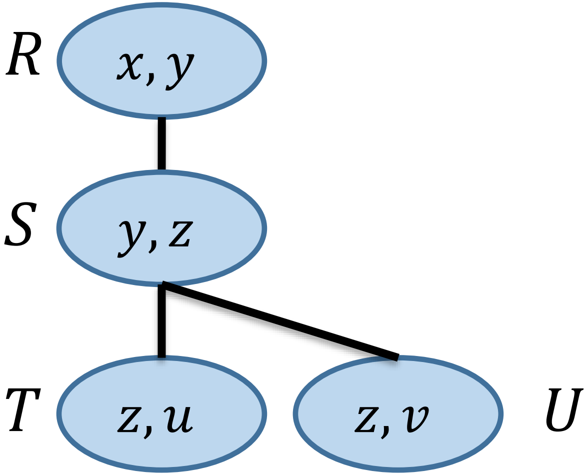
Tziavelis, Gatterbauer, Riedewald. Beyond Equi-joins: Ranking, Enumeration and Factorization. PVLDB’21 https://doi.org/10.14778/3476249.3476306

Join Trees

- Acyclic CQs ⇔ Join Trees

$Q(x, y, z, u) :- R(x, y), S(y, z), T(z, u), U(z, v)$

- Join Tree:
  - Atoms as nodes
  - For each variable X, the nodes containing X are connected



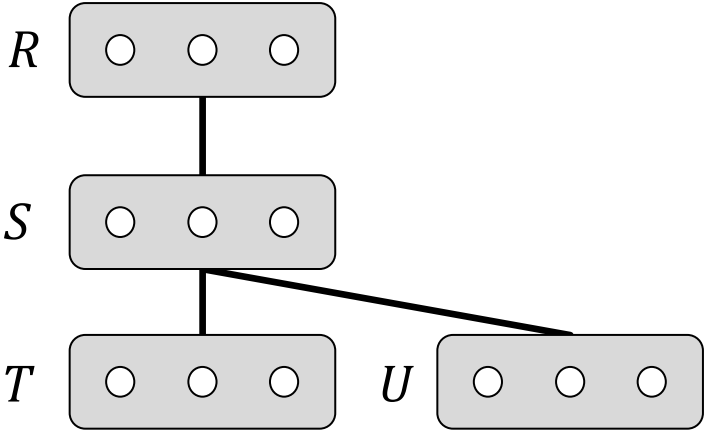
Path → Tree → Cyclic

DP

Hypertree decompositions

Tree Queries

Tree-DP
(Non-serial Dynamic Programming
with "diverging branches")



Guarantees for Full Acyclic CQs

n: #tuples
ℓ: #atoms
α: arity

- Data Complexity & Subset-Monotonicity [TAGRY20,DK21]:

$$\mathrm{TT}(k) = O(n + k \log k)$$

- Combined Complexity & Subset-Monotonicity [TAGRY20]:

$$\mathrm{TT}(k) = O(n\ell\alpha + k(\log k + \ell\alpha))$$

- Combined Complexity & Strong-Subset-Monotonicity [TGR22]:

$$\mathrm{TT}(k) = O(n\ell\alpha + k(\log(\min\{k, n^{\ell-\mathrm{diam}(Q)+1}\}) + \ell\alpha)$$

[TAGRY20] Tziavelis, Ajwani, Gatterbauer, Riedewald, Yang. Optimal Algorithms for Ranked Enumeration of Answers to Full Conjunctive Queries. PVLDB’20 https://doi.org/10.14778/3397230.3397250

