Answering Conjunctive Queries under Updates

Christoph Berkholz, Jens Keppeler, Nicole Schweikardt

Humboldt-Universität zu Berlin

Abstract

Enumerating and counting answers to k-ary conjunctive queries under single-tuple updates.

- Upper bounds: CQ *q-hierarchical* $\Longrightarrow$ constant update time
- Lower bounds: CQ **not** q-hierarchical $\Longrightarrow$ $\Omega(n^{1-\varepsilon})$ update time, n = size of the active domain (under OV- and OMv-conjectures).

Dichotomy for counting CQs

Theorem Let φ be a CQ.

- If φ is q-hierarchical, then $|\varphi(D)|$ can be computed with *linear* preprocessing time and *constant* update time.
- Otherwise, assuming the OMv-conjecture and the OV-conjecture, there is no algorithm that computes $|\varphi(D)|$ with arbitrary preprocessing time and $O(n^{1-\varepsilon})$ update time.

Dichotomy for enumerating CQs

Theorem Let φ be a *self-join free* CQ.

- If φ is q-hierarchical, then $\varphi(D)$ can be enumerated with *constant* delay and *constant* update time after *linear* preprocessing.
- Otherwise, assuming the OMv-conjecture, there is no algorithm with arbitrary preprocessing time and $O(n^{1-\varepsilon})$ update time that enumerates $\varphi(D)$ with $O(n^{1-\varepsilon})$ delay.

Algorithmic conjectures

Online matrix-vector multiplication (OMv)

Input Boolean $n \times n$ matrix M and stream $v_1, \ldots, v_n$ of n-dimensional Boolean vectors.

Task After preprocessing M, compute Mv_i before v_{i+1} arrives.

OMv-conjecture (Henzinger et al. 2015) For every $\varepsilon > 0$, no algorithm solves OMv in time $O(n^{3-\varepsilon})$.

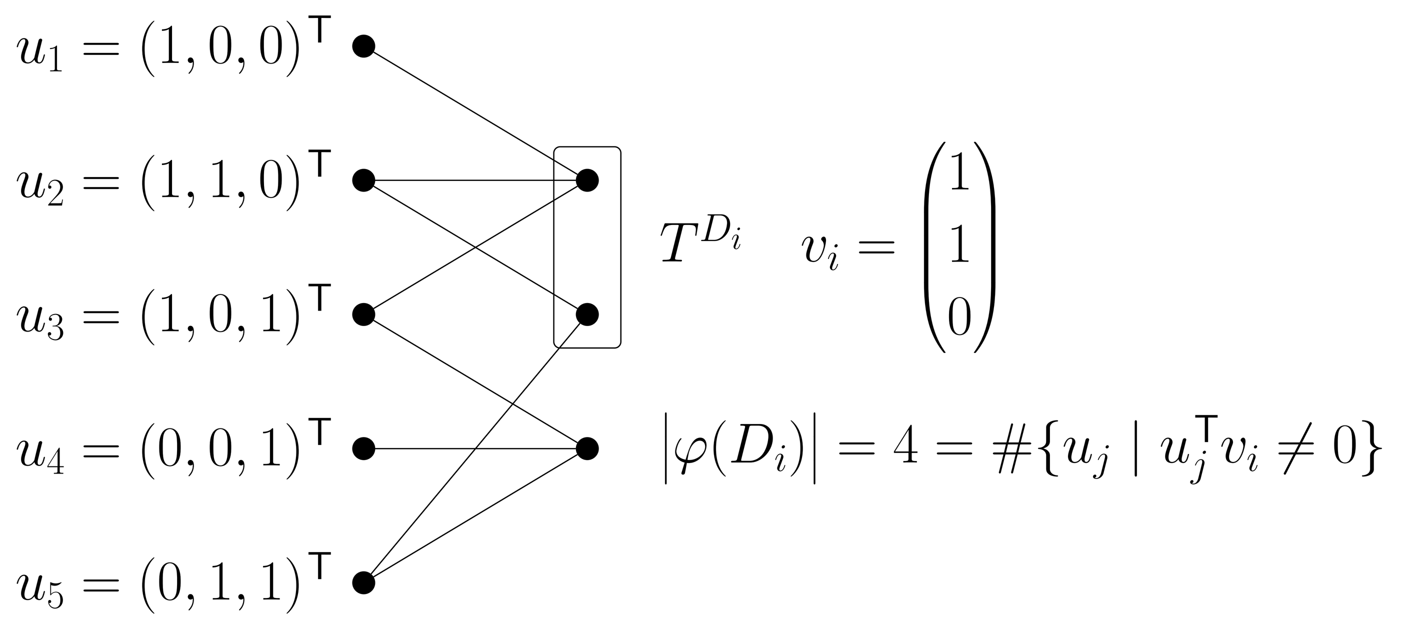
Orthogonal vectors (OV)

