Visiting Georg's Kingdoms

What are the Limits for Efficient Conjunctive Query Evaluation Under Constraints?

Pablo Barceló[1] Diego Figueira[2] Georg Gottlob[3] Andreas Pieris[4]

[1] Institute for Foundational Research on Data & DCC, University of Chile

[2] Labri, CNRS Bordeaux

[3] Department of Computer Science, University of Oxford

[4] LFCS, University of Edinburgh

Conjunctive Queries

The core of relational query languages

$R_1(x_1),\ldots,R_n(x_n) \rightarrow \mathrm{Ans}(z)$

In general CQ evaluation is NP-complete

and takes time $|D|^{O(|q|)}$

Acyclic CQs

A CQ is acyclic if it admits a join tree

Theorem:

Acyclic CQs can be evaluated in time $O(|D| \cdot |q|)$

[Yannakakis, VLDB 1981]

Generalized Hypertreewidth

Captures the "degree of acyclicity" of a CQ

Most CQs encountered in practice have low hypertreewidth

(nearly-acyclic)

HW(k) = CQs of generalized hypertreewidth at most k

(Acyclicity = HW(1))

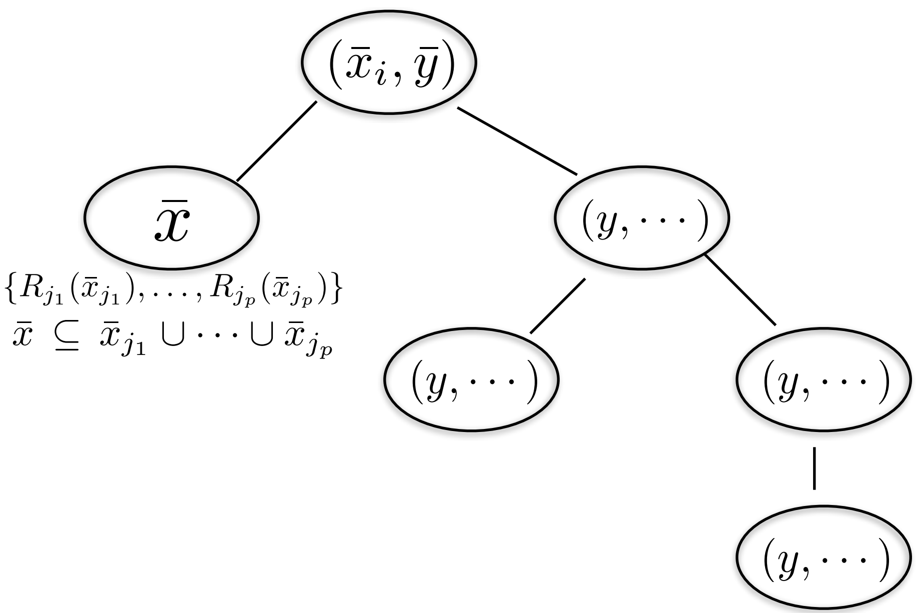
Theorem:

CQs in HW(k) can be evaluated in time $O(|D|^k \cdot |q|)$

[Gottlob, Leone & Scarcello, PODS 1999]

Generalized Hypertree Decompositions

$R_1(x_1),\ldots,R_n(x_n) \rightarrow \mathrm{Ans}(z)$



1. Each node is labeled with some variables from the CQ and a set of atoms that “cover” such variables
2. The variables of each atom in the CQ are contained in some node
3. Appearances of variables are connected

Its width is:
max number of atoms covering a node

The generalized hypertreewidth of a CQ is the minimum width of its generalized hypertree decompositions

Larger Islands of Tractability

A CQ is semantically in HW(k) iff it is equivalent to a CQ in HW(k)

("Semantic acyclicity" = semantically in HW(1))

Theorem:

Evaluation of CQs which are semantically in HW(k) is in PTIME

[Chen & Dalmau, CP 2005]

Assuming q is semantically in HW(k): $q(D)$ = true iff $q \rightarrow D$ iff $q \rightarrow_k D$

the duplicator has a winning strategy

for the so-called *existential k-cover game*,

which can be checked in time $O(|q|^{2k} \cdot |D|^{2k})$

Decidability of “Semantically in HW(k)”

