Tractable Orders for Direct Access to Ranked Answers of Conjunctive Queries

Nofar Carmeli[1], Nikolaos Tziavelis[2],
Wolfgang Gatterbauer[2], Benny Kimelfeld[1], Mirek Riedewald[2]

[1]Technion, Israel
[2]Northeastern University, Boston

Motivation

Employees

Name	Role	Address
Jack	Junior dev	Boston
Jill	Senior dev	Brookline
Joanna	Senior dev	Braintree

Renumeration

Period	Role	Salary
11/2020	Junior dev	4000
11/2020	Senior dev	4500
12/2020	Junior dev	7000
12/2020	Senior dev	7100

Travel

Address	Cost
Boston	50
Brookline	100
Braintree	200

$$Q(\text{Name}, \text{Role}, \text{Address}, \text{Period}, \text{Salary}, \text{Cost}) \leftarrow \text{Employees}(\text{Name}, \text{Role}, \text{Address}), \text{Renumeration}(\text{Period}, \text{Role}, \text{Salary}), \text{Travel}(\text{Role}, \text{Salary})$$

Query Answers

sort by salary+cost

Name	Role	Address	Period	Salary	Cost
Jack	Junior dev	Boston	11/2020	4000	50
Jill	Senior dev	Brookline	11/2020	4500	100
Joanna	Senior dev	Braintree	11/2020	4500	200
Jack	Junior dev	Boston	12/2020	7000	50
…					

4th result

Want:

- Median
- Boxplot
- Jump to any rank

without materializing all answers

Outline

- Direct access: Problem & Background
- Lexicographic orders
- Sum-of-weights orders
  - Selection Problem
- Conclusion

Ranked Direct Access Problem

- Also called: random access, jth answer

- Problem: query & ordering
- Input: database instance of size n
- Algorithm:
  - Preprocessing
  - Access: given k, return answer k in the sorted list of answers (or out-of-bound)

- Our focus: quasilinear preprocessing, polylog access time

$<n \operatorname{polylog} n, \operatorname{polylog} n>$ (data complexity)

Answer Orderings

Lexicographic

- Ordering of free variables
  e.g. [Address, Salary, Cost, Role, Name, Period]
- or just [Address, Salary] (partial lex. order)

Sum-of-weights

- Weights to domain values of free variables
- Ranking by the sum of weights
- Can simulate any lexicographic order

Salary	w
4000	1
4500	2
7000	3

Address	w
Boston	10
Braintree	20
Brookline	30

Name	Role	Address	Period	Salary	Cost	w
Jack	Junior dev	Boston	11/2020	4000	50	11
Jack	Junior dev	Boston	12/2020	7000	50	13
Joanna	Senior dev	Braintree	11/2020	4500	200	22
Jill	Senior dev	Brookline	11/2020	4500	100	32

Equivalent to [Address, Salary]

Related work

- (Unranked) Enumeration [BaganDurandGrandjean CSL'07] [Brault-Baron PhD Thesis 13]
  const (or polylog) delay possible ⇔* free-connex

- Ranked enumeration [TAjwaniGatterbauerRiedewaldYang PVLDB'20]
  sum of weights (or lexicographic), log delay, free-connex

- Direct access (restricted order support)
  - via elimination order [Brault-Baron PhD Thesis 13]
  - via join tree [CZeeviBerkholzKimelfeldSchweikardt PODS'20]
  - via q-tree (dynamic settings, q-hierarchical only) [Keppeler PhD Thesis 20]

All using: data complexity, RAM model

Definitions

An acyclic CQ has a graph with:

A free-connex CQ also requires:

1. a node for every atom
possibly also subsets

2. tree

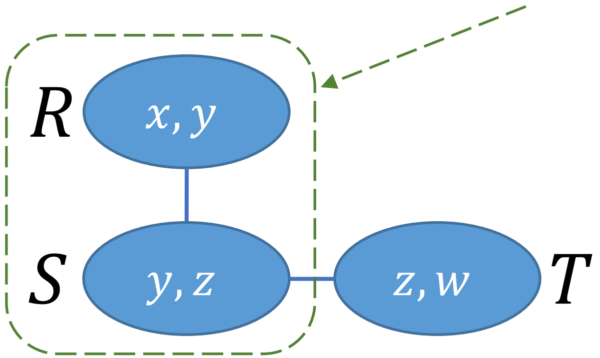
3. for every variable X:
the nodes containing X
are connected

