# Ontology-based Data Access: Theory and Practice


**Guohui Xiao**

*KRDB Research Centre*

*Free University of Bozen-Bolzano*

**Roman Kontchakov**

*Department of Computer Science & Inf. Systems*

*Birkbeck, University of London*


http://ontop.inf.unibz.it/ijcai-2018-tutorial

# Size of OMQ Rewritings: Target Languages

**UCQ** = unions of SPJ queries

as produced, e.g., by PerfectRef size $|q|^{|\mathcal{T}|} \cdot 2^{O(|q|^2)}$

# Size of OMQ Rewritings: Target Languages

**UCQ** = unions of SPJ queries

as produced, e.g., by PerfectRef size $|q|^{|\mathcal{T}|} \cdot 2^{O(|q|^2)}$

**PE** (positive existential formulas) = SPJU queries

# Size of OMQ Rewritings: Target Languages

**UCQ** = unions of SPJ queries

as produced, e.g., by PerfectRef size $|q|^{|\mathcal{T}|} \cdot 2^{O(|q|^2)}$

**PE** (positive existential formulas) = SPJU queries

$\Pi_2$**-PE** have matrix of the form $\wedge\vee$ = semiconjunctive queries (SCQs)

for ontologies without $\exists y$ on RHS, the size is $|q| \cdot |\mathcal{T}|$

# Size of OMQ Rewritings: Target Languages

**UCQ** = unions of SPJ queries

as produced, e.g., by PerfectRef size $|q|^{|\mathcal{T}|} \cdot 2^{O(|q|^2)}$

**PE** (positive existential formulas) = SPJU queries

**$\Pi_2$-PE** have matrix of the form $\wedge\vee$ = semiconjunctive queries (SCQs)

for ontologies without $\exists y$ on RHS, the size is $|q| \cdot |\mathcal{T}|$

**$\Sigma_3$-PE** have matrix of the form $\vee\wedge\vee$ = unions of SCQs

e.g., tree-witness rewriting size $O(2^{|\Theta_Q|} \cdot |Q|^2)$ with $\underbrace{|\Theta_Q| \leq 3^{|q|}}_{\text{number of tree witnesses}}$

# Size of OMQ Rewritings: Target Languages

**UCQ** = unions of SPJ queries

as produced, e.g., by PerfectRef size $|q|^{|\mathcal{T}|} \cdot 2^{O(|q|^2)}$

**PE** (positive existential formulas) = SPJU queries

**$\Pi_2$-PE** have matrix of the form $\wedge\vee$ = semiconjunctive queries (SCQs)

for ontologies without $\exists y$ on RHS, the size is $|q| \cdot |\mathcal{T}|$

**$\Sigma_3$-PE** have matrix of the form $\vee\wedge\vee$ = unions of SCQs

e.g., tree-witness rewriting size $O(2^{|\Theta_Q|} \cdot |Q|^2)$ with $\underbrace{|\Theta_Q| \leq 3^{|q|}}_{\text{number of tree witnesses}}$

**$\Pi_4$-PE** have matrix of the form $\wedge\vee\wedge\vee$

e.g., modified tree-witness rewriting without conflicts has size $O(|\Theta_Q| \cdot |Q|^2)$

# Size of OMQ Rewritings: Target Languages

**UCQ** = unions of SPJ queries

as produced, e.g., by PerfectRef size $|q|^{|\mathcal{T}|} \cdot 2^{O(|q|^2)}$

**PE** (positive existential formulas) = SPJU queries

**$\Pi_2$-PE** have matrix of the form $\wedge\vee$ = semiconjunctive queries (SCQs)

for ontologies without $\exists y$ on RHS, the size is $|q| \cdot |\mathcal{T}|$

**$\Sigma_3$-PE** have matrix of the form $\vee\wedge\vee$ = unions of SCQs

e.g., tree-witness rewriting size $O(2^{|\Theta_Q|} \cdot |Q|^2)$ with $\underbrace{|\Theta_Q| \leq 3^{|q|}}_{\text{number of tree witnesses}}$

