Dichotomies in Ontology-Mediated Querying with the Guarded Fragment

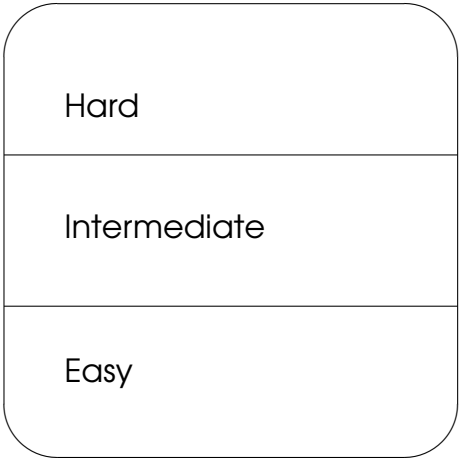
Frank Wolter

University of Liverpool

Based on joint work with A. Hernich, C. Lutz and F. Papacchini (PODS 2017)

Dichotomy Theorems

Given a class of problems, we would like to classify them into the hard and the easy problems. Ideally, there shouldn't be any intermediate problems.

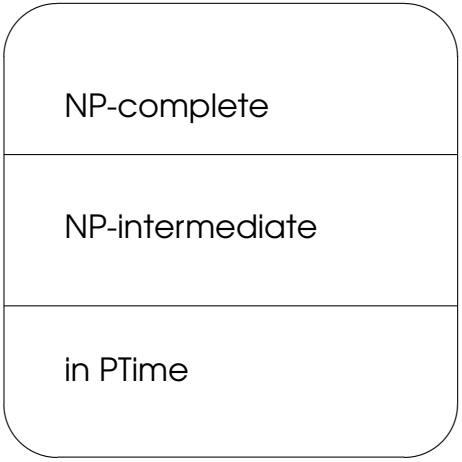


We focus on P/NP Dichotomy Theorems

By Ladner's Theorem, there are NP-intermediate problems (if P≠NP).

Moreover, being in P is undecidable for problems in NP (if P≠NP).

Thus, we can expect P/NP dichotomy theorems only for rather restricted classes of problems.



Homomorphism (or CSP) Problems

Consider an undirected graph H. How hard is the following problem:

- Input: an undirected graph G.
- Question: is there a homomorphism h from G to H?

($(h(a), h(b))$ is an edge in H if (a, b) is an edge in G.)

Homomorphism (or CSP) Problems

Consider an undirected graph H. How hard is the following problem:

- Input: an undirected graph G.
- Question: is there a homomorphism h from G to H?

($(h(a), h(b))$ is an edge in H if (a, b) is an edge in G.)

- if H is a single self-loop?

Homomorphism (or CSP) Problems

Consider an undirected graph H. How hard is the following problem:

- Input: an undirected graph G.
- Question: is there a homomorphism h from G to H?

($(h(a), h(b))$ is an edge in H if (a, b) is an edge in G.)

- if H is a single self-loop?
- if $H = K_2$ (K_2 complete graph on two vertices)?

Homomorphism (or CSP) Problems

Consider an undirected graph H. How hard is the following problem:

- Input: an undirected graph G.
- Question: is there a homomorphism h from G to H?

($(h(a), h(b))$ is an edge in H if (a, b) is an edge in G.)

- if H is a single self-loop?
- if $H = K_2$ (K_2 complete graph on two vertices)?
- if $H = K_3$?

Homomorphism (or CSP) Problems

Consider an undirected graph H. How hard is the following problem:

- Input: an undirected graph G.
- Question: is there a homomorphism h from G to H?

($(h(a), h(b))$ is an edge in H if (a, b) is an edge in G.)

- if H is a single self-loop?
- if $H = K_2$ (K_2 complete graph on two vertices)?
- if $H = K_3$?

Hell and Nesetril (1990): This problem is in PTime iff H contains a self-loop or is bipartite. Otherwise this problem is NP-complete.

Generalization to Relational Structures (CSP)

Let $\mathcal{H}$ be a finite relational structure (also called template). The constraint satisfaction problem for $\mathcal{H}$,

$$\mathrm{CSP}(\mathcal{H})$$

is the following decision problem:

- Input: a finite relational structure $\mathcal{D}$.
- Question: is there a homomorphism from $\mathcal{D}$ to $\mathcal{H}$?