4. a subtree with
exactly the free
variables

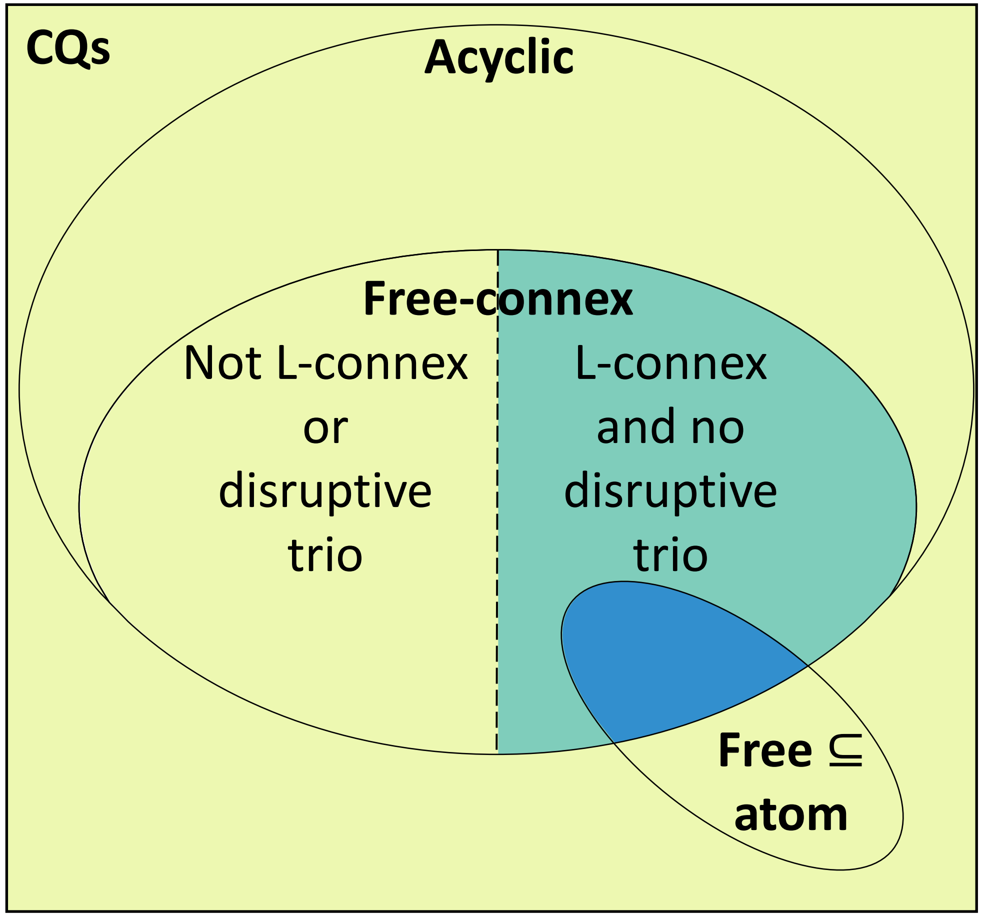
free − connex

acyclic

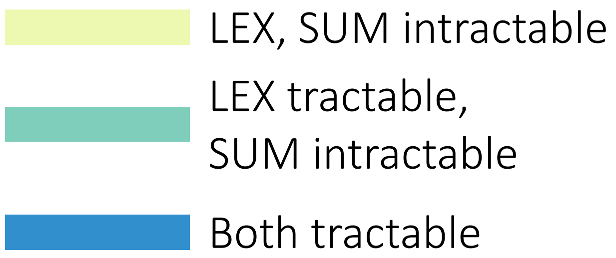
$$Q(x,y,z) \leftarrow R(x,y), S(y,z), T(z,w)$$



Overview of Direct Access



Tractable≡ <n polylog n , polylog n>



* Lower bounds assume: no self-joins, conventional hypotheses in fine-grained complexity

Outline

- Direct access: Problem & Background
- Lexicographic orders
- Sum-of-weights orders
  - Selection Problem
- Conclusion

Dichotomy

✓ $Q_1(v_1, v_2, v_3) \leftarrow R(v_1, v_2), S(v_2, v_3)$

✗ $Q_2(v_1, v_2, v_3) \leftarrow R(v_1, v_3), S(v_3, v_2)$

Given: CQ Q, ordering L of free(Q),

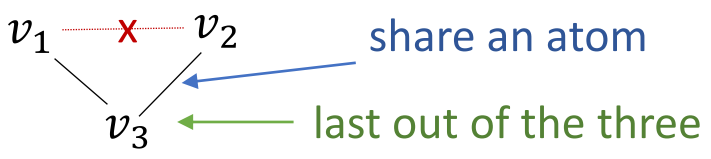
lexicographic access in $<n \text{ polylog } n, \text{polylog } n>$

$\Updownarrow$*

free-connex, no disruptive trio

* Lower bounds assume:
  (1) no self-joins
  (2) hardness of matrix multiplication and hyperclique detection

Disruptive Trio



Partial Lexicographical Ordering

- possible $\Leftrightarrow$ a completion for a feasible full ordering

a subset of

Given: CQ Q, ordering L of free(Q),

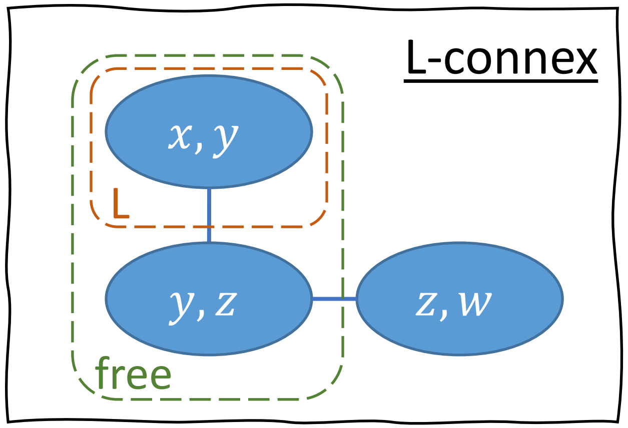
partial

lexicographic access in $\langle n \operatorname{polylog} n, \operatorname{polylog} n \rangle$