**$\Pi_4$-PE** have matrix of the form $\wedge\vee\wedge\vee$

e.g., modified tree-witness rewriting without conflicts has size $O(|\Theta_Q| \cdot |Q|^2)$

**NDL** (non-recursive datalog) = PE + shared subformulas = SPJU + views

e.g., Presto produced rewritings of size $|\mathcal{T}|^{O(1)} \cdot 2^{|q|}$

# Size of OMQ Rewritings: Target Languages

**UCQ** = unions of SPJ queries

as produced, e.g., by PerfectRef size $|q|^{|\mathcal{T}|} \cdot 2^{O(|q|^2)}$

**PE** (positive existential formulas) = SPJU queries

**$\Pi_2$-PE** have matrix of the form $\wedge\vee$ = semiconjunctive queries (SCQs)

for ontologies without $\exists y$ on RHS, the size is $|q| \cdot |\mathcal{T}|$

**$\Sigma_3$-PE** have matrix of the form $\vee\wedge\vee$ = unions of SCQs

e.g., tree-witness rewriting size $O(2^{|\Theta_Q|} \cdot |Q|^2)$ with $\underbrace{|\Theta_Q| \leq 3^{|q|}}_{\text{number of tree witnesses}}$

**$\Pi_4$-PE** have matrix of the form $\wedge\vee\wedge\vee$

e.g., modified tree-witness rewriting without conflicts has size $O(|\Theta_Q| \cdot |Q|^2)$

**NDL** (non-recursive datalog) = PE + shared subformulas = SPJU + views

e.g., Presto produced rewritings of size $|\mathcal{T}|^{O(1)} \cdot 2^{|q|}$

**FO** (first-order formulas) = PE + negation ≈ RA

# Size of OMQ Rewritings: Target Languages

**UCQ** = unions of SPJ queries

as produced, e.g., by PerfectRef size $|q|^{|\mathcal{T}|} \cdot 2^{O(|q|^2)}$

**PE** (positive existential formulas) = SPJU queries

$\Pi_2$**-PE** have matrix of the form $\wedge\vee$ = semiconjunctive queries (SCQs)

for ontologies without $\exists y$ on RHS, the size is $|q| \cdot |\mathcal{T}|$

$\Sigma_3$**-PE** have matrix of the form $\vee\wedge\vee$ = unions of SCQs

e.g., tree-witness rewriting size $O(2^{|\Theta_Q|} \cdot |Q|^2)$ with $\underbrace{|\Theta_Q| \leq 3^{|q|}}_{\text{number of tree witnesses}}$

$\Pi_4$**-PE** have matrix of the form $\wedge\vee\wedge\vee$

e.g., modified tree-witness rewriting without conflicts has size $O(|\Theta_Q| \cdot |Q|^2)$

**NDL** (non-recursive datalog) = PE + shared subformulas = SPJU + views

e.g., Presto produced rewritings of size $|\mathcal{T}|^{O(1)} \cdot 2^{|q|}$

**FO** (first-order formulas) = PE + negation ≈ RA

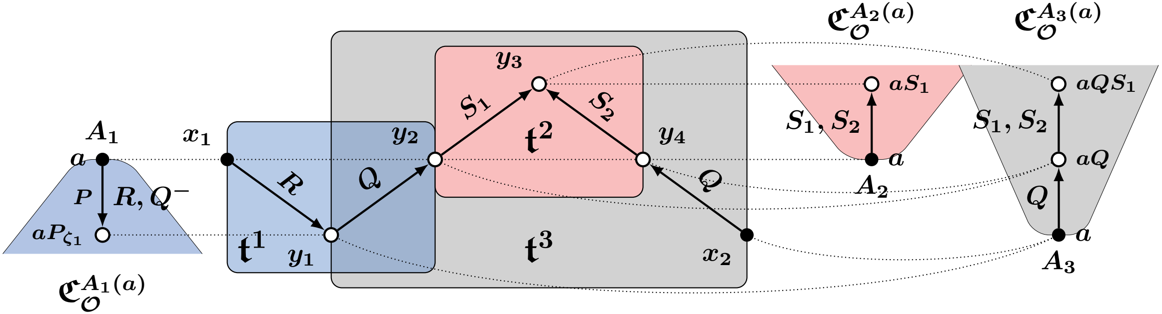
no combined approach (no additional constants, no assumptions on data)

