# Conjunctive Query Containment over Trees

Henrik Björklund Wim Martens Thomas Schwentick

University of Dortmund, Germany

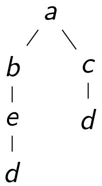
# Outline

# Outline

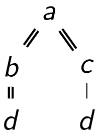
# What are Conjunctive Queries over Trees

## We know XPath

Tree:



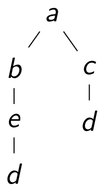

Pattern:



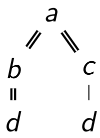
# What are Conjunctive Queries over Trees

## What are Conjunctive Queries

Tree:



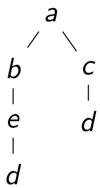

Pattern:



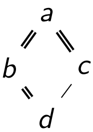
# What are Conjunctive Queries over Trees

## What are Conjunctive Queries

Tree:



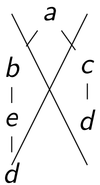

Pattern:



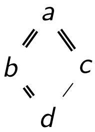
# What are Conjunctive Queries over Trees

## What are Conjunctive Queries

Tree:




Pattern:

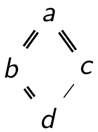


# What are Conjunctive Queries over Trees

## What are Conjunctive Queries

Tree:


Pattern:



# What are Conjunctive Queries over Trees

## A Conjunctive Query (CQ) is

a positive existential first-order formula without disjunction over

- unary predicates $a(x)$ (i.e., variable $x$ is labeled $a$)
- binary predicates
  - $Child(x, y)$;
  - $NextSibling(x, y)$;
  - $Following(x, y)$;

  and their transitive (and reflexive) closures

# What are Conjunctive Queries over Trees

## A Conjunctive Query (CQ) is

a positive existential first-order formula without disjunction over

- unary predicates $a(x)$ (i.e., variable $x$ is labeled $a$)
- binary predicates
  - $Child(x, y)$;
  - $NextSibling(x, y)$;
  - $Following(x, y)$;

  and their transitive (and reflexive) closures

## Example

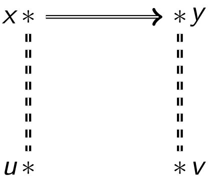
$\exists u, v, x, y. Child^*(x, u) \wedge NextSibling^+(x, y) \wedge Child^*(y, v)$

# What are Conjunctive Queries over Trees

## Example

$\exists u, v, x, y . Child^*(x, u) \wedge NextSibling^+(x, y) \wedge Child^*(y, v)$

## Graphical Query Representation

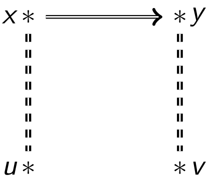


# What are Conjunctive Queries over Trees

## Example

$\exists u, v, x, y. \mathit{Child}^*(x, u) \wedge \mathit{NextSibling}^+(x, y) \wedge \mathit{Child}^*(y, v)$

## Graphical Query Representation




That is,

$\mathit{Following}(u, v) = \exists x, y. \mathit{Child}^*(x, u) \wedge \mathit{NextSibling}^+(x, y) \wedge \mathit{Child}^*(y, v)$

# Semantics of Conjunctive Queries

**We (mainly) consider Boolean satisfaction**

- tree $t$ models CQ $Q$ if

$$Q \text{ can be embedded into } t \text{ (denoted } t \models Q\text{)}$$

- The language $L(Q)$ of $Q$ is the set of trees modelling $Q$

# Our Problems of Interest

## We (mainly) consider Boolean satisfaction

- tree $t$ models CQ $Q$ if

$$Q \text{ can be embedded into } t \text{ (denoted } t \models Q\text{)}$$

- The language $L(Q)$ of $Q$ is the set of trees modelling $Q$

## Our Problems of Interest

- Containment: Given CQs $P$ and $Q$, is $L(P) \subseteq L(Q)$?
- Satisfiability: Given CQ $Q$, is $L(Q) \neq \emptyset$?
- Containment w.r.t. a DTD: Given CQs $P$ and $Q$, and a DTD $D$, is

$$L(D) \cap L(P) \subseteq L(D) \cap L(Q)?$$

# Why Conjunctive Queries over Trees

- They are a clean and simple query model
- They are closely related to XPath 2.0 (using path intersection)
- They are used in several contexts:
  - Web information extraction [Baumgartner et al. 2001, Gottlob and Koch 2004]
  - Computational linguistics
  - Dominance constraints [Marcus et al. 1983]