$\Updownarrow$*

L-connex,

free-connex, no disruptive trio



* Lower bounds assume:

(1) no self-joins

(2) hardness of matrix multiplication and hyperclique detection

Hardness

- Assumption: Q_1 cannot be enumerated in $\langle n\,\text{polylog}\,n, \text{polylog}\,n\rangle$
- Reduction:

binary search for next different v_1, v_2 values

v_1	v_2	v_3
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_1	c_3
a_1	b_1	c_4
a_1	b_1	c_5
a_1	b_2	c_1
a_1	b_2	c_2
a_2	b_1	c_1

Enumerate

$$Q_1(v_1, v_2) \leftarrow R(v_1, v_3), S(v_3, v_2)$$

using

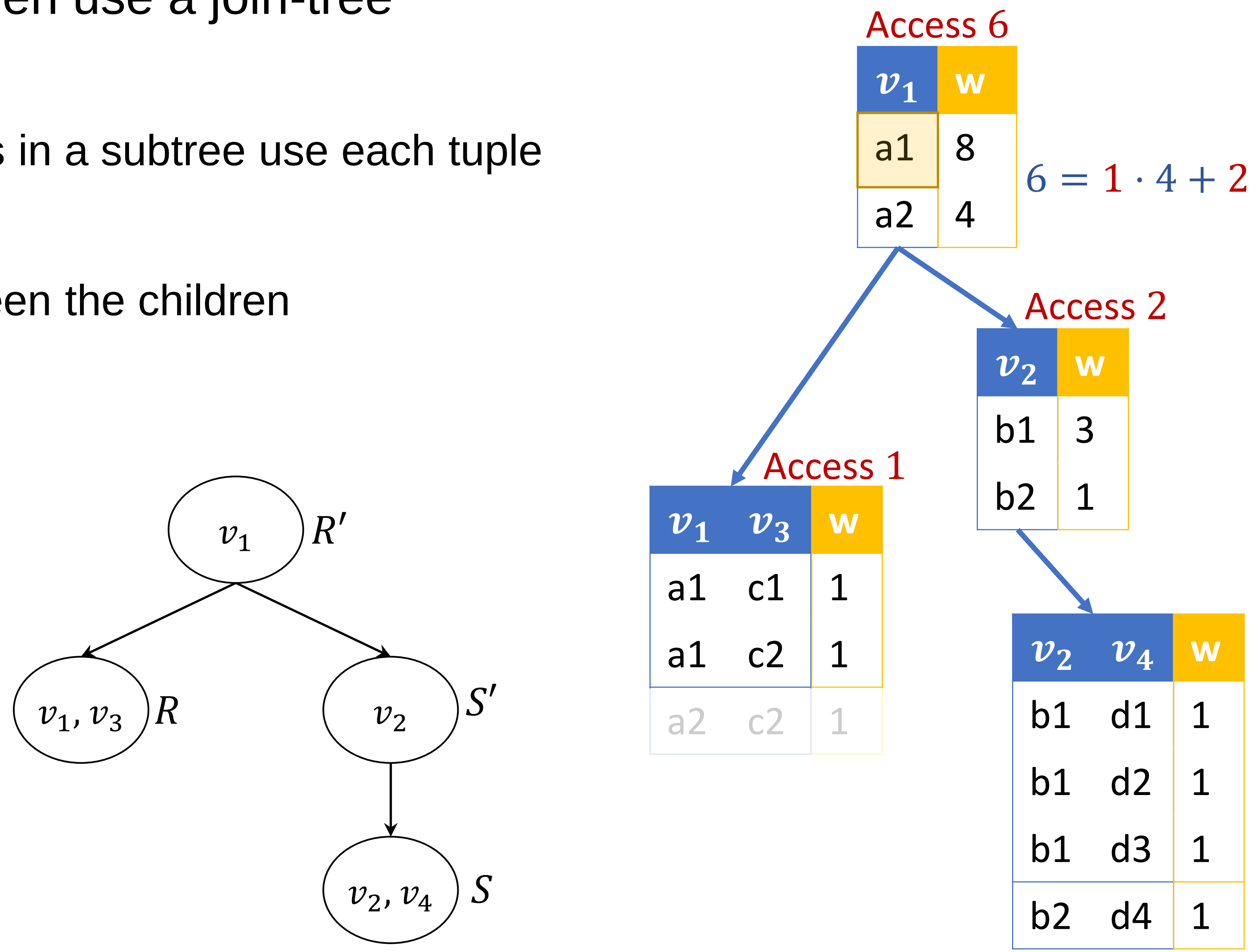
Lexicographic direct-access

$$Q_2(v_1, v_2, v_3) \leftarrow R(v_1, v_3), S(v_3, v_2)$$

- Log number of direct-access calls between answers
  - Q_1 enumeration with polylog delay, contradiction