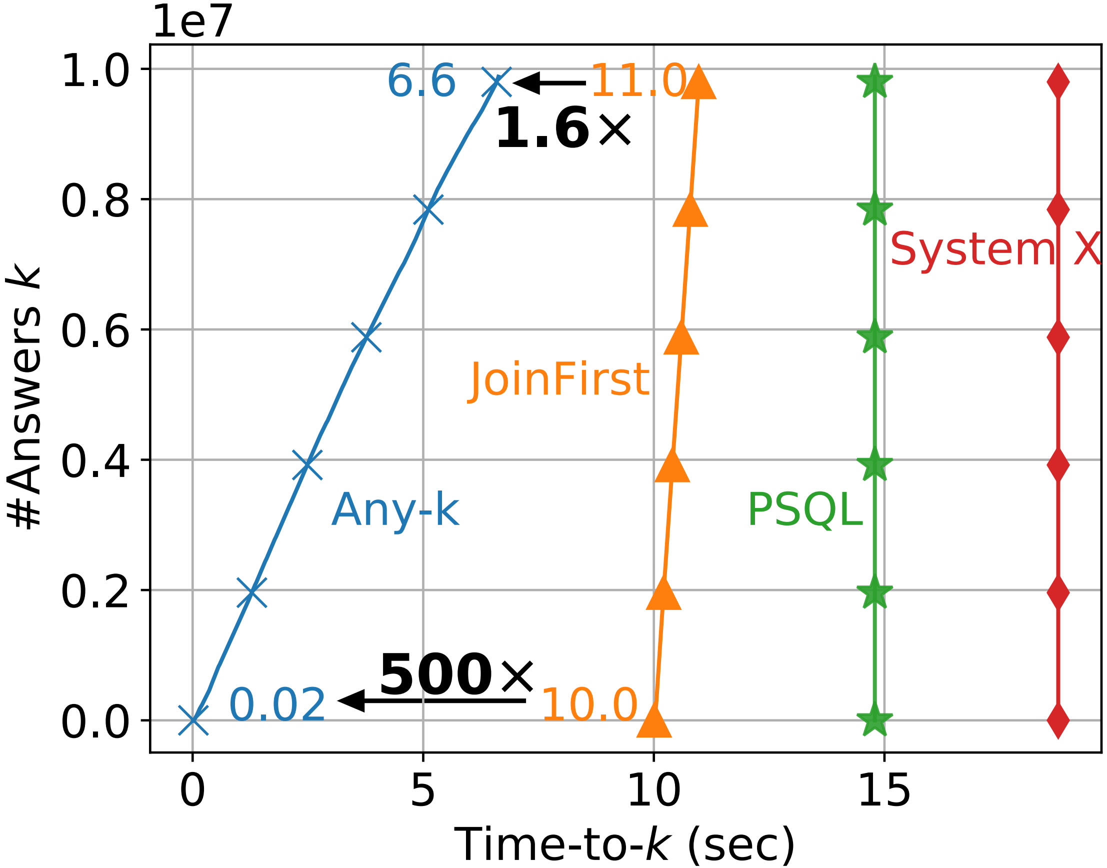
[DK21] Deep, Koutris. Ranked Enumeration of Conjunctive Query Results. ICDT’21 https://doi.org/10.4230/LIPIcs.ICDT.2021.5

[TGR22] Tziavelis, Gatterbauer, Riedewald. Any-k Algorithms for Enumerating Ranked Answers to Conjunctive Queries. arXiv’22 https://arxiv.org/abs/2205.05649

Ranked Enumeration in Practice

Very Large Data Bases

RESULTS reproduced



[TAGRY20] Tziavelis, Ajwani, Gatterbauer, Riedewald, Yang. Optimal Algorithms for Ranked Enumeration of Answers to Full Conjunctive Queries. PVLDB'20 https://doi.org/10.14778/3397230.3397250

Outline

- Ranked Enumeration & Dynamic Programming
  - DP as a DAG
  - Semirings
  - Any-k Algorithms
- Ranked Enumeration for (Full) Conjunctive Queries
  - Mapping CQs to DP
  - Ranking Function & Query Structure
- Conclusion