Generalization to Relational Structures (CSP)

Let $\mathcal{H}$ be a finite relational structure (also called template). The constraint satisfaction problem for $\mathcal{H}$,

$$\mathrm{CSP}(\mathcal{H})$$

is the following decision problem:

- Input: a finite relational structure $\mathcal{D}$.
- Question: is there a homomorphism from $\mathcal{D}$ to $\mathcal{H}$?

Feder-Vardi Conjecture (1993): There is a P/NP dichotomy for CSPs. Equivalently, the is such a dichotomy for digraphs.

Generalization to Relational Structures (CSP)

Let $\mathcal{H}$ be a finite relational structure (also called template). The constraint satisfaction problem for $\mathcal{H}$,

$$\mathrm{CSP}(\mathcal{H})$$

is the following decision problem:

- Input: a finite relational structure $\mathcal{D}$.
- Question: is there a homomorphism from $\mathcal{D}$ to $\mathcal{H}$?

Feder-Vardi Conjecture (1993): There is a P/NP dichotomy for CSPs. Equivalently, the is such a dichotomy for digraphs.

Lots of progress over the past 20 years (mainly due to algebraic reformulation):

- Early result: There is a P/NP Dichotomy for CSPs with two elements (Schaefer 1978).
- Example: There is a P/NP Dichotomy for CSPs with three elements (Bulatov 2006).

Ontology Mediated Querying of Data (Example 1)

- **Data** $\mathcal{D}$: finite set of ground atoms (often regarded as finite relational structure); e.g.,

$$\mathbf{LiverpoolAcademic}(\mathbf{peter}), \quad \mathbf{HasLiverpoolId}(\mathbf{sue}, \mathbf{Liv123})$$

Ontology Mediated Querying of Data (Example 1)

- **Data** $\mathcal{D}$: finite set of ground atoms (often regarded as finite relational structure); e.g.,

$$\textbf{LiverpoolAcademic}(\textbf{peter}), \quad \textbf{HasLiverpoolId}(\textbf{sue}, \textbf{Liv123})$$

- **Ontology** $\mathcal{O}$: a finite set of FO-sentences; e.g.,

$$\forall x \ (\textbf{LiverpoolAcademic}(x) \rightarrow \exists y \ \textbf{HasLiverpoolId}(x, y))$$

Ontology Mediated Querying of Data (Example 1)

- **Data** $\mathcal{D}$: finite set of ground atoms (often regarded as finite relational structure); e.g.,

$$\textbf{LiverpoolAcademic}(\textbf{peter}), \quad \textbf{HasLiverpoolId}(\textbf{sue}, \textbf{Liv123})$$

- **Ontology** $\mathcal{O}$: a finite set of FO-sentences; e.g.,

$$\forall x \ (\textbf{LiverpoolAcademic}(x) \rightarrow \exists y \ \textbf{HasLiverpoolId}(x, y))$$

- **Query** $q(\vec{x})$: an FO-formula; e.g.,

$$q(x) = \exists y \ \textbf{HasLiverpoolId}(x, y)$$

Ontology Mediated Querying of Data (Example 1)

- **Data** $\mathcal{D}$: finite set of ground atoms (often regarded as finite relational structure); e.g.,

$$\textbf{LiverpoolAcademic}(\textbf{peter}), \quad \textbf{HasLiverpoolId}(\textbf{sue}, \textbf{Liv123})$$

- **Ontology** $\mathcal{O}$: a finite set of FO-sentences; e.g.,

$$\forall x \ (\textbf{LiverpoolAcademic}(x) \rightarrow \exists y \ \textbf{HasLiverpoolId}(x, y))$$

- **Query** $q(\vec{x})$: an FO-formula; e.g.,

$$q(x) = \exists y \ \textbf{HasLiverpoolId}(x, y)$$

A tuple $\vec{a} \in \textbf{dom}(\mathcal{D})$ is a **certain answer** for q and $\mathcal{O}$ over $\mathcal{D}$ if

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

Ontology Mediated Querying of Data (Example 1)

- **Data** $\mathcal{D}$: finite set of ground atoms (often regarded as finite relational structure); e.g.,

$$\textbf{LiverpoolAcademic}(\textbf{peter}), \quad \textbf{HasLiverpoolId}(\textbf{sue}, \textbf{Liv123})$$

- **Ontology** $\mathcal{O}$: a finite set of FO-sentences; e.g.,

$$\forall x \; (\textbf{LiverpoolAcademic}(x) \rightarrow \exists y \; \textbf{HasLiverpoolId}(x, y))$$