Input Two sets U and V of n Boolean vectors of dimension d.

Question Are there $u \in U$ and $v \in V$ such that $u^{\mathsf{T}} v = 0$?

OV-conjecture (Williams 2005) For every $\varepsilon > 0$, no algorithm solves OV for $d = \lceil \log^2 n \rceil$ in time $O(n^{2-\varepsilon})$.

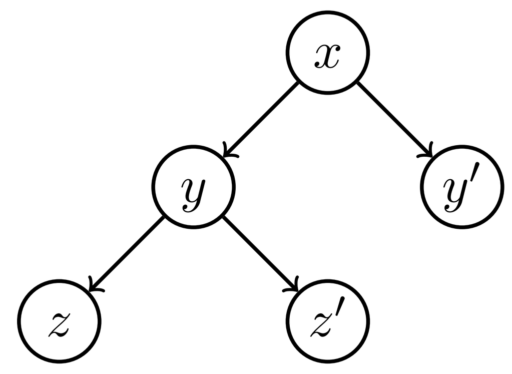
Example If non-q-hierarchical $\varphi(x) = \exists y\ E(x,y) \wedge T(y)$ can be counted in $O(n^{1-\varepsilon})$ update time, then the OV-conjecture fails.



- update T^{D_i} for every v_i and test against all $u_j \in U$ via $|\varphi(D_i)|$

q-hierarchical queries

$$\varphi(x,y,z,y',z') = \big(Rxyz \wedge Rxyz' \wedge Exy \wedge Exy' \wedge Sxyz \big)$$



