

Acyclic CQs With Self-Joins And Projections

Towards a Dichotomy for Constant-Delay Answer Enumeration

Clment Rouvroy, Nofar Carmeli, David Carral, Luc Segoufin

Question

Which acyclic CQs with self-joins and projections can be answered with linear preprocessing and constant delay?

Known [Bagan, Durand, Grandjean 2007; Brault-Baron 2013; Carmeli, Segoufin 2023]

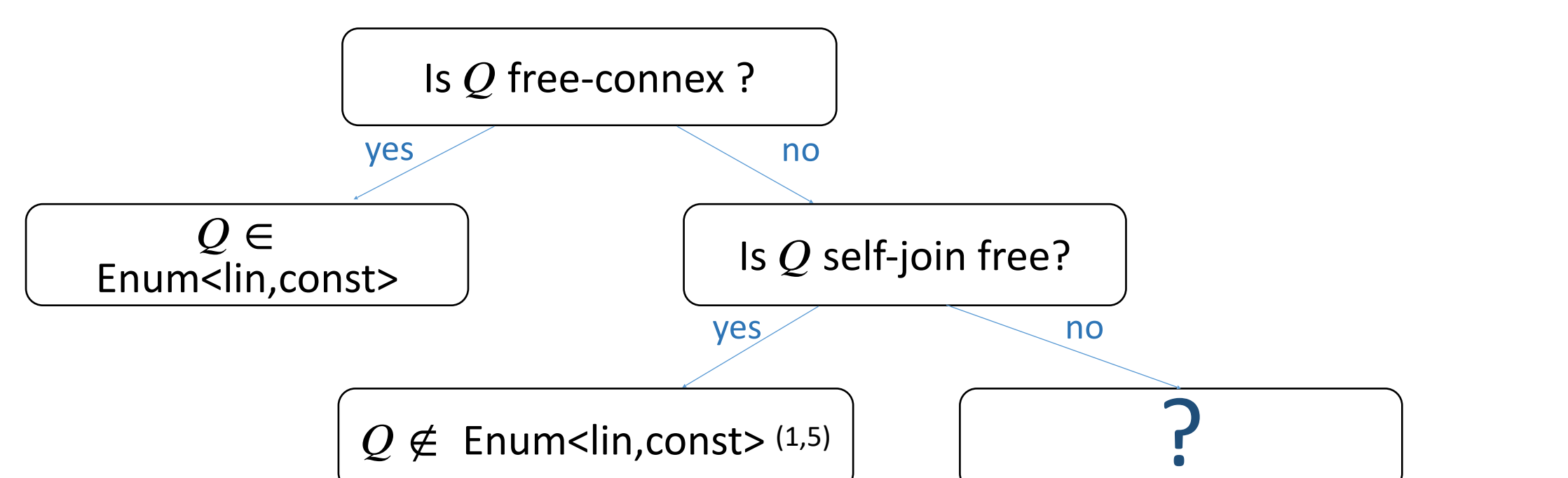
Given a self-join-free conjunctive query Q ,
 $Q \in \text{Enum}\langle \text{lin}, \text{const} \rangle \stackrel{(1,5)}{\Leftrightarrow} Q$ is free-connex acyclic

Definition of $\text{Enum}\langle \text{lin}, \text{const} \rangle$: queries solvable with linear preprocessing and constant delay in the RAM model in data complexity.

Definition of acyclic and free-connex:

1. There is a Tree
2. With a node for every atom (possibly also subsets)
3. For every variable: the nodes containing it form a subtree
4. There is a subtree with exactly the free variables

We fix Q to be acyclic. Full cyclic queries with self-joins were studied by [Carmeli, Segoufin 2023]. We focus here on projections.