Algorithm

- Adopt previous approach [C Zeevi Berkholz Kimelfeld Schweikardt PODS'20]
  - Free-connex to full acyclic, then use a join-tree
  - Preprocessing:
    - DP up the tree
    - computes how many answers in a subtree use each tuple
  - Access:
    - recurse down the tree
    - splits the desired index between the children

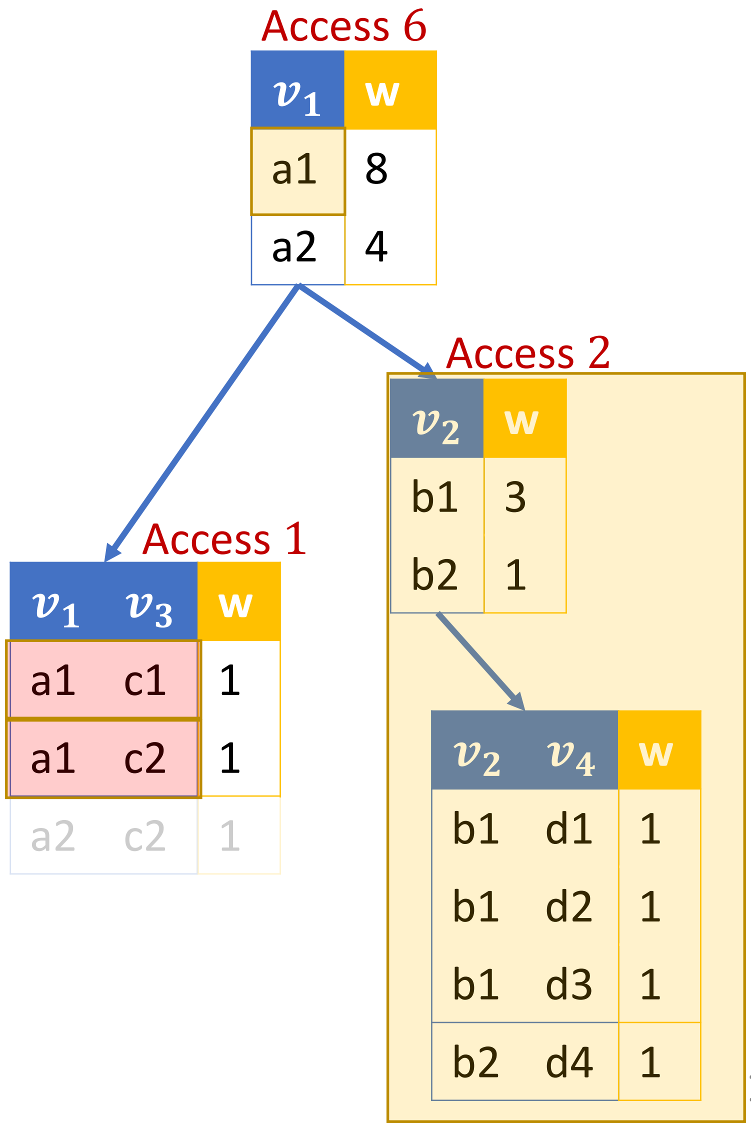


Algorithm

- Adopt previous approach [C Zeevi Berkholz Kimelfeld Schweikardt PODS'20]
  - Free-connex to full acyclic, then use a join-tree
  - Preprocessing:
    - DP up the tree
    - computes how many answers in a subtree use each tuple
  - Access:
    - recurse down the tree
    - splits the desired index between the children

Resulting order:

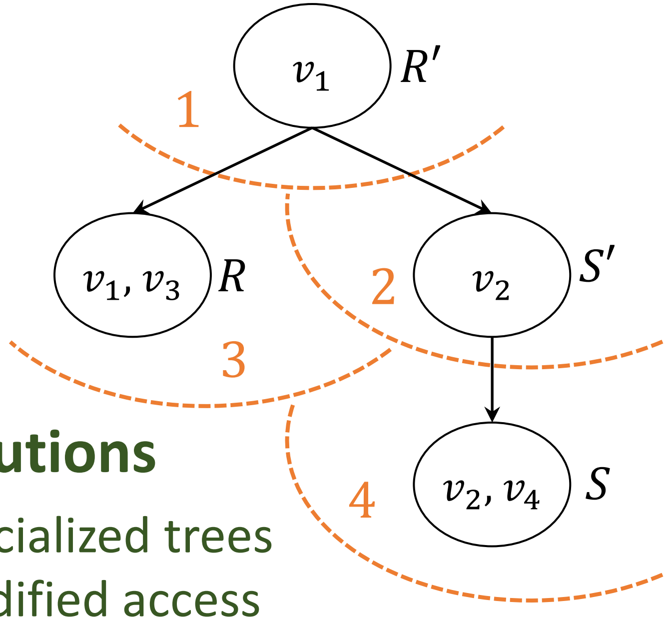
v_1	v_3	v_2	v_4
a1	c1	b1	d1
a1	c1	b1	d2
a1	c1	b1	d3
a1	c1	b2	d4
a1	c2	b1	d1
a1	c2	b1	d2
a1	c2	b1	d3
a1	c2	b2	d4
			…