- **Query** $q(\vec{x})$: an FO-formula; e.g.,

$$q(x) = \exists y \; \textbf{HasLiverpoolId}(x, y)$$

A tuple $\vec{a} \in \textbf{dom}(\mathcal{D})$ is a **certain answer** for q and $\mathcal{O}$ over $\mathcal{D}$ if

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

Here

$$\mathcal{D} \cup \mathcal{O} \models q(a) \quad \Leftrightarrow \quad a \in \{\textbf{sue}, \textbf{peter}\}$$

Ontology Mediated Querying of Data (Reachability)

- **Ontology** $\mathcal{O}$:

$$\{\forall x\ (\exists y\ (\mathbf{H}(y) \wedge \mathbf{parent}(x, y)) \rightarrow \mathbf{H}(x))\}$$

Ontology Mediated Querying of Data (Reachability)

- **Ontology** $\mathcal{O}$:

$$\{\forall x\ (\exists y\ (\mathsf{H}(y) \wedge \mathsf{parent}(x, y)) \rightarrow \mathsf{H}(x))\}$$

- **Query** q:

$$q(x) = \mathsf{H}(x)$$

Ontology Mediated Querying of Data (Reachability)

- **Ontology** $\mathcal{O}$:

$$\{\forall x\ (\exists y\ (\mathsf{H}(y) \wedge \mathsf{parent}(x, y)) \rightarrow \mathsf{H}(x))\}$$

- **Query** q:

$$q(x) = \mathsf{H}(x)$$

- **Data** $\mathcal{D}$:

$$\mathsf{parent}(b_0, b_1), \cdots, \mathsf{parent}(b_5, b_6), \quad \mathsf{H}(b_6)$$

Ontology Mediated Querying of Data (Reachability)

- **Ontology** $\mathcal{O}$:

$$\{\forall x\ (\exists y\ (\mathsf{H}(y) \wedge \mathsf{parent}(x, y)) \rightarrow \mathsf{H}(x))\}$$

- **Query** q:

$$q(x) = \mathsf{H}(x)$$

- **Data** $\mathcal{D}$:

$$\mathsf{parent}(b_0, b_1), \cdots, \mathsf{parent}(b_5, b_6), \quad \mathsf{H}(b_6)$$

- **Certain answers** for $q(x)$ and $\mathcal{O}$ over $\mathcal{D}$ are:

$$\mathcal{D} \cup \mathcal{O} \models q(a) \quad \Leftrightarrow \quad a \in \{b_0, b_1, b_2, b_3, b_4, b_5, b_6\}.$$

Ontology Mediated Querying of Data (Colorability)

- **Ontology** $\mathcal{O}$:
  - $\forall x\ (\mathbf{red}(x) \vee \mathbf{blue}(x) \vee \mathbf{green}(x))$.
  - $\forall x\ (\mathbf{red}(x) \wedge E(x,y) \wedge \mathbf{red}(y) \rightarrow \mathbf{clash}(x))$ (same for **blue**, **green**).

Ontology Mediated Querying of Data (Colorability)

- **Ontology** $\mathcal{O}$:
  - $\forall x\ (\mathbf{red}(x) \vee \mathbf{blue}(x) \vee \mathbf{green}(x))$.
  - $\forall x\ (\mathbf{red}(x) \wedge E(x,y) \wedge \mathbf{red}(y) \rightarrow \mathbf{clash}(x))$ (same for **blue**, **green**).
- **Query** q:

$$q() = \exists x\ \mathbf{clash}(x)$$

Ontology Mediated Querying of Data (Colorability)

- **Ontology** $\mathcal{O}$:
  - $\forall x\ (\textbf{red}(x) \vee \textbf{blue}(x) \vee \textbf{green}(x))$.
  - $\forall x\ (\textbf{red}(x) \wedge E(x,y) \wedge \textbf{red}(y) \rightarrow \textbf{clash}(x))$ (same for **blue**, **green**).
- **Query** q:

$$q() = \exists x\ \textbf{clash}(x)$$

- **Data** $\mathcal{D}$: undirected graph

$$\mathcal{D} = (W, E)$$

Ontology Mediated Querying of Data (Colorability)

- **Ontology** $\mathcal{O}$:
  - $\forall x\ (\mathbf{red}(x) \vee \mathbf{blue}(x) \vee \mathbf{green}(x))$.
  - $\forall x\ (\mathbf{red}(x) \wedge E(x,y) \wedge \mathbf{red}(y) \rightarrow \mathbf{clash}(x))$ (same for **blue**, **green**).
- **Query** q:

$$q() = \exists x\ \mathbf{clash}(x)$$

- **Data** $\mathcal{D}$: undirected graph

$$\mathcal{D} = (W, E)$$

- **Certain Answers** to q and $\mathcal{O}$ over $\mathcal{D}$:

$$\mathcal{O} \cup \mathcal{D} \models q \quad \text{iff} \quad \mathcal{D} \text{ is not 3-colorable}$$