A Free-path can be easy with Self-Joins

Assumption: cannot compute the product of two $n \times n$ matrices in $O(n^2)$ (2)

$Q_1(x_1, x_2, x_3) :- \exists z. x_1 \xrightarrow{R_1} z \xrightarrow{R_2} x_2$ No self-join, with a free-path. $Q_1 \notin \text{CDLin}$ (1,5)

$Q_2(x_1, x_2, x_3) :- \exists z. x_1 \xrightarrow{R_1} z \xrightarrow{R_2} x_2$ With self-join, with a free-path. $Q_2 \in \text{CDLin}$

Proof. One can enumerate $Q_3 = x_1 \xrightarrow{R_1} x_3 \xrightarrow{R_2} x_2$ in CDLin. But:

$$a \xrightarrow{R_1} b \xrightarrow{R_2} c \in D \quad \Rightarrow \quad a \xrightarrow{R_1} b \xrightarrow{R_2} c \in D$$

- Hence, (1) Enumerate Q_3 and build $R_4(a, c) = \{(a, c) \mid \exists b. (a, b, c) \in Q_3(D)\}$
 (2) Enumerate $Q_4(x_1, x_2, x_3) = x_1 \xrightarrow{R_3} x_2$ to get all answers of Q_2 . Conclude using Cheater's Lemma.

Some queries with free-paths and self-joins are in CDLin.

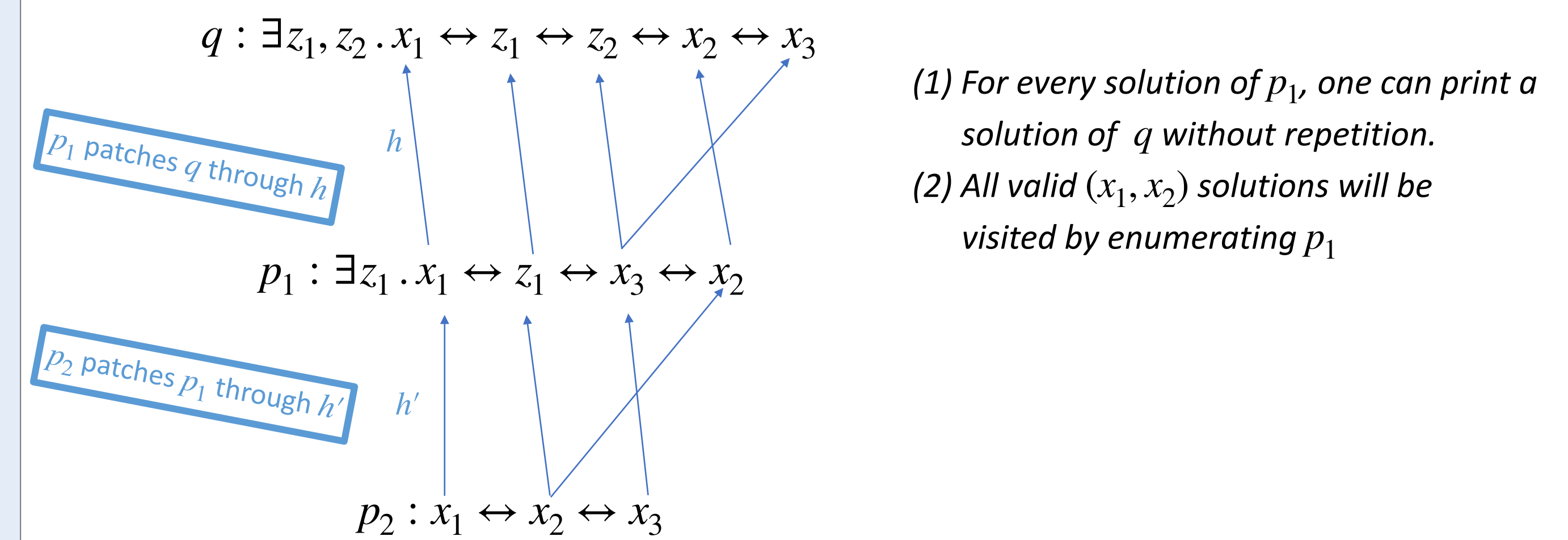
Our results

A recursive criterion that patches free-paths by materializing sub-relations using endomorphisms.