# Tree-Witness Rewriting: Recap

$$\begin{array}{lll} A_1(x) \to \exists y\, P(x,y) & P(x,y) \to R(x,y) & P(x,y) \to Q(y,x) \\ A_2(x) \to \exists y\, S_1(x,y) & S_1(x,y) \to S_2(x,y) & \\ A_3(x) \to \exists y\, Q(x,y) & \exists y\, Q(y,x) \to \exists y\, S_1(x,y) & \end{array}$$

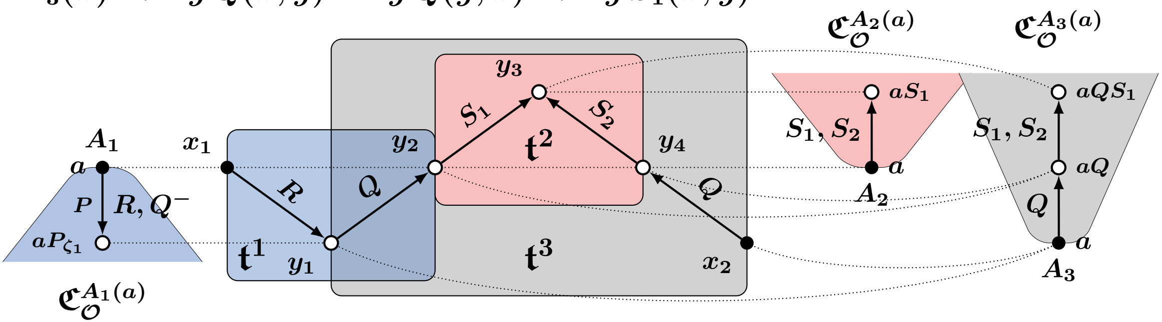
# Tree-Witness Rewriting: Recap

$$A_1(x) \to \exists y\, P(x,y) \qquad P(x,y) \to R(x,y) \qquad P(x,y) \to Q(y,x)$$
$$A_2(x) \to \exists y\, S_1(x,y) \qquad S_1(x,y) \to S_2(x,y)$$
$$A_3(x) \to \exists y\, Q(x,y) \qquad \exists y\, Q(y,x) \to \exists y\, S_1(x,y)$$



# Tree-Witness Rewriting: Recap

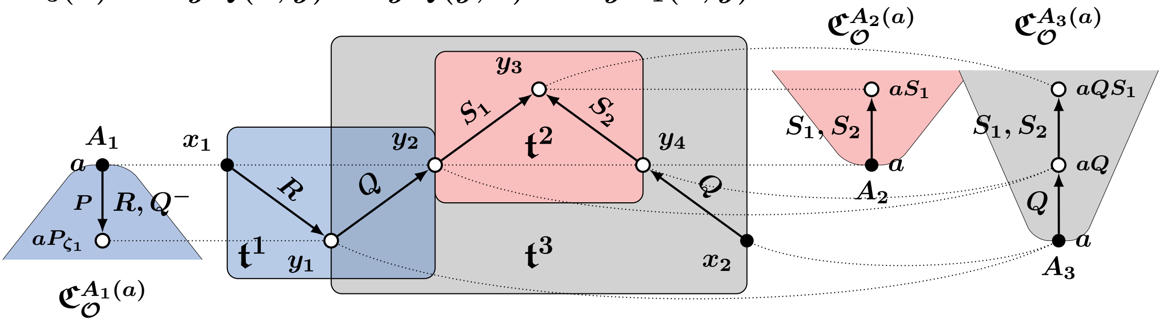
$$
\begin{array}{lll}
A_1(x) \to \exists y\, P(x,y) & P(x,y) \to R(x,y) & P(x,y) \to Q(y,x) \\
A_2(x) \to \exists y\, S_1(x,y) & S_1(x,y) \to S_2(x,y) & \\
A_3(x) \to \exists y\, Q(x,y) & \exists y\, Q(y,x) \to \exists y\, S_1(x,y) &
\end{array}
$$