Ontology Mediated Querying of Data (Colorability)

- **Ontology** $\mathcal{O}$:
  - $\forall x\ (\mathbf{red}(x) \vee \mathbf{blue}(x) \vee \mathbf{green}(x))$.
  - $\forall x\ (\mathbf{red}(x) \wedge E(x,y) \wedge \mathbf{red}(y) \rightarrow \mathbf{clash}(x))$ (same for **blue**, **green**).
- **Query** q:

$$q() = \exists x\ \mathbf{clash}(x)$$

- **Data** $\mathcal{D}$: undirected graph

$$\mathcal{D} = (W, E)$$

- **Certain Answers** to q and $\mathcal{O}$ over $\mathcal{D}$:

$$\mathcal{O} \cup \mathcal{D} \models q \quad \text{iff} \quad \mathcal{D} \text{ is not 3-colorable}$$

- One can do this for every CSP($\mathcal{H}$).

Relevant Languages for Ontology-Mediated Querying

Lots of results over the past 15 years on the complexity of deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a}),$$

Relevant Languages for Ontology-Mediated Querying

Lots of results over the past 15 years on the complexity of deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a}),$$

where q typically a **conjunctive query** (or **primitive positive sentence**), that is an FO-sentence of the form:

$$\exists \vec{x} \bigwedge_{i \in I} R_i(\vec{x_i})$$

Relevant Languages for Ontology-Mediated Querying

Lots of results over the past 15 years on the complexity of deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a}),$$

where q typically a **conjunctive query** (or **primitive positive sentence**), that is an FO-sentence of the form:

$$\exists \vec{x} \bigwedge_{i \in I} R_i(\vec{x}_i)$$

$\mathcal{O}$ often in a fragment of the **guarded fragment (GF)** of FO only admit guarded quantifiers

$$\forall \vec{y}(\alpha(\vec{x}, \vec{y}) \rightarrow \varphi(\vec{x}, \vec{y})), \quad \exists \vec{y}(\alpha(\vec{x}, \vec{y}) \wedge \varphi(\vec{x}, \vec{y}))$$

where $\varphi(\vec{x}, \vec{y})$ is in GF and $\alpha(\vec{x}, \vec{y})$ is an atomic formula containing all variables in $\vec{x} \cup \vec{y}$.

Relevant Languages for Ontology-Mediated Querying

Lots of results over the past 15 years on the complexity of deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a}),$$

where q typically a **conjunctive query** (or **primitive positive sentence**), that is an FO-sentence of the form:

$$\exists \vec{x} \bigwedge_{i \in I} R_i(\vec{x}_i)$$

$\mathcal{O}$ often in a fragment of the **guarded fragment (GF)** of FO only admit guarded quantifiers

$$\forall \vec{y}(\alpha(\vec{x}, \vec{y}) \rightarrow \varphi(\vec{x}, \vec{y})), \quad \exists \vec{y}(\alpha(\vec{x}, \vec{y}) \wedge \varphi(\vec{x}, \vec{y}))$$

where $\varphi(\vec{x}, \vec{y})$ is in GF and $\alpha(\vec{x}, \vec{y})$ is an atomic formula containing all variables in $\vec{x} \cup \vec{y}$.

GF inherits many nice properties from modal and description logics.

Relevant Languages for Ontology-Mediated Querying

Lots of results over the past 15 years on the complexity of deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a}),$$

where q typically a **conjunctive query** (or **primitive positive sentence**), that is an FO-sentence of the form:

$$\exists \vec{x} \bigwedge_{i \in I} R_i(\vec{x}_i)$$

$\mathcal{O}$ often in a fragment of the **guarded fragment (GF)** of FO only admit guarded quantifiers

$$\forall \vec{y}(\alpha(\vec{x}, \vec{y}) \to \varphi(\vec{x}, \vec{y})), \quad \exists \vec{y}(\alpha(\vec{x}, \vec{y}) \wedge \varphi(\vec{x}, \vec{y}))$$

where $\varphi(\vec{x}, \vec{y})$ is in GF and $\alpha(\vec{x}, \vec{y})$ is an atomic formula containing all variables in $\vec{x} \cup \vec{y}$.