Lexicographic Orders

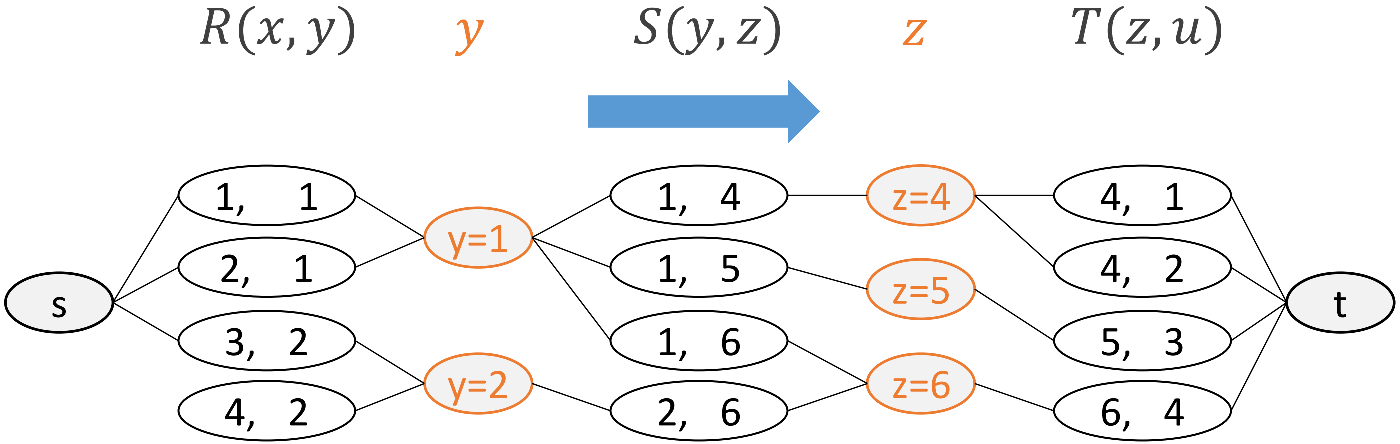
$Q(x, y, z, u) :- R(x, y), S(y, z), T(z, u)$ $x \rightarrow y \rightarrow z \rightarrow u$

- Lexicographic order as semiring:
  - Option 1: Map to $(\min, +)$ semiring with appropriate weights
  - Option 2: Define semiring on tuples with one position per variable and two appropriate operations (lexicographic min, union)
- Logarithmic delay for any lexicographic order
- Can we do better by taking into account the structure of the query?

Lexicographic Orders

$$Q(x,y,z,u) :- R(x,y), S(y,z), T(z,u)$$

$x \rightarrow y \rightarrow z \rightarrow u$



Constant-delay enumeration if lexicographic order agrees with a (reverse) α-elimination order for the query. [BKOZ13, BDG07]

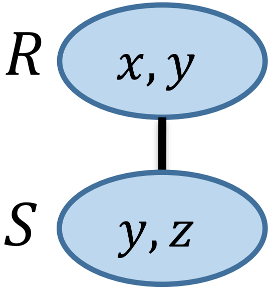
[BDG07] Bagan, Durand, Grandjean. On Acyclic Conjunctive Queries and Constant Delay Enumeration. CSL'07 https://doi.org/10.1007/978-3-540-74915-8_18
[BKOZ13] Bakibayev, Kočiský, Olteanu, Závodný. VLDB'13 https://doi.org/10.14778/2556549.2556579

Ranking Function Compatible with Tree Decomposition

$Q(x, y, z) :- R(x, y), S(y, z)$

$g(x, y) + g(y, z)$

Not subset-monotone



$g(x, y)$

$+$

$g(y, z)$

✓

Logarithmic delay for ranking functions that are compatible with a tree decomposition (which determines preprocessing). [DK21]

[DK21] Deep, Koutris. Ranked Enumeration of Conjunctive Query Results. ICDT’21 https://doi.org/10.4230/LIPIcs.ICDT.2021.5

Outline

- Ranked Enumeration & Dynamic Programming
  - DP as a DAG
  - Semirings
  - Any-k Algorithms
- Ranked Enumeration for (Full) Conjunctive Queries
  - Mapping CQs to DP
  - Ranking Function & Query Structure
- Conclusion

Conclusion

- Same ranked enumeration algorithms appear in many different problems and the common link is DP and semirings
- Different monotonicity notions allow for different algorithms
- We can potentially do more if we take into account the structure of the problem
- Practical results outperforming database systems by orders of magnitude

Open Questions

- Precise characterization of tractable queries + ranking functions.
  - Lower bounds
  - Algorithms for Holistic-Monotone ranking functions (e.g., MEDIAN)?
  - Can we leverage the structure of the query more to cover more cases?
- Understand better the relationship between $\mathrm{TT}(k)$ and delay
- Relationship of this DP framework to circuits
  - Can all these algorithms be carried over?

Thank you!

Website: https://northeastern-datalab.github.io/anyk/