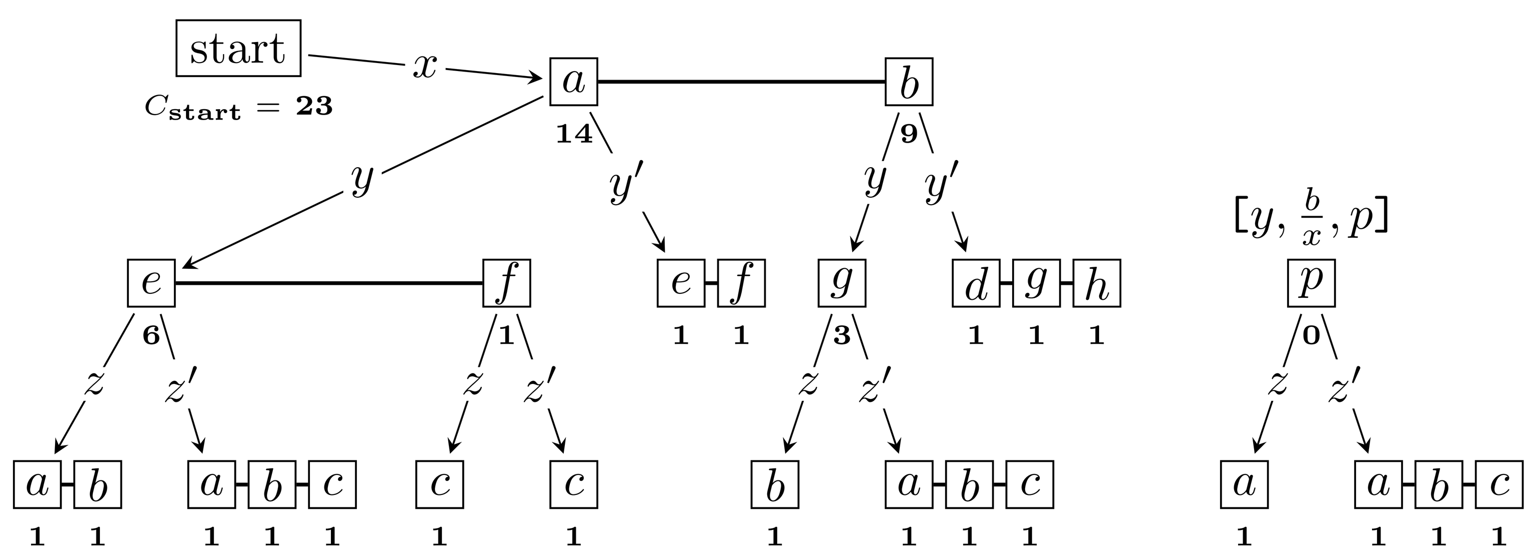
A conjunctive query is *q-hierarchical* if it has a q-tree in which

- variables of every atom form a path in this tree starting at the root,
- free variables form a connected subtree containing the root.

Data structure for q-hierarchical queries

$$\varphi(x,y,z,y',z') = \big(Rxyz \wedge Rxyz' \wedge Exy \wedge Exy' \wedge Sxyz \big)$$

$E^D = \{(a,e),\ (a,f),\ (b,d),\ (b,g),\ (b,h)\}$,
$S^D = \{(a,e,a),\ (a,e,b),\ (a,f,c),\ (b,g,b),\ (b,p,a)\}$,
$R^D = S^D \cup \{(a,e,c),\ (b,g,a),\ (b,g,c),\ (b,p,b),\ (b,p,c)\}$.



x	a	a	a	a	a	a	a	a	a	a	a	a	a	a	b	b	b	b	b	b	b	b	b
y	e	e	e	e	e	e	e	e	e	e	e	e	f	f	g	g	g	g	g	g	g	g	g
z	a	a	a	a	a	a	b	b	b	b	b	b	c	c	b	b	b	b	b	b	b	b	b
z'	a	a	b	b	c	c	a	a	b	b	c	c	c	c	a	a	a	b	b	b	c	c	c
y'	e	f	e	f	e	f	e	f	e	f	e	f	e	f	d	g	h	d	g	h	d	g	h

Data structure represents the query result (of size $\|D\|^k$) in space $O(\|D\|)$ and can be updated in constant time on single-tuple updates.

Data structure allows to:

- answer a Boolean query in constant time,
- compute the size $|\varphi(D)|$ of the query result in constant time,
- enumerate the query result with constant delay between the tuples,
- test for a given t, whether $t \in \varphi(D)$ in constant time,
- enumerate the change $\varphi(D_{\text{old}}) \,\triangle\, \varphi(D_{\text{new}})$ in the result with constant delay and compute its size in constant time.

Here, constant time (wrt. data complexity) means $O(\mathit{poly}(\varphi))$ and there are no large "hidden constants".

It turns out that evaluation of q-hierarchical queries is also efficient in practice. See: Idris, Ugarte, Vansummeren: "The dynamic Yannakakis Algorithm: Compact and Efficient Query Processing Under Updates" at SIGMOD'17.

Contact

Mail {berkholz, schweikn, keppelej}@informatik.hu-berlin.de

Url www.informatik.hu-berlin.de/logik

Full version https://arxiv.org/abs/1702.06370