Theorem:

A CQ is semantically in HW(k) iff its core is in HW(k)

[B., Romero & Vardi, PODS 2013]

Theorem:

Deciding if a CQ is semantically in HW(k) is NP-complete

[Dalmau, Kolaitis & Vardi, CP 2002]

“Semantically in HW(k)” Exhausts Tractability

(for fixed arity schemas)

Theorem:

Asume W[1] $\neq$ FTP.
Fix r and let ***C*** be a recursively enumerable class of CQs over schemas of maximum arity at most r. Then the following are equivalent:

- Evaluation for CQs in ***C*** is tractable
- Evaluation for CQs in ***C*** is fixed-parameter tractable — it can be solved in time p(|*D*|)•f(|*q*|), for p a polynomial and f a computable function
- There is k such that each CQ in ***C*** is equivalent to one in HW(k) — the cores in ***C*** are of bounded generalized hypertreewidth
- The cores in ***C*** are of bounded “treewidth”

[Grohe, FOCS 2003]

Tractable CQ Evaluation under Constraints

- Can we apply the constraints to reformulate a CQ as one in HW(k)?
- If so, how does this help query evaluation?
- Can we check if a CQ satisfies such conditions?

Constraints Enrich “Semantically in HW(k)”

$$E(x,y),\ E(y,z),\ E(z,x) \ \rightarrow\ Ans()$$

is not semantically acyclic

Under the assumption that the database satisfies the constraint

$$E(x,y),\ E(y,z) \ \rightarrow\ E(z,x)$$

it is equivalent to the acyclic query

$$E(x,y),\ E(y,x) \ \rightarrow\ Ans()$$

“Semantically in HW(k)” Under Constraints

Input: a CQ q, and a set of constraints Σ

Question: is there a CQ q' in HW(k) such that $q \equiv_{\Sigma} q'$

$q(D) = q'(D)$, for every database D that satisfies Σ

1. Does “semantically in HW(k)” under constraints helps evaluation?
2. When is the above problem decidable?
3. What is the complexity?

Database Constraints

Equality-generating Dependencies (egds):

$$\forall \bar{x} (\phi(\bar{x}) \rightarrow x_i = x_j)$$

Tuple-generating Dependencies (tgds):

$$\forall \bar{x} \forall \bar{y} (\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z} \psi(\bar{x}, \bar{z}))$$

Query Evaluation under egds

Theorem:

Evaluation of CQs semantically in HW(k) under egds is FPT
(over databases that satisfy the egds)

[B., Figueira, Gottlob & Pieris, unpublished]

Assuming q is semantically acyclic under Σ, for every D that satisfies Σ:

$$q(D) = \text{true} \quad \text{iff} \quad \text{chase}(q,\Sigma) \rightarrow_k D$$

it is of polynomial size,

can be computed in exponential time

Corollary:

Evaluation of CQs semantically in HW(k) under FDs is in PTIME

Decidability of “Semantically in HW(k)” under egds

Theorem:

Semantic acyclicity under egds is undecidable

[B., Figueira, Gottlob & Pieris, unpublished]

Theorem:

“Semantically in HW(k)” under unary FDs over unary and binary schemas is decidable in 2EXPTIME

[Figueira, LICS 2016]