$$
q' = \bigvee_{\Theta \subseteq \Theta_Q \text{ compatible}} \Big( \bigwedge_{S(z) \in q \setminus q_\Theta} S(z) \ \wedge \bigwedge_{\mathfrak{t} \in \Theta} \mathsf{tw}_{\mathfrak{t}} \Big)
$$

# Tree-Witness Rewriting: Recap

$$A_1(x) \to \exists y\, P(x,y) \qquad P(x,y) \to R(x,y) \qquad P(x,y) \to Q(y,x)$$
$$A_2(x) \to \exists y\, S_1(x,y) \qquad S_1(x,y) \to S_2(x,y)$$
$$A_3(x) \to \exists y\, Q(x,y) \qquad \exists y\, Q(y,x) \to \exists y\, S_1(x,y)$$

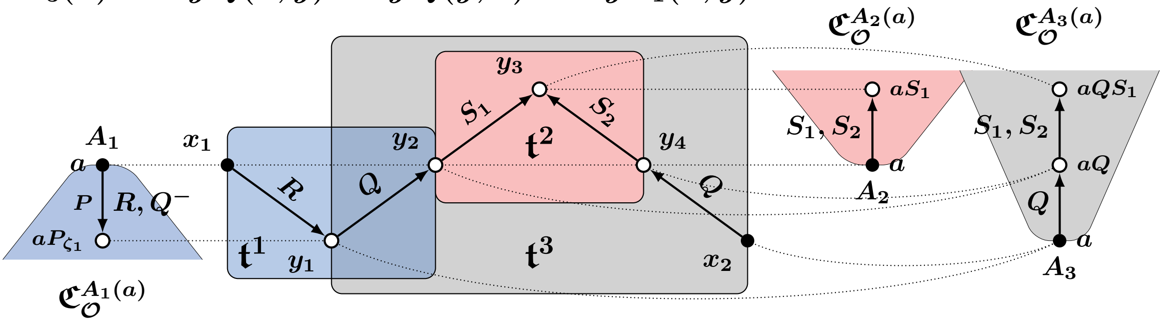



$$q' = \bigvee_{\Theta \subseteq \Theta_Q \text{ compatible}} \Big( \bigwedge_{S(z) \in q \setminus q_\Theta} S(z) \;\wedge \bigwedge_{\mathfrak{t} \in \Theta} \mathsf{tw}_{\mathfrak{t}} \Big)$$

5 **compatible** subsets:
$\emptyset$, $\{\mathfrak{t}_1\}$, $\{\mathfrak{t}_2\}$, $\{\mathfrak{t}_3\}$ and $\{\mathfrak{t}_1, \mathfrak{t}_2\}$
(exponentially many in general)

# Tree-Witness Rewriting: Recap

$$A_1(x) \to \exists y\, P(x,y) \qquad P(x,y) \to R(x,y) \qquad P(x,y) \to Q(y,x)$$
$$A_2(x) \to \exists y\, S_1(x,y) \qquad S_1(x,y) \to S_2(x,y)$$
$$A_3(x) \to \exists y\, Q(x,y) \qquad \exists y\, Q(y,x) \to \exists y\, S_1(x,y)$$