Algorithm

- Adopt previous approach [C Zeevi Berkholz Kimelfeld Schweikardt PODS'20]
  - Free-connex to full acyclic, then use a join-tree
  - Preprocessing:
    - DP up the tree
    - computes how many answers in a subtree use each tuple
  - Access:
    - recurse down the tree
    - splits the desired index between the children

no disruptive trio $\Longrightarrow$ layered join-tree



v_1	w
a1	8
a2	4

v_1	v_3	w
a1	c1	1
a1	c2	1
a2	c2	1

v_2	w
b1	3
b2	1

v_2	v_4	w
b1	d1	1
b1	d2	1
b1	d3	1
b2	d4	1

Problems

1. Tree determines order
2. Independent branches

Solutions

1. Specialized trees
2. Modified access

Outline

- Direct access: Problem & Background
- Lexicographic orders
- Sum-of-weights orders
  - Selection Problem
- Conclusion

Dichotomy

Given: CQ Q,

sum-of-weights access in $<n \operatorname{polylog} n, \operatorname{polylog} n>$

⇕*

acyclic, an atom contains all free variables

$Q_1(x, z) \leftarrow R(x, y, z), S(y, z)$ ✓

$Q_2(x, z) \leftarrow R(x, y), S(y, z)$ ✗

* Lower bounds assume:
  (1) no self-joins
  (2) hardness of 3-SUM and hyperclique detection

Hardness

- Observation: Binary search finds a weight with logarithmic accesses

3SUM hypothesis
given 3 sets of integers $|A| = |B| = |C| = n$,
deciding $\exists\ a \in A, b \in B, c \in C$ s.t. $a + b + c = 0$
cannot be done in time $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$

- Use two independent free variables

$$Q_2(x, z) \leftarrow R(x, y), S(y, z)$$

Direct access impossible in $\langle n^{2-\varepsilon}, n^{1-\varepsilon} \rangle$

x	y
a_1	0
a_2	0

A

y	z
0	b_1
0	b_2

B

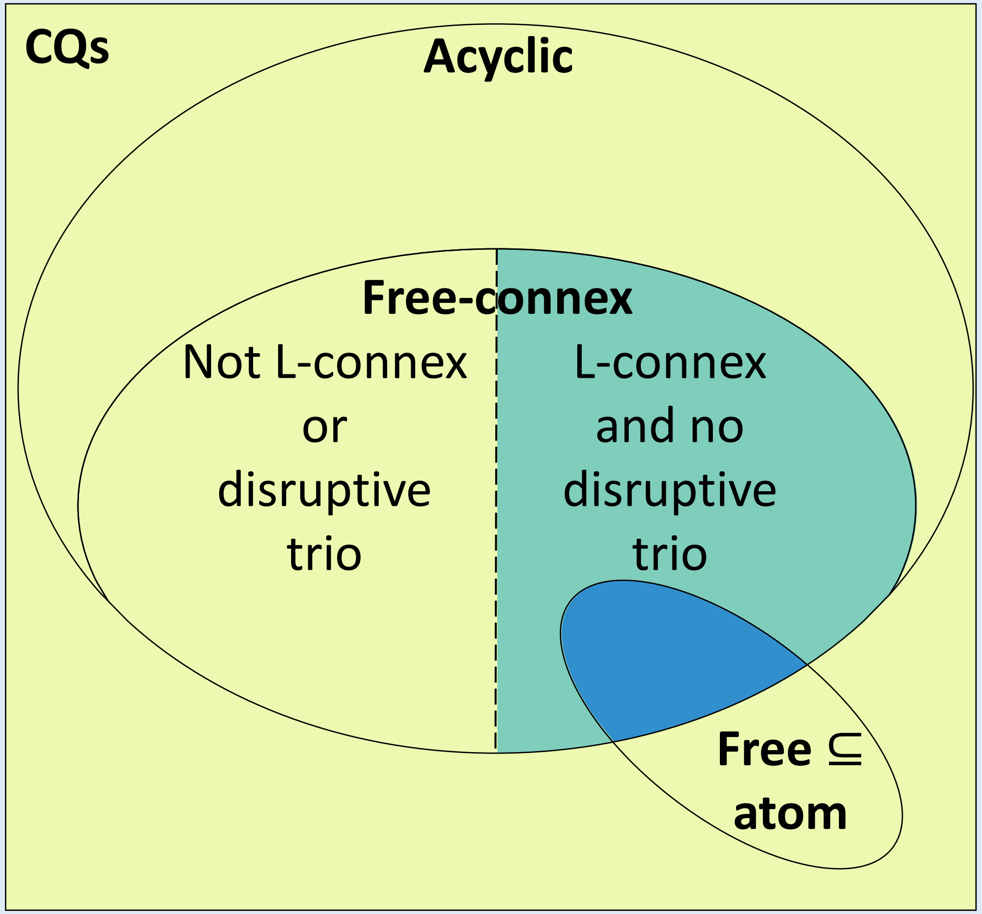
x	y	z	w
a_1	0	b_1	$a_1 + b_1$
a_1	0	b_2	$a_1 + b_2$
a_2	0	b_1	$a_2 + b_1$
a_2	0	b_2	$a_2 + b_2$

Binary search for $-c$ ($\forall c$)

