An Open Question in Dynamic Descriptive Complexity

Thomas Schwentick

Venice, March 2017
Open Question

• Does Reachability $\in$ DynProp hold?
PRINCIPLES of DATA MANAGEMENT -- databasetheory.org

A website on the Theoretical Foundations of Data Management

Welcome to databasetheory.org, your friendly stop over for all things database-theoretical ;)

Aim of this site

This site is one of the many initiatives that are coming out of a Dagstuhl Perspectives Workshop on the Foundations of Data Management and is to collect

- resources, including
  - DB theory conference info (ICDT,PODS)
  - Gems of PODS page
  - Info on Ron Fagin event (held at PODS 2016)
  - Links to textbooks.
- events, listing upcoming conferences, workshops, etc.
- job announcements.

It also features a blog, currently hosted by Jan Van den Bussche, and surveys that appeared in the Sigmod Record Database Principles Column.

We want this site to be the place to be for anything related to the principles of data management and for this we need your help. For any suggestions and content to be added please send an email to info@databasetheory.org. If you are interested in blogging on principles of data management, please register for an account.

What is this site about?

The research area on the principles of data management has its roots in database theory in which, as the name suggests, theoretical aspects of database management systems are studied.
Dynamic Complexity: The Setting

- Auxiliary data can speed up query answering after change
- Dynamic Algorithms

- Rather than dynamic algorithms, we study weak query languages as update mechanism in a dynamic setting:
  - Relational calculus (FO-logic)

- Auxiliary data might enable FO-logic to maintain queries that it can not express

Definition
- $\text{DynFO} \overset{\text{def}}{=} \text{queries that can be maintained by first-order logic with auxiliary relations under deletion and insertion of edges}$
Example: Reachability under Insertion

- **Obvious idea:** store the transitive closure of the edge relation in a binary auxiliary relation $T$

- **Update rule:**
  on insert $(u, v)$ into $E$
  update $T(x, y)$ as $T(x, y) \lor (T(x, u) \land T(v, y))$
  - determines the pairs $(x, y)$ in $T$ after insertion of $(u, v)$ to $E$
Motivation & Goals

- **Why DynFO?**
  - captures essentially what can be maintained in a relational database (core SQL)
  - meaningful from a complexity theoretic point of view:
    - $\textbf{FO}(+ \times) \equiv \text{uniform } \textbf{AC}^0$
      - $\equiv$ circuit families of bounded depth and poly size
    - the most natural logic

- **General goal of our research:**
  - Understand the expressive power of DynFO
  - Which queries are in DynFO?
  - Which queries are not in DynFO?
Expressibility

- In recent years we learned a lot about what can be done in DynFO
  - Reachability is in DynFO
  - All MSO-queries on graphs of bounded tree-width are in DynFO
  - Undirected Reachability under FO-defined insertions is in DynFO
  - AC¹-computable queries under parameter-free definable changes are in DynFO
  - Bottom line: DynFO is much more powerful than we thought

- Expressibility results are nice (and maybe even useful)
  - But one could argue that they just constitute a collection of tricks
  - We do not understand the “real nature” of Dynamic Descriptive Complexity before we are able to prove inexpressibility results
Inexpressibility Results: A Sad State

- Easy observation: \( q \in \text{DynFO} \Rightarrow q \in \text{PTIME} \)
  - Just insert the tuples of \( \mathcal{D} \) into an empty database one by one, and compute all updates

- So far there are no other general lower bound results for \text{DynFO}

- Most existing lower bounds apply to
  - auxiliary relations of bounded arity or
  - restricted logics or
  - both...

- \( k \)-ary \text{DynFO}: Update programs with at most \( k \)-ary auxiliary relations (plus the query relation)

---

Theorem [Dong, Su 95/98]

- \text{REACHABILITY} \notin \text{unary DynFO}
- For every \( k > 0 \), \( k \)-ary \text{DynFO} is strictly weaker than \( (k + 1) \)-ary \text{DynFO} on structures of “growing” arity
Dynamic programs with quantifier-free formulas

- Hesse initiated the study of dynamic programs with quantifier-free update formulas \[\text{[Hesse 03]}\]

**Definition**

- **DynProp:**
  - Queries that can be maintained in \textit{DynFO} with quantifier-free formulas and aux \textit{relations}
- **DynQF:**
  - Queries that can be maintained in \textit{DynFO} with quantifier-free formulas and aux \textit{functions} (and relations)

**Theorem** \[\text{[Hesse 03]}\]

- **DET-REACH \in DynProp** (no quantifiers, aux relations)

**Theorem** \[\text{[Hesse 03]}\]

- **SYM-REACH \in unary DynQF** (no quantifiers, unary aux functions \& relations)

**Theorem** \[\text{[Gelade, Marquardt, TS 09/12]}\]

- **DynProp \nsubseteq DynQF**

\[\text{DynQF formulas can use “if-then-else”-terms}\]
### Inexpressibility Results for Quantifier-free Fragments

#### Theorem [Gelade, Marquardt, TS 09/12]
- Alternating Reachability $\not\in \text{DynProp}$

#### Theorem [Gelade, Marquardt, TS 09/12]
- $\text{FO} \not\subseteq \text{DynProp}$

#### Theorem [Zeume, TS 13]
- Unary, binary, and ternary $\text{DynProp}$ form a strict hierarchy on graphs

#### Theorem [Zeume 14]
- The arity hierarchy of $\text{DynProp}$ is strict on graphs under insertions
- Actually, it is also strict under insertions and deletions
  - Vortmeier, Zeume, unpublished

### Open Questions

- Does $\text{REACHABILITY} \in \text{DynProp}$ hold?
- Does $\text{REACHABILITY} \in \text{DynQF}$ hold?
References