Two hardness obstructions. Let X denote all free variables
 Assuming BMM: $\exists P$ free-path. $\forall h. h(X \setminus P) \cap P = \emptyset$
 Assuming VUTD: $\exists P$ free-path. $\exists z \in P \setminus X. \forall h. z \notin h(X)$

Some interesting open problems that could help to capture both easy and hard queries.

Recursive Patching for CDLin Enumeration of free-paths



We say that q has a patching depth $\delta(q) \leq 2$, and this can be translated to an enumeration algorithm.

A query of finite patching depth is in CDLin...

- (1) One can patch every free-path of q .
 (2) If q is in CDLin, then we can detect 4-clique in $O(n^3)$.

Lemma: If two free-paths share a quantified variable, they need to be patched together.

...but we need to patch multi-frontier component, not individual free-path !

Hardness

Under BMM, a let-alone free-path implies hardness.

$$q(x_1, x_2, x_3) :- \exists z. R(x_1, x_2) \wedge R(x_2, z) \wedge R(z, x_3)$$

Let M_1, M_2 be the two matrices. We use the *tagging technique* [Carmeli, Segoufin 2023] and build:

$$\forall i \in \{1 \dots n\}. \langle x_1, \perp \rangle \rightarrow \langle x_2, i \rangle \quad \forall (i, k) \in M_1. \langle x_2, i \rangle \rightarrow \langle z, k \rangle \quad \forall (k, j) \in M_2. \langle z, k \rangle \rightarrow \langle x_3, j \rangle$$

VUTD also handles partially touched free-paths.

$$q(x_1, x_2, x_3) :- \exists z_1, z_2. x_1 \rightarrow z_1 \rightarrow z_2 \rightarrow x_2 \leftarrow x_3$$

x_3 can be mapped to z_2 but z_1 is touched by no free variable. Put A values on z_1 :



Now, there is a triangle if and only if $\exists (\langle x_1; u \rangle, \langle x_2; w \rangle, \langle x_3; w \rangle) \in q(D)$ s.t. $(u, w) \in E_{1,3}$.

But, the total running time is $O(n^{1+\alpha} + n^{3\alpha})$.

Hence, if q is in CDLin, it breaks VUTD for all $\alpha < 1/2$.

Open Problems!

Consider $q : \exists z_1, z_2. x_1 \rightarrow z_1 \rightarrow z_2 \rightarrow x_2 \wedge x_3$ with x_3 taking any domain value. Open: is q in CDLin ?

We prove that this is equivalent to the self-join-free problem:

$$\text{Is } q : \exists z_1, z_2. x_1 \xrightarrow{R_1} z_1 \xrightarrow{R_2} z_2 \xrightarrow{R_3} x_2 \text{ enumerable with delay } |dom(D)| ?$$

This motivates research on a dichotomy for DomLinLin even without self-joins.

Hypotheses

- (1) **Assuming sBMM:** Boolean matrices cannot be multiplied in linear time in the number of the 1 entries
- (2) **Assuming BMM:** Boolean $n \times n$ matrices cannot be multiplied in time $O(n^2)$
- (3) **Assuming sTriangle:** The existence of a triangle in an undirected graph with m edges cannot be decided in time $O(m)$
- (4) **Assuming Triangle:** The existence of a triangle in an undirected graph with n nodes cannot be decided in time $O(n^2)$
- (5) **Assuming sHyperclique:** $\forall k \geq 3$ the existence of a k -hyperclique in a $(k-1)$ -uniform hypergraph with m edges cannot be decided in time $O(m)$
- (6) **Assuming Hyperclique:** $\forall k \geq 3$ the existence of a k -hyperclique in a $(k-1)$ -uniform hypergraph with n nodes cannot be decided in time $O(n^{k-1})$
- (7) **Assuming VUTD (Vertex-Unbalanced Triangle Detection):** $\forall \alpha \in (0, 1]$ the existence of a triangle in a tripartite graph with $|V_1| = n$ and $|V_2| = |V_3| = \Theta(n^\alpha)$ cannot be decided in time $O(n^{1+\alpha})$ [Bringmann, Carmeli 2022]