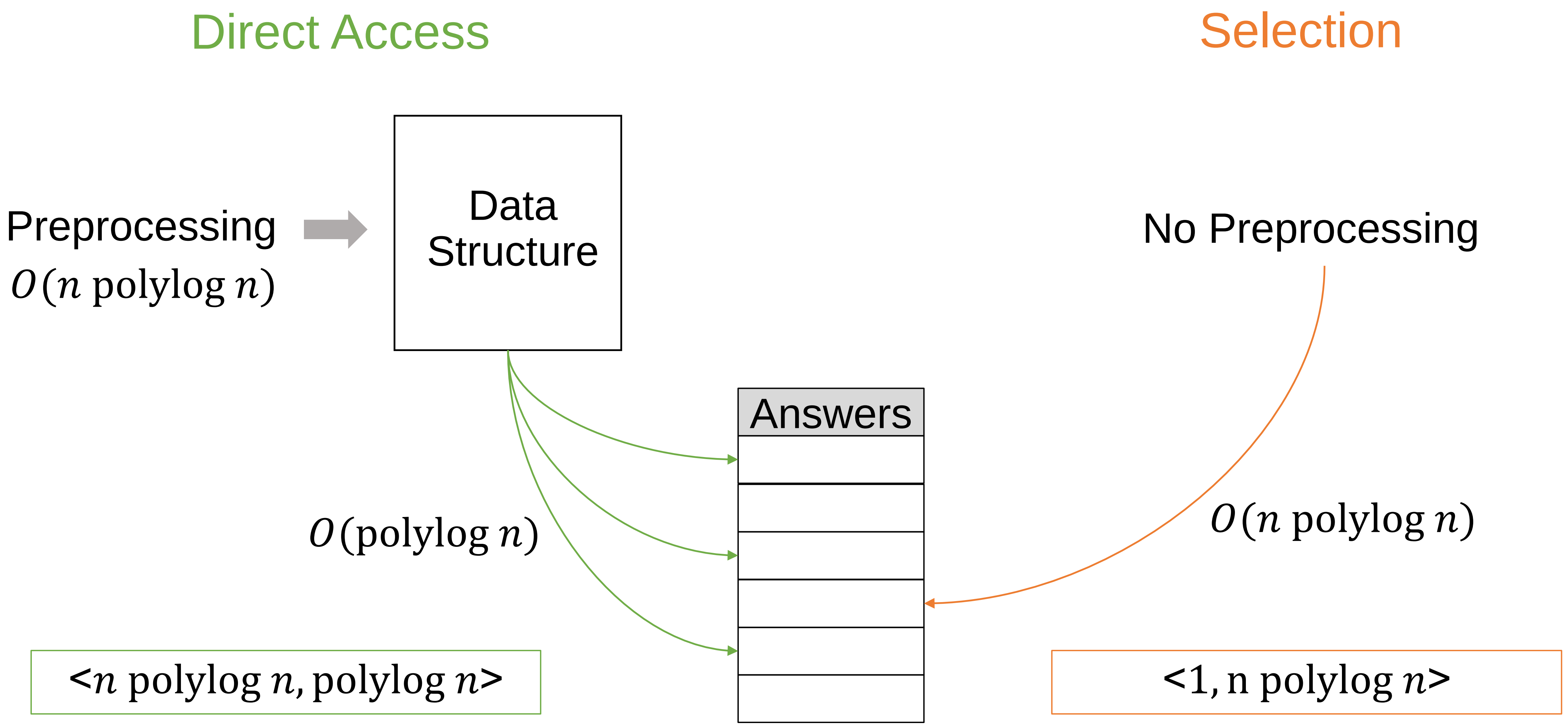
Outline

- Direct access: Problem & Background
- Lexicographic orders
- Sum-of-weights orders
  - Selection Problem
- Conclusion

Direct Access



Selection vs Direct Access



Selection Dichotomy

Given: **full** CQ Q,

sum-of-weights selection in $O(nlogn)$

⇕*   w.r.t. hyperedge containment

At most two maximal atoms

$Q_1(x, y, z) \leftarrow R(x, y), S(y, z), T(y)$ ✓

$Q_2(x, y, z, u) \leftarrow R(x, y), S(y, z), T(z, u)$ ✗

* Lower bounds assume:
  (1) no self-joins
  (2) hardness of 3-SUM and hyperclique detection

Tractable Cases for Selection

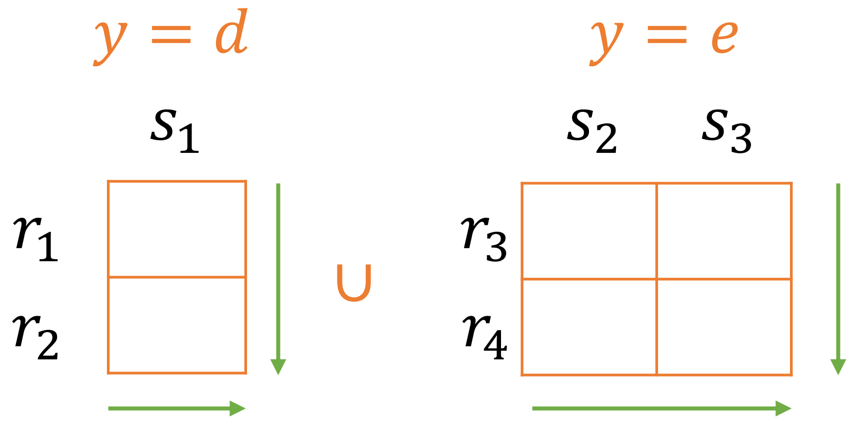
$$Q(x, y, z) \leftarrow R(x, y), S(y, z)$$