GF inherits many nice properties from modal and description logics.

We also consider the **2-variable guarded fragment of FO with counting (GC_2)**:

$$\forall x(x = x \to (\exists^{\geq 200} y\ \textbf{author\_of}(x, y) \to \textbf{ProlificAuthor}(x)))$$

Some Complexity Results

Deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

for q a conjunctive query and

- $\mathcal{O}$ empty: NP-complete (homomorphism problem).

Some Complexity Results

Deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

for q a conjunctive query and

- $\mathcal{O}$ empty: NP-complete (homomorphism problem).
- $\mathcal{O}$ in GF or GC_2: 2ExpTime-complete (Baranyi et al 2010).

Some Complexity Results

Deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

for q a conjunctive query and

- $\mathcal{O}$ empty: NP-complete (homomorphism problem).
- $\mathcal{O}$ in GF or GC_2: 2ExpTime-complete (Baranyi et al 2010).
- $\mathcal{O}$ in 2-variable fragment of FO: undecidable (Rosati 2007).

Some Complexity Results

Deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

for q a conjunctive query and

- $\mathcal{O}$ empty: NP-complete (homomorphism problem).
- $\mathcal{O}$ in GF or GC_2: 2ExpTime-complete (Baranyi et al 2010).
- $\mathcal{O}$ in 2-variable fragment of FO: undecidable (Rosati 2007).

Slightly misleading!

Data Complexity

When deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

assume that $\mathcal{O}$ and q are **small** and $\mathcal{D}$ is **large**.

Data Complexity

When deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

assume that $\mathcal{O}$ and q are **small** and $\mathcal{D}$ is **large**.

Then it is reasonable to assume that

- $\mathcal{O}$ and q are fixed and $\mathcal{D}$ is the only input; thus focus on **data complexity**.

Data Complexity

When deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

assume that $\mathcal{O}$ and q are **small** and $\mathcal{D}$ is **large**.

Then it is reasonable to assume that

- $\mathcal{O}$ and q are fixed and $\mathcal{D}$ is the only input; thus focus on **data complexity**.

We obtain:

- $\mathcal{O}$ empty: AC_0.

Data Complexity

When deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

assume that $\mathcal{O}$ and q are **small** and $\mathcal{D}$ is **large**.

Then it is reasonable to assume that

- $\mathcal{O}$ and q are fixed and $\mathcal{D}$ is the only input; thus focus on **data complexity**.

We obtain:

- $\mathcal{O}$ empty: AC_0.
- $\mathcal{O}$ in GF or GC_2: coNP-complete (Baranyi et al 2010).

Data Complexity

When deciding

$$\mathcal{D} \cup \mathcal{O} \models q(\vec{a})$$

assume that $\mathcal{O}$ and q are **small** and $\mathcal{D}$ is **large**.

Then it is reasonable to assume that

- $\mathcal{O}$ and q are fixed and $\mathcal{D}$ is the only input; thus focus on **data complexity**.

We obtain:

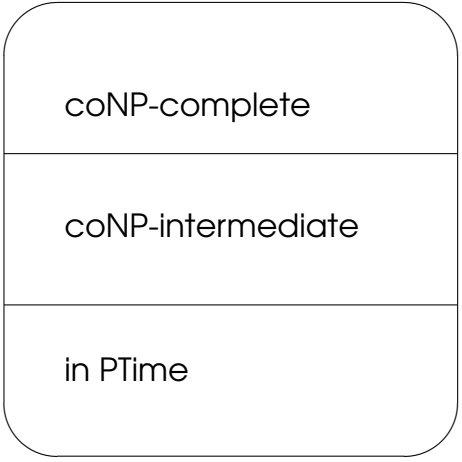
- $\mathcal{O}$ empty: AC_0.
- $\mathcal{O}$ in GF or GC_2: coNP-complete (Baranyi et al 2010).

For data complexity coNP-hardness is very bad news! It has become a huge industry to determine fragments of GF and GC_2 in PTime (or even better FO or datalog rewritable fragments)

.

P/coNP Dichotomy Theorems for Ontology Mediated Querying in GF and GC_2

- $\mathcal{O}$ **is in PTime** if for every conjunctive query q deciding $\mathcal{O} \cup \mathcal{D} \models q$ is in PTime in data complexity.
- $\mathcal{O}$ **is coNP-hard** if there exists a conjunctive query q such that deciding $\mathcal{O} \cup \mathcal{D} \models q$ is coNP-hard in data complexity.