For FDs the problem remains open

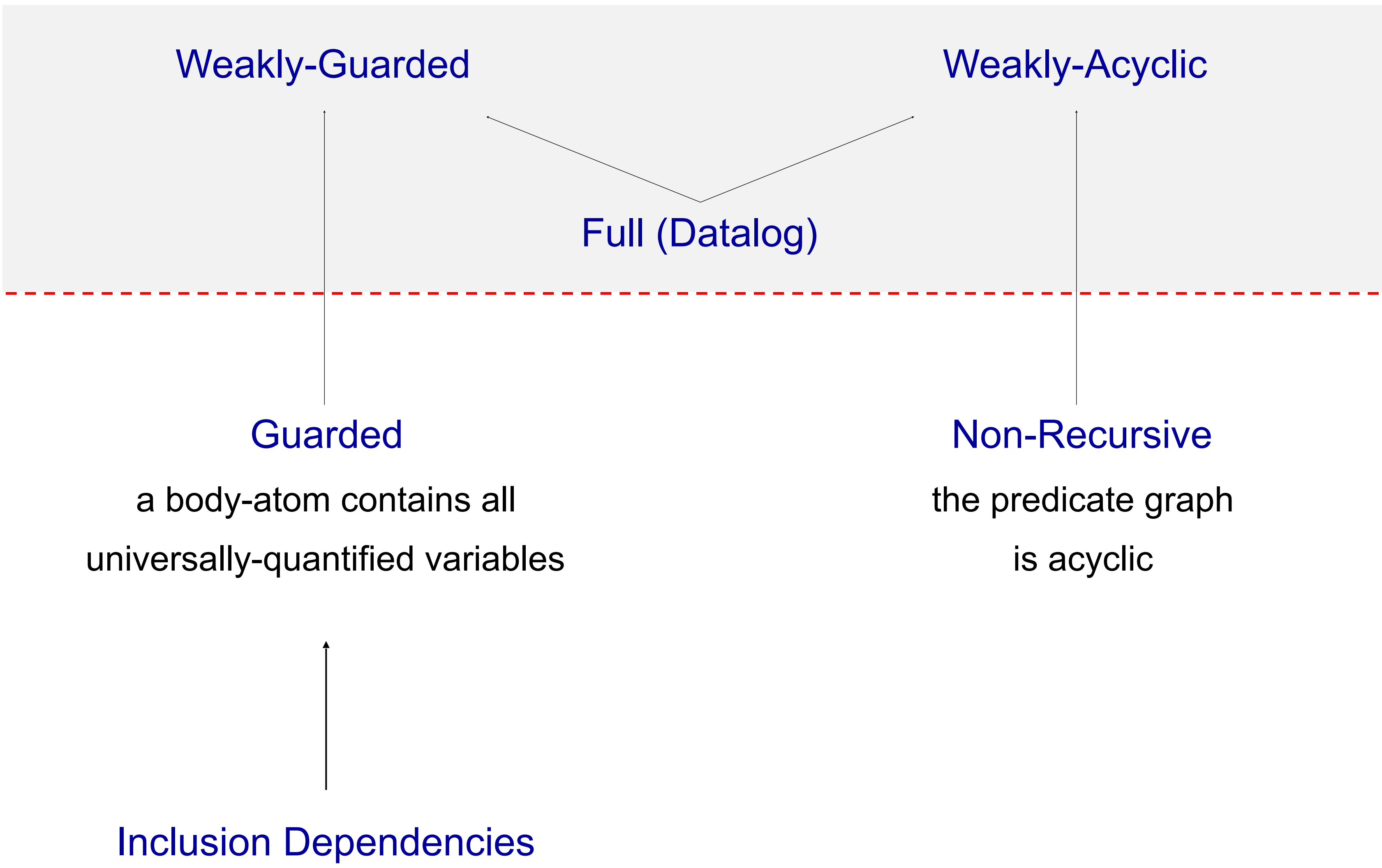
In Summary

- The notion “semantically in HW(k)” under egds defines an inaccessible island of efficiency for CQ evaluation (fixed-parameter tractability)
- For FDs it defines an island of tractability, which might also be inaccessible
- Tractability results based on pebble games correspond to a promise version of the problem: we hold the “promise” that the input is semantically in HW(k) under the set of egds/FDs
- “To the best of my knowledge the first time the “promise evaluation" approach may actually make practical sense” G. Gottlob

“Semantically in HW(k)” under tgds

To obtain positive results, we restrict to classes for which CQ containment is decidable

Classes of Tgds



Query Evaluation under guarded tgds

Theorem:

Evaluation of CQs semantically in HW(k) under guarded tgds is in PTIME (over databases that satisfy the tgds)