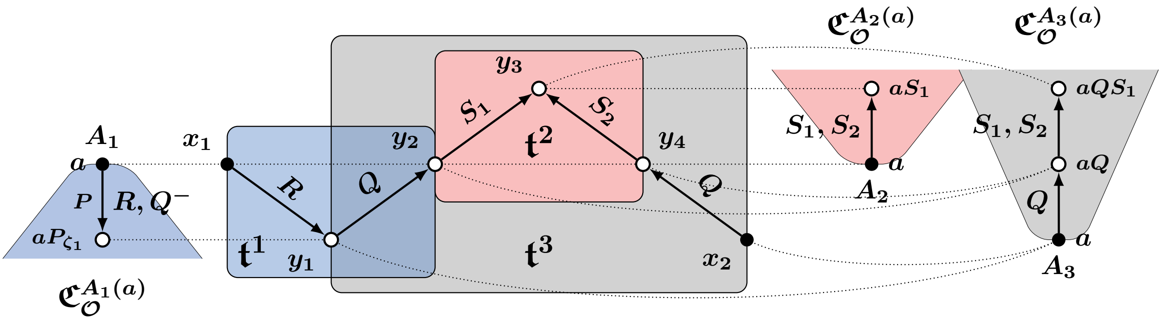



$$q' = \bigvee_{\Theta \subseteq \Theta_Q \text{ compatible}} \Big( \bigwedge_{S(z)\in q\setminus q_\Theta} S(z) \ \land \bigwedge_{\mathfrak{t}\in\Theta} \mathsf{tw}_{\mathfrak{t}}\Big)$$

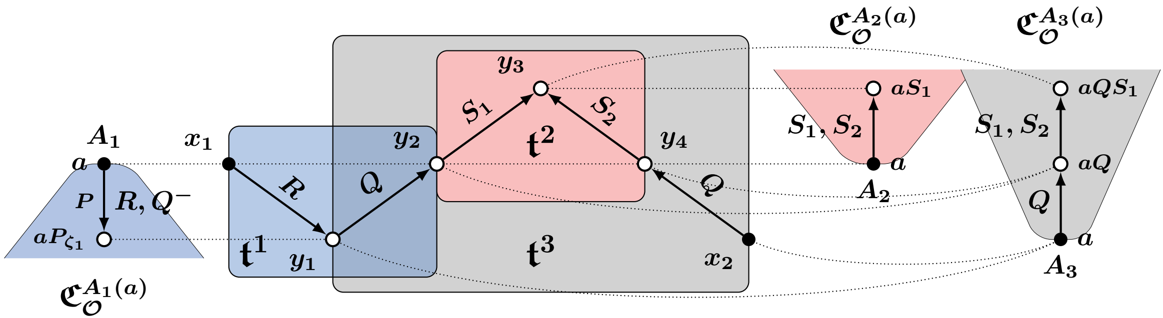
5 **compatible** subsets:
$\emptyset$, $\{\mathfrak{t}_1\}$, $\{\mathfrak{t}_2\}$, $\{\mathfrak{t}_3\}$ and $\{\mathfrak{t}_1, \mathfrak{t}_2\}$
(exponentially many in general)

a **homomorphism** $q \to \mathfrak{C}_{\mathcal{O}}(\mathcal{D})$ = a map from variables of $q$ to **ind**$(\mathcal{A})$
\+
an **compatible subset** of $\Theta_Q$ such that
1. each tree witness is 'generated' by the data
2. each atom outside tree witnesses is 'present' in the data

# Tree Witness Rewritings as Hypergraphs



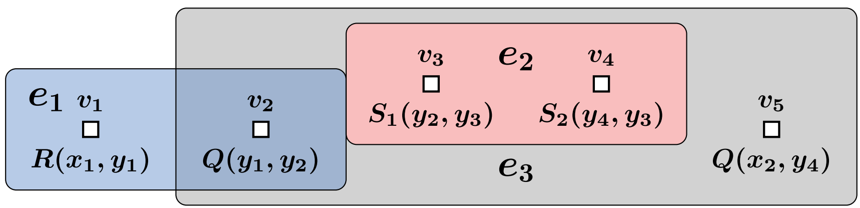
# Tree Witness Rewritings as Hypergraphs