R

	x	y
r_1	a	d
r_2	b	d
r_3	a	e
r_4	c	e

S

y	z	
d	f	s_1
e	f	s_2
e	g	s_3



Sort R, S on weights =>
Every row/column sorted

Selection on a union of sorted matrices
of dimensions $m_i \times n_i$
possible in time $O(\sum \max(m_i, n_i))$

[Frederickson Johnson 1984]

* We do not materialize the matrices, but compute the values of the cells on-the-fly

Tractable Cases for Selection (Lexicographic)

$$Q(x, y, z) \leftarrow R(x, y), S(y, z)$$

Lex Order $[x, z, y]$:

- Direct access intractable (disruptive trio)
- Selection tractable (as sum-of-weights)

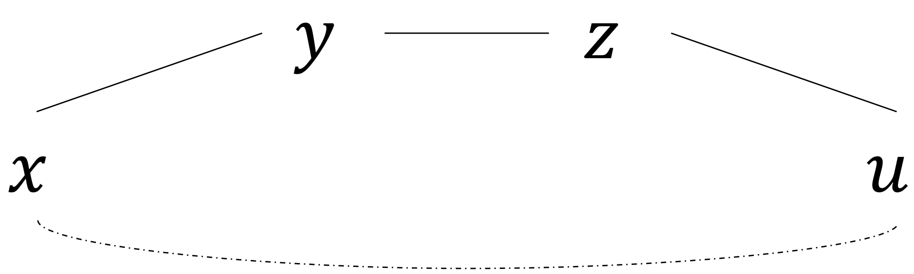
Intractable Cases for Selection

- 3-path Q_{3p} (reduction from Boolean triangle $Q_{\triangle}$)

$$Q_{\triangle}() \leftarrow R(x,y), S(y,z), T(z,x)$$

$$Q_{3p}(x,y,z) \leftarrow R(x,y), S(y,z), T(z,u)$$

Sum-of-weights Selection



$$w(x) + w(u) = 0 \Leftrightarrow x = u$$

- Identify answers to $Q_{\triangle}$ with $x = u$
- Set weights s.t. weight of “triangle” answers becomes 0
- Binary search to find zero weight answer

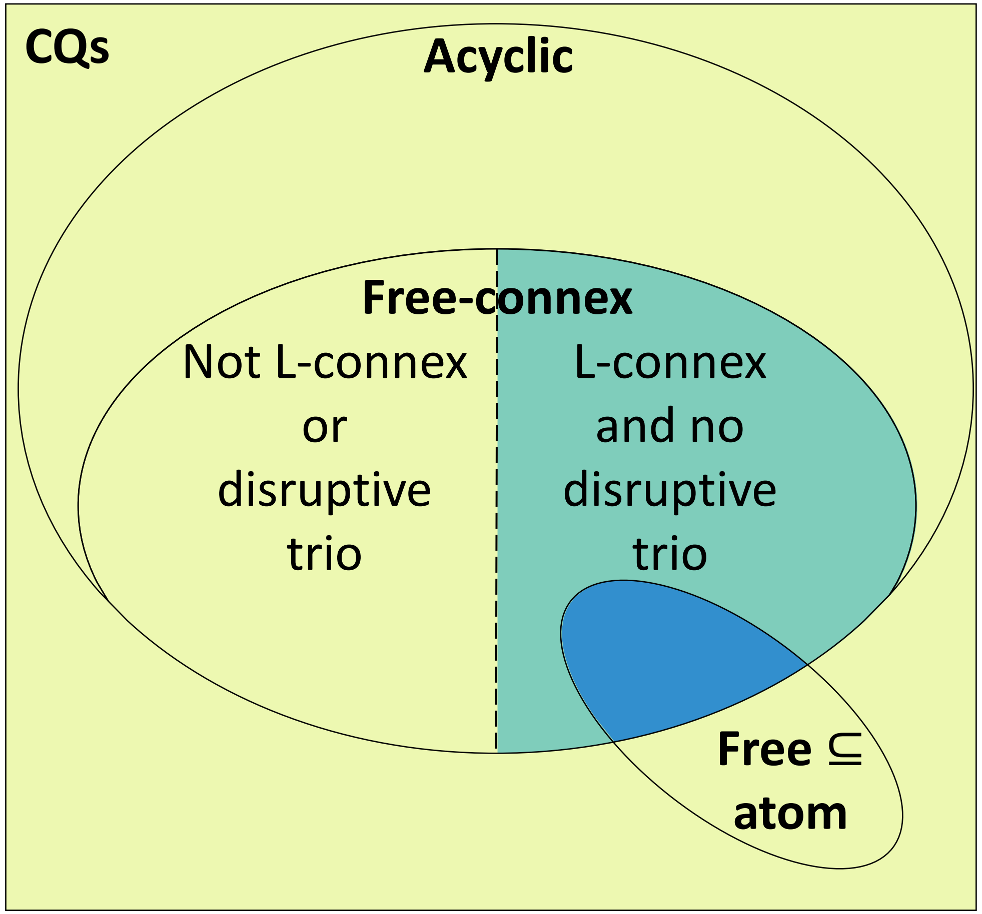
Outline

- Direct access: Problem & Background
- Lexicographic orders
- Sum-of-weights orders
  - Selection Problem
- Conclusion