[B., Gottlob & Pieris, PODS 2016]

assuming q is semantically in HW(k) under Σ, for every D that satisfies Σ:

$$q(D) = \text{true} \quad \text{iff} \quad \text{chase}(q,\Sigma) \rightarrow_k D$$

$$\text{iff} \quad q \rightarrow_k D$$

can be checked in polynomial time

“Semantically in HW(k)” under guarded tgds

Theorem:

“Semantically in HW(k)” under guarded tgds is:
- 2EXPTIME-complete in general
- EXPTIME-complete for fixed arity
- NP-complete for fixed schema

[B., Gottlob & Pieris, PODS 2016]

Theorem:

“Semantically in HW(k)” under inclusion dependences is:
- PSPACE-complete in general
- NP-complete for fixed arity

[B., Gottlob & Pieris, PODS 2016]

…in fact, it behaves like CQ containment

[Calì, Gottlob & Kifer, KR 2008] for guarded tgds

[Johnson & Klug, PODS 1982] for inclusion dependencies

Guarded Tgds: Small Query Property

Proposition: Consider a set Σ of guarded tgds, and a CQ q.
If q is semantically in HW(k) under Σ, then there is a CQ q' in HW(k) such that $|q'| \leq O(k) \cdot |q|^2$ and $q \equiv_\Sigma q'$

[B., Gottlob & Pieris, PODS 2016]

Guess-and-check algorithm:

1. Guess a CQ q' in HW(k) of size at most $O(k) \cdot |q|^2$
2. Verify that $q \equiv_\Sigma q'$

Up to Now

Weakly-Guarded

Weakly-Acyclic

Full

✓ Guarded

a body-atom contains all the universally-quantified variables

Non-Recursive

the predicate graph is acyclic

✓ Inclusion Dependencies

Query Evaluation under Nonrecursive tgds

Theorem:

Evaluation of CQs semantically in HW(k) under non recursive sets of tgds is in FPT (over databases that satisfy the tgds)

[B., Gottlob & Pieris, PODS 2016]

Assuming q is semantically in HW(k) under Σ, for every D that satisfies Σ:

$q(D)$ = true iff chase(q,Σ) $\rightarrow_k$ D iff $q \rightarrow_k D$

❌

But for nonrecursive sets of tgds, chase(q,Σ) is of double-exponential size

chase(q,Σ) $\rightarrow_k$ D can be checked in time polynomial in D and double-exponential in q

“Semantically in HW(k)” under nonrecursive tgds

Theorem:

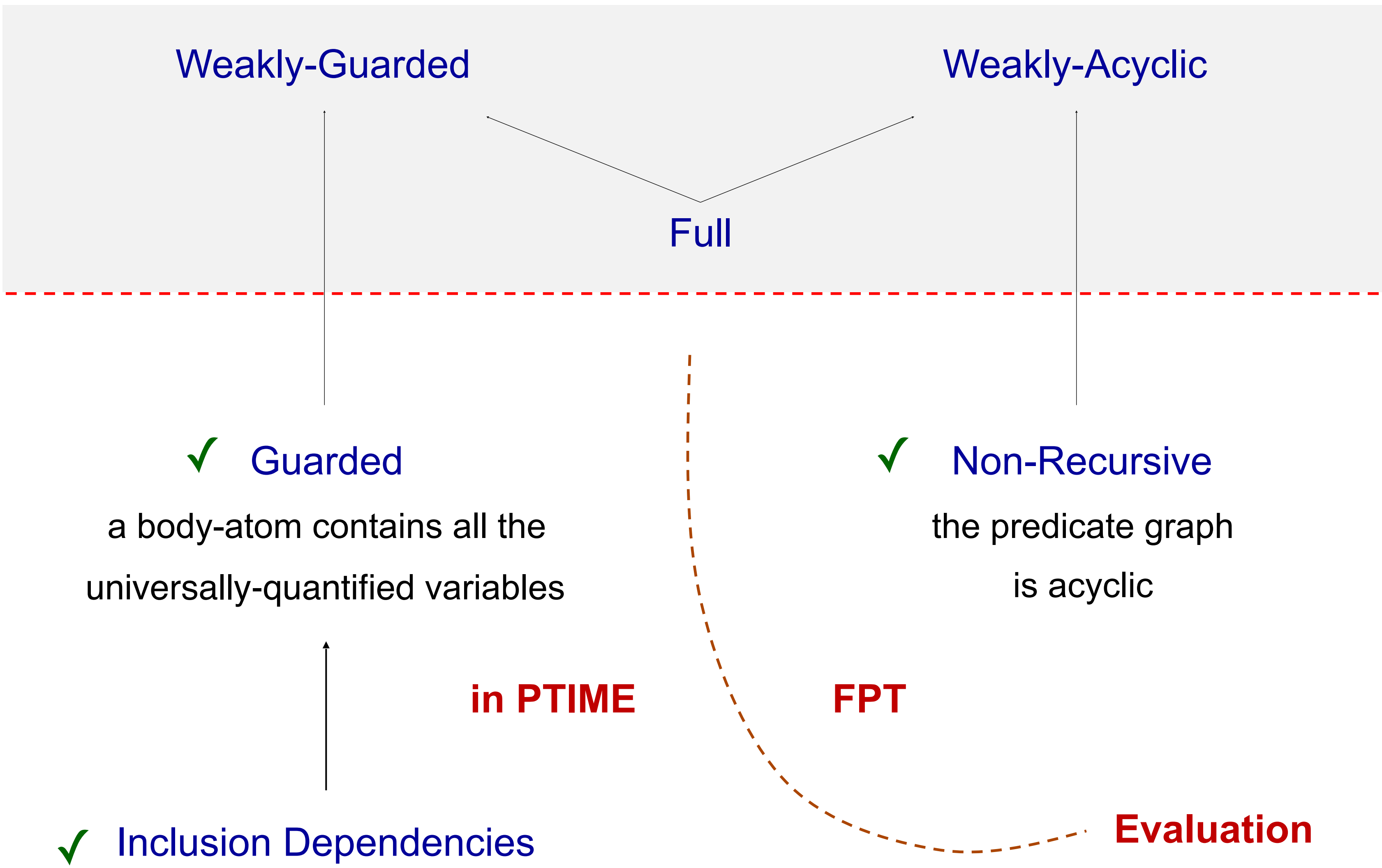
“Semantically in HW(k)” under non-recursive tgds is:

- NEXPTIME-complete, even for fixed arity
- NP-complete for fixed schema

[B., Gottlob & Pieris, PODS 2016]

…in fact, it behaves like CQ containment
[Lukasiewicz et al., AAAI 2015]

Up to Now



Weakly-Guarded
Weakly-Acyclic
Full
✓ Guarded
a body-atom contains all the
universally-quantified variables
✓ Non-Recursive
the predicate graph
is acyclic
in PTIME
FPT
Evaluation
✓ Inclusion Dependencies

The Case of Full Tgds

Theorem:

Evaluation of CQs semantically in HW(k) under full tgds is FPT (over databases that satisfy the tgds)

[B., Figueira, Gottlob & Pieris, unpublished]

Theorem:

Semantic acyclicity under full tgds (Datalog) is undecidable

[B., Gottlob & Pieris, PODS 2016]

Summary

	egds	FDs	guarded tgds	nonrecursive sets of tgds	full tgds
Evaluation	FPT	PTIME	PTIME	FPT	FPT
Identification	Undecidable	?	2EXP-comp	NEXP-comp	Undecidable

Further Advancements

- The previous results continue to hold for UCQs

- Our techniques yield CQ approximations in HW(k)

  - consider a CQ q that is not semantically in HW(k) under Σ

  - we obtain an acyclic CQ q' that is maximally contained in Q under Σ

What remains to be done?

- Delimit the limits of tractability for CQs under constraints à *la* Grohe (guarded and non recursive tgds, FDs, egds)?
- Develop a better understanding of the decidability of the notion of "semantically in HW(k)" for FDs
- Obtain positive results in the presence of both tgds and egds?
- Understand how to obtain maximum benefit of the semantic information contained in the data in order to speed-up CQ evaluation?