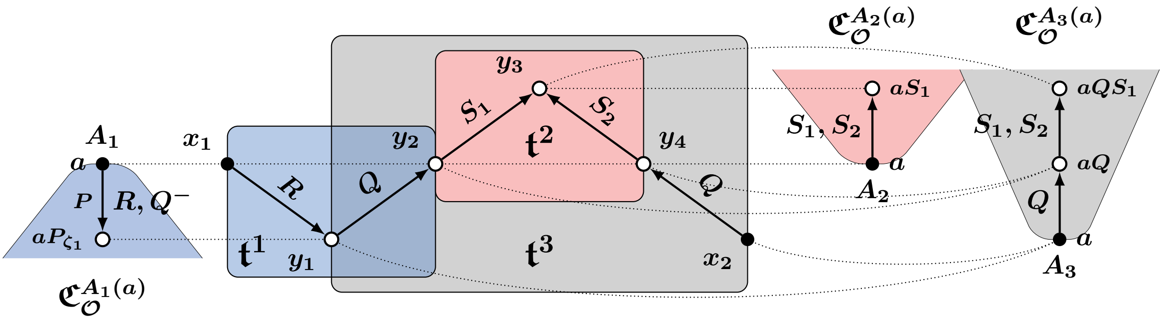

**OMQ hypergraph**

vertices = query atoms

hyperedges = sets of query atoms that can be mapped to trees



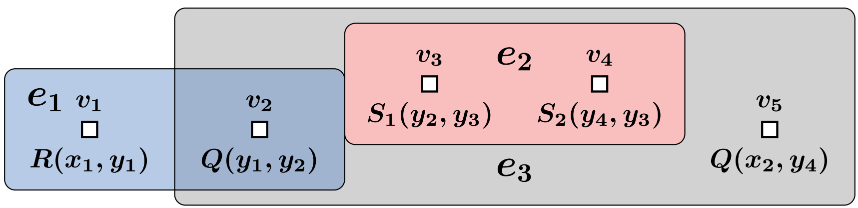
# Tree Witness Rewritings as Hypergraphs




**OMQ hypergraph**

vertices = query atoms

hyperedges = sets of query atoms that can be mapped to trees




$$f_H = \bigvee_{E' \text{ independent}} \Big( \bigwedge_{v \in V \setminus V_{E'}} p_v \wedge \bigwedge_{e \in E'} p_e \Big)$$

**hypergraph function $f_H$**

# From OMQs to Hypergraph Programs (HGPs)

a **HGP** $P$ is a hypergraph $H$ whose vertices are labelled by
$0$, $1$, or a literal over $p_1, \dots, p_n$

$P$ returns $1$ on an assignment $\alpha\colon \{p_1, \dots, p_n\} \to \{0, 1\}$
if there is an independent subset in $H$ that covers all zeroes under $\alpha$

# From OMQs to Hypergraph Programs (HGPs)

a **HGP** $P$ is a hypergraph $H$ whose vertices are labelled by $0$, $1$, or a literal over $p_1, \dots, p_n$

$P$ returns $1$ on an assignment $\alpha\colon \{p_1, \dots, p_n\} \to \{0, 1\}$ if there is an independent subset in $H$ that covers all zeroes under $\alpha$

a HGP is **monotone** if none of the labels is negative

degree of a hypergraph = the maximum number of hyperedges that contain a vertex

# From OMQs to Hypergraph Programs (HGPs)

a **HGP** $P$ is a hypergraph $H$ whose vertices are labelled by $0, 1$, or a literal over $p_1, \dots, p_n$

$P$ returns $1$ on an assignment $\alpha \colon \{p_1, \dots, p_n\} \to \{0, 1\}$ if there is an independent subset in $H$ that covers all zeroes under $\alpha$

a HGP is **monotone** if none of the labels is negative

degree of a hypergraph = the maximum number of hyperedges that contain a vertex

**class of OMQs**
all OMQs

size of rewritings

**(monotone) complexity class**
mNP/poly

**class of HGPs**
HGPs of degree 3

**class of circuits / NBPs**
nondeterministic Boolean circuits