# Known Results

[Gottlob,Koch,Schulz JACM 2006] investigated

(combined) complexity of conjunctive queries over trees

| | $C$ | $C^+$ | $C^*$ | $NS$ | $NS^+$ | $NS^*$ | $F$ |
|---|---|---|---|---|---|---|---|
| $C$ | in **P** | **NP** | **NP** | in **P** | in **P** | in **P** | **NP** |
| $C^+$ | | in **P** | in **P** | **NP** | **NP** | **NP** | **NP** |
| $C^*$ | | | in **P** | **NP** | **NP** | **NP** | **NP** |
| $NS$ | | | | in **P** | in **P** | in **P** | **NP** |
| $NS^+$ | | | | | in **P** | in **P** | **NP** |
| $NS^*$ | | | | | | in **P** | **NP** |
| $F$ | | | | | | | in **P** |

**PTIME** fragments: CQ($C$,$NS$,$NS^+$,$NS^*$), CQ($C^+$,$C^*$), CQ($F$)

Together, this is a dichotomy for CQ($S$), where $S$ is a set of axes

# Outline

# Our Results (1)

Containment:

| | $C$ | $C^+$ | $C^*$ | $NS$ | $NS^+$ | $NS^*$ | $F$ |
|---|---|---|---|---|---|---|---|
| $C$ | in **P** | $\Pi_2^P$ | $\Pi_2^P$ | **coNP** | **coNP** | **coNP** | $\Pi_2^P$ |
| $C^+$ | | **coNP** | **coNP** | $\Pi_2^P$ | $\Pi_2^P$ | $\Pi_2^P$ | $\Pi_2^P$ |
| $C^*$ | | | **coNP** | $\Pi_2^P$ | $\Pi_2^P$ | $\Pi_2^P$ | $\Pi_2^P$ |
| $NS$ | | | | in **P** | **coNP** | **coNP** | $\Pi_2^P$ |
| $NS^+$ | | | | | **coNP** | **coNP** | $\Pi_2^P$ |
| $NS^*$ | | | | | | **coNP** | $\Pi_2^P$ |
| $F$ | | | | | | | **coNP** |

**coNP** fragments: CQ($C$,$NS$,$NS^+$,$NS^*$), CQ($C^+$,$C^*$), CQ($F$)

Together, this is a trichotomy for CQ($S$), where $S$ is a set of axes

# Our Results (2)

Satisfiability:

| | $C$ | $C^+$ | $C^*$ | $NS$ | $NS^+$ | $NS^*$ | $F$ |
|---|---|---|---|---|---|---|---|
| $C$ | in **P** | **NP** (*) | **NP** | in **P** | in **P** | in **P** | **NP** |
| $C^+$ | | in **P** | in **P** | ? | ? | ? | ? |
| $C^*$ | | | in **P** | ? | ? | ? | ? |
| $NS$ | | | | in **P** | **NP** | **NP** | **NP** |
| $NS^+$ | | | | | in **P** | in **P** | in **P** |
| $NS^*$ | | | | | | in **P** | in **P** |
| $F$ | | | | | | | in **P** |

(*) already obtained in [Hidders DBPL 2003]

# Our Results (3)

Containment w.r.t. a schema:

... already **EXPTIME** hard for *Child*-only queries, w.r.t. a DTD

# Boolean versus N-Ary Queries

So the results hold for Boolean queries. . .

### What about N-ary queries?

- Containment: if the fragment has a *Child*-axis

  then the results carry over to N-ary queries

- Satisfiability:

  all results carry over to N-ary queries

# Outline

# Outline

# Satisfiability of CQ($NS$,$NS^+$) is **NP**-hard

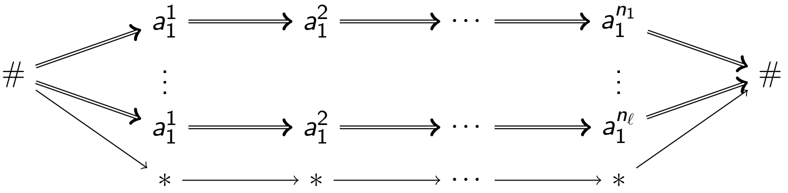
## Reduction from Shortest Common Supersequence:

Given

- a set of strings $S$ and
- an integer $k$,

is there a string $s$ of length at most $k$
that is a supersequence of every string in $S$?