Overview and Conclusion

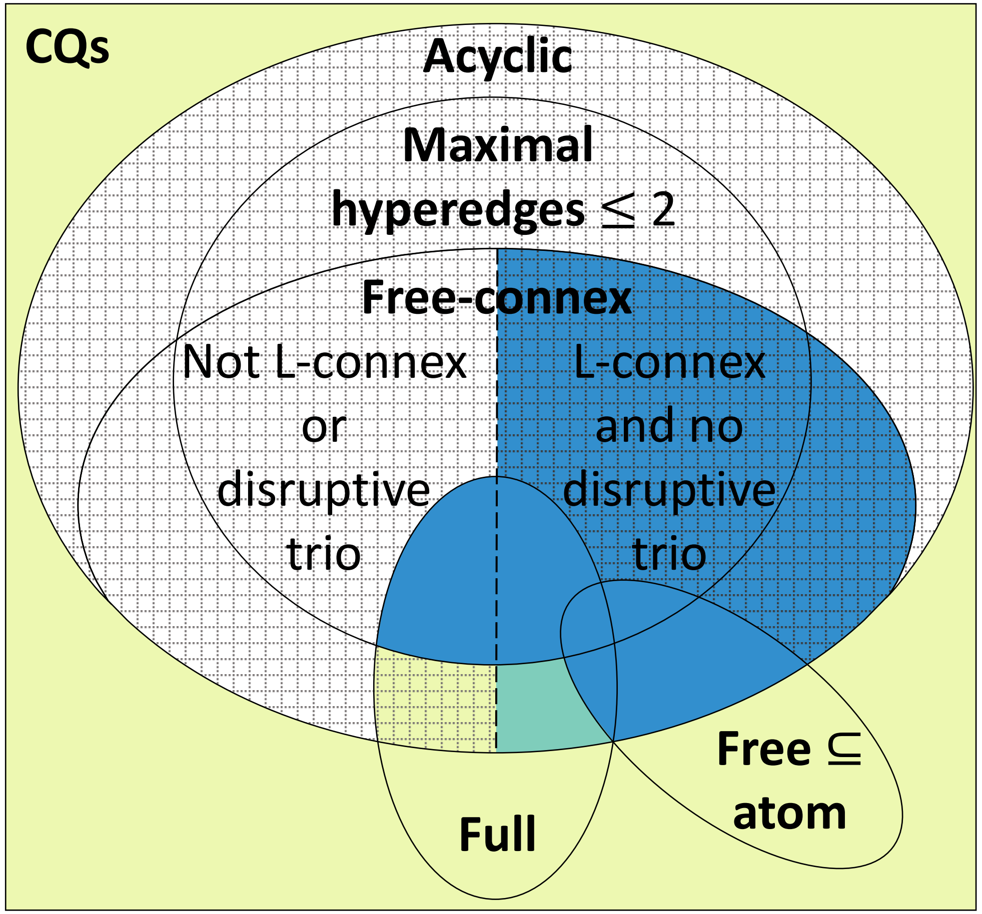
Direct Access

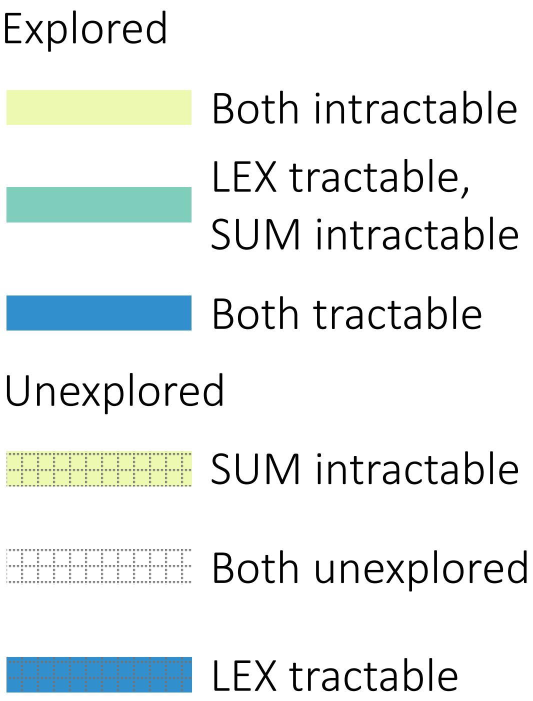
Tractable≡<n polylog n , polylog n>



Selection

Tractable≡ <1, n polylog n >





* Upper bounds for direct access are <sorting-cost, log n>
* Lower bounds assume: no-self joins, hypotheses in fine-grained complexity

Thank you!