Relevant Fragments

We consider the fragments uGF and uGC_2 of GF and GC_2 which are **invariant under disjoint unions**.

Relevant Fragments

We consider the fragments uGF and uGC_2 of GF and GC_2 which are **invariant under disjoint unions**.

Modulo logical equivalence uGF and uGC_2 sentences take the form

$$\forall x(x = x \to \varphi(x)), \quad \forall \vec{x}(R(\vec{x}) \to \varphi(\vec{x}))$$

where φ contains no closed subformulas and does not use equality as a guard.

Relevant Fragments

We consider the fragments uGF and uGC_2 of GF and GC_2 which are **invariant under disjoint unions**.

Modulo logical equivalence uGF and uGC_2 sentences take the form

$$\forall x(x = x \to \varphi(x)), \quad \forall \vec{x}(R(\vec{x}) \to \varphi(\vec{x}))$$

where φ contains no closed subformulas and does not use equality as a guard.

The **depth** of a uGF or uGC_2 sentence is the number of nestings of guarded quantifiers without the outermost universal guarded quantifier. The following sentence has depth 1:

$$\forall x(x = x \to \forall y(\mathbf{author\_of}(x, y) \to \mathbf{Book}(y)))$$

Relevant Fragments

We consider the fragments uGF and uGC_2 of GF and GC_2 which are **invariant under disjoint unions**.

Modulo logical equivalence uGF and uGC_2 sentences take the form

$$\forall x(x = x \to \varphi(x)), \quad \forall \vec{x}(R(\vec{x}) \to \varphi(\vec{x}))$$

where φ contains no closed subformulas and does not use equality as a guard.

The **depth** of a uGF or uGC_2 sentence is the number of nestings of guarded quantifiers without the outermost universal guarded quantifier. The following sentence has depth 1:

$$\forall x(x = x \to \forall y(\mathbf{author\_of}(x, y) \to \mathbf{Book}(y)))$$

In **uGF^- and uGC_2^-** we admit only $x = x$ as the outermost guard.

Relevant Fragments

We consider the fragments uGF and uGC_2 of GF and GC_2 which are **invariant under disjoint unions**.

Modulo logical equivalence uGF and uGC_2 sentences take the form

$$\forall x(x = x \to \varphi(x)), \quad \forall \vec{x}(R(\vec{x}) \to \varphi(\vec{x}))$$

where φ contains no closed subformulas and does not use equality as a guard.

The **depth** of a uGF or uGC_2 sentence is the number of nestings of guarded quantifiers without the outermost universal guarded quantifier. The following sentence has depth 1:

$$\forall x(x = x \to \forall y(\mathbf{author\_of}(x, y) \to \mathbf{Book}(y)))$$

In **uGF^- and uGC_2^-** we admit only $x = x$ as the outermost guard.

385 out of 411 ontologies in the **Bioportal** repository are in GC_2^-(depth 1)

Illustration (all in $\mathsf{GC}_2^-(1)$)

- The ontology

$$\mathcal{O}_1 = \{\forall x(\exists^{\geq 200} y\ \textbf{author\_of}(x, y) \rightarrow \textbf{ProlificWriter}(x))\}$$

  is in PTime.

Illustration (all in $\mathsf{GC}_2^-(1)$)

- The ontology

$$\mathcal{O}_1 = \{\forall x(\exists^{\geq 200} y\ \textbf{author\_of}(x, y) \rightarrow \textbf{ProlificWriter}(x))\}$$

  is in PTime.

- The ontology

$$\mathcal{O}_2 = \{\forall x(\textbf{Writer}(x) \rightarrow \exists y(\textbf{author\_of}(x, y) \wedge \textbf{Book}(y)))\}$$

  is in PTime.

Illustration (all in $\mathsf{GC}_2^-(1)$)

- The ontology

$$\mathcal{O}_1 = \{\forall x(\exists^{\geq 200} y\ \textbf{author\_of}(x, y) \rightarrow \textbf{ProlificWriter}(x))\}$$

  is in PTime.

- The ontology

$$\mathcal{O}_2 = \{\forall x(\textbf{Writer}(x) \rightarrow \exists y(\textbf{author\_of}(x, y) \wedge \textbf{Book}(y)))\}$$

  is in PTime.

- The ontology $\mathcal{O}_1 \cup \mathcal{O}_2$ is coNP-hard.

Illustration (all in $\mathsf{GC}_2^-(1)$)

- The ontology

$$\mathcal{O}_1 = \{\forall x(\exists^{\geq 200} y\ \textbf{author\_of}(x, y) \rightarrow \textbf{ProlificWriter}(x))\}$$

  is in PTime.