## Gadget



# Similar proofs...

**Similar proofs can be used for**

- Satisfiability: **NP**-hardness for
  $\mathrm{CQ}(C,C^{+})$ $\mathrm{CQ}(NS,NS^{+})$ $\mathrm{CQ}(NS,F)$
  $\mathrm{CQ}(C,C^{*})$ $\mathrm{CQ}(NS,NS^{*})$ $\mathrm{CQ}(C,F)$ (extra trick needed)
- Containment: **coNP**-hardness for
  $\mathrm{CQ}(C^{+})$ $\mathrm{CQ}(NS^{+})$ $\mathrm{CQ}(F)$
  $\mathrm{CQ}(C^{*})$ $\mathrm{CQ}(NS^{*})$ $\mathrm{CQ}(C,NS)$

# Outline

# Small Model Property

## Small Model Property (SMP)

If $L(P) \not\subseteq L(Q)$ then there's a polynomial-size counterexample

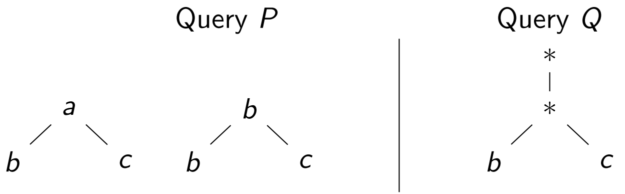
## Corollary

*For Containment, all the* **coNP** *and* $\Pi_2^P$ *upper bounds follow from SMP and [Gottlob et al. JACM 2006]*

# Outline

This sounds really obvious, but. . .

**Example**




Eventually:

case study, in which one case is a constraint satisfaction problem

# Outline

# The $\Pi_2^P$ Lower Bound proofs...

...are mostly harder

### Reductions from ∀∃ 1-in-3SAT:

Given a set $C_1, \ldots, C_m$ of triples from

$$\{x_1, \ldots, x_{n_x}\} \uplus \{y_1, \ldots, y_{n_y}\},$$

Does there exist,

- for every truth assignment for $\{x_1, \ldots, x_{n_x}\}$,
- a truth assignment for $\{y_1, \ldots, y_{n_y}\}$

such that each $C_i$ has exactly one true variable?

### Lemma

∀∃ *1-in-3SAT is* $\Pi_2^P$*-complete*

# Outline

# Final Remarks

- CQ Containment over Trees:
  - Gottlob-Koch-Schulz dichotomy $\rightarrow$ trichotomy
  - **PTIME** evaluation $\rightarrow$ **PTIME** or **coNP** containment
  - **NP** evaluation $\rightarrow$ $\Pi_2^P$ containment
- CQ Satisfiability over Trees:
  - Lower complexities than Containment (**PTIME** and **NP**)
  - Dichotomy changes (e.g. CQ($NS^+, F$))
- CQ Containment w.r.t. a Schema:
  - Probably more interesting in practice...
  - ...but complexity is higher! (already **EXPTIME** for CQ(Child))

# Final Remarks

- CQ Containment over Trees:
  - Gottlob-Koch-Schulz dichotomy $\rightarrow$ trichotomy
  - **PTIME** evaluation $\rightarrow$ **PTIME** or **coNP** containment
  - **NP** evaluation $\rightarrow$ $\Pi_2^P$ containment
- CQ Satisfiability over Trees:
  - Lower complexities than Containment (**PTIME** and **NP**)
  - Dichotomy changes (e.g. CQ($NS^+, F$))
- CQ Containment w.r.t. a Schema:
  - Probably more interesting in practice…
  - …but complexity is higher! (already **EXPTIME** for CQ(Child))

# Final Remarks

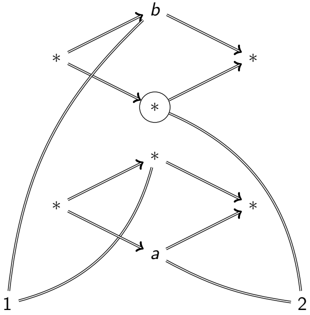
- CQ Containment over Trees:
  - Gottlob-Koch-Schulz dichotomy $\rightarrow$ trichotomy
  - **PTIME** evaluation $\rightarrow$ **PTIME** or **coNP** containment
  - **NP** evaluation $\rightarrow$ $\Pi_2^P$ containment
- CQ Satisfiability over Trees:
  - Lower complexities than Containment (**PTIME** and **NP**)
  - Dichotomy changes (e.g. CQ($NS^+, F$))
- CQ Containment w.r.t. a Schema:
  - Probably more interesting in practice. . .
  - . . . but complexity is higher! (already **EXPTIME** for CQ(Child))

# Backup Slides

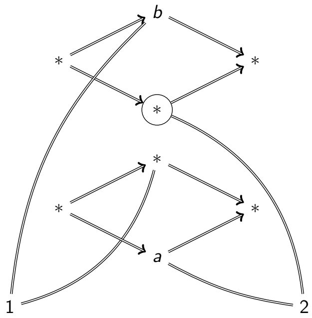
# What's so hard about the other SAT problems?

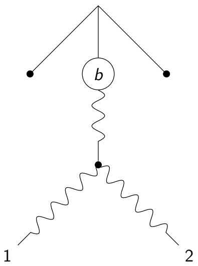
Our **PTIME** techniques don't work anymore...

# What's so hard about the other SAT problems?



# What's so hard about the other SAT problems?



# What's so hard about the other SAT problems?