- The ontology

$$\mathcal{O}_2 = \{\forall x(\textbf{Writer}(x) \rightarrow \exists y(\textbf{author\_of}(x, y) \wedge \textbf{Book}(y)))\}$$

  is in PTime.

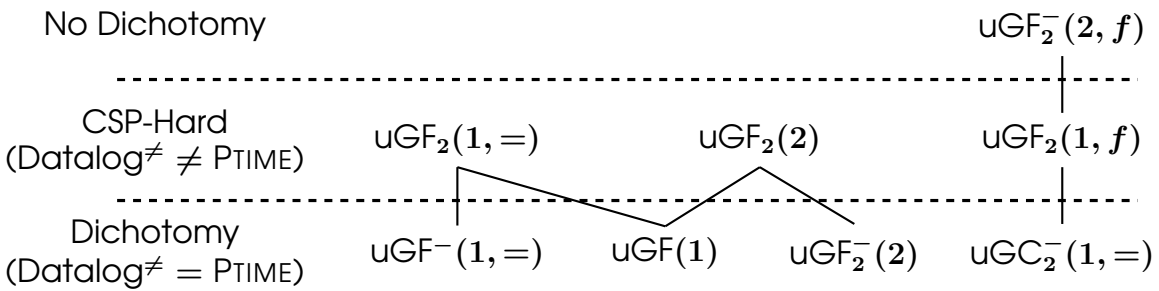
- The ontology $\mathcal{O}_1 \cup \mathcal{O}_2$ is coNP-hard.

- The ontology

$$\mathcal{O}_1 \cup \mathcal{O}_2 \cup \{\forall xy(\textbf{author\_of}(x, y) \rightarrow \textbf{Book}(y))\}$$

  is again in PTime.

Summary of Results



Number in brackets indicates

- depth,
- f presence of partial functions,
- $\cdot_2$ restriction to two variables,
- $\cdot^-$ restricts outermost guards to be equality.

Necessary condition for PTime: Materializability

An ontology $\mathcal{O}$ is **materializable** if for every $\mathcal{D}$ there exists a model $\mathcal{A}$ of $\mathcal{O} \cup \mathcal{D}$ such that for all conjunctive queries q:

$$\mathcal{O} \cup \mathcal{D} \models q \quad \Leftrightarrow \quad \mathcal{A} \models q$$

Necessary condition for PTime: Materializability

An ontology $\mathcal{O}$ is **materializable** if for every $\mathcal{D}$ there exists a model $\mathcal{A}$ of $\mathcal{O} \cup \mathcal{D}$ such that for all conjunctive queries q:

$$\mathcal{O} \cup \mathcal{D} \models q \quad \Leftrightarrow \quad \mathcal{A} \models q$$

Let $\mathcal{O}$ be an FO ontology invariant under disjoint unions. If $\mathcal{O}$ is not materializable, then $\mathcal{O}$ is coNP-hard.

Materializability is not a sufficient for PTime

We construct a materializable ontology from the ontology $\mathcal{O}$ encoding three-colorability:

- $\forall x\ (\mathbf{red}(x) \vee \mathbf{blue}(x) \vee \mathbf{green}(x))$.
- $\forall x\ (\mathbf{red}(x) \wedge \boldsymbol{E}(x, y) \wedge \mathbf{red}(y) \rightarrow \mathbf{clash}(x))$ (same for **blue, green**).

Clearly $\mathcal{O}$ itself is not materializable.

Replace $\mathbf{red}(x)$, $\mathbf{blue}(x)$, and $\mathbf{green}(x)$ by complex formulas that are not directly visible to conjunctive queries, e.g.

$$\exists y(\boldsymbol{R}_{\mathbf{red}}(x, y) \wedge \forall z(\boldsymbol{S}_{\mathbf{red}}(y, z) \rightarrow \mathbf{red}(z)))$$

Then the resulting ontology is still coNP-hard but materializable.

A sufficient condition for PTime (even datalog-rewritability)

Every relational structure $\mathcal{D}$ can be unravelled into a

guarded tree-decomposable

structure $\mathcal{D}^*$ (sometimes also called **acyclic**).

A sufficient condition for PTime (even datalog-rewritability)

Every relational structure $\mathcal{D}$ can be unravelled into a

guarded tree-decomposable

structure $\mathcal{D}^*$ (sometimes also called **acyclic**).

This unravelling preserves formulas in GF and GC_2.

A sufficient condition for PTime (even datalog-rewritability)

Every relational structure $\mathcal{D}$ can be unravelled into a

guarded tree-decomposable

structure $\mathcal{D}^*$ (sometimes also called **acyclic**).

This unravelling preserves formulas in GF and GC_2.

An ontology $\mathcal{O}$ is **unravelling tolerant** if for every $\mathcal{D}$ the following holds for the unravelling $\mathcal{D}^*$ of $\mathcal{D}$: for all acyclic conjunctive queries q:

$$\mathcal{O} \cup \mathcal{D} \models q \quad \Leftrightarrow \quad \mathcal{O} \cup \mathcal{D}^* \models q$$

A sufficient condition for PTime (even datalog-rewritability)

Every relational structure $\mathcal{D}$ can be unravelled into a

guarded tree-decomposable

structure $\mathcal{D}^*$ (sometimes also called **acyclic**).

This unravelling preserves formulas in GF and GC_2.

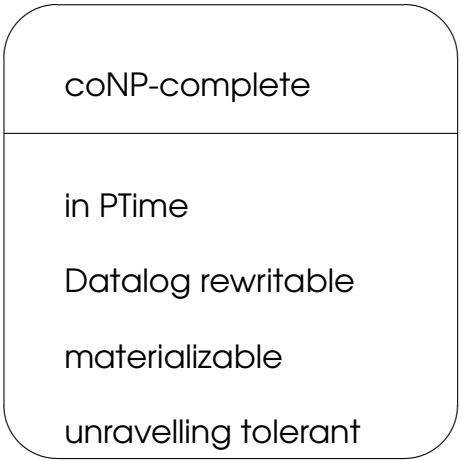
An ontology $\mathcal{O}$ is **unravelling tolerant** if for every $\mathcal{D}$ the following holds for the unravelling $\mathcal{D}^*$ of $\mathcal{D}$: for all acyclic conjunctive queries q:

$$\mathcal{O} \cup \mathcal{D} \models q \quad \Leftrightarrow \quad \mathcal{O} \cup \mathcal{D}^* \models q$$

Let $\mathcal{O}$ be a uGF or uGC_2 ontology. If $\mathcal{O}$ is unravelling tolerant, then $\mathcal{O}$ is in PTime (actually datalog-rewritable).

The Dichotomy Theorem

Let $\mathcal{O}$ be in any of the languages uGF$^-(1,=)$, uGF(1), uGF$_2^-(2)$, uGC$_2^-(1,=)$. Then we have the following classification:



Undecidability and Non-Dichotomy

In $\mathrm{uGF}_2^-(2, f)$ we have symbols for partial functions (weak counting), depth 2 formulas, and at most two variables.

For $\mathrm{uGF}_2^-(2, f)$ ontologies it is undecidable whether they are in PTime and whether they are coNP-hard (unless P=NP). Materializability and datalog rewritability are undecidable.

Undecidability and Non-Dichotomy

In $\mathrm{uGF}_2^-(2, f)$ we have symbols for partial functions (weak counting), depth 2 formulas, and at most two variables.

For $\mathrm{uGF}_2^-(2, f)$ ontologies it is undecidable whether they are in PTime and whether they are coNP-hard (unless P=NP). Materializability and datalog rewritability are undecidable.

To show non-dichotomy we prove a variation of Ladner's Theorem: The exists a Turing machine whose run fitting problem

can a partial run be extended to a full run of the machine

is neither in PTime nor NP-hard (unless P=NP).

Undecidability and Non-Dichotomy

In $\mathrm{uGF}_2^-(2, f)$ we have symbols for partial functions (weak counting), depth 2 formulas, and at most two variables.

For $\mathrm{uGF}_2^-(2, f)$ ontologies it is undecidable whether they are in PTime and whether they are coNP-hard (unless P=NP). Materializability and datalog rewritability are undecidable.

To show non-dichotomy we prove a variation of Ladner's Theorem: The exists a Turing machine whose run fitting problem

can a partial run be extended to a full run of the machine

is neither in PTime nor NP-hard (unless P=NP).

Using this result we show:

For $\mathrm{uGF}_2^-(2, f)$ ontologies there is no P/coNP dichotomy (unless P=NP).

Problems

- Is there a P/coNP dichotomy for uGF? Many smaller steps...

- Decidability: assume we have a dichotomy for a class of ontologies. Is it decidable whether an ontology $\mathcal{O}$ from the class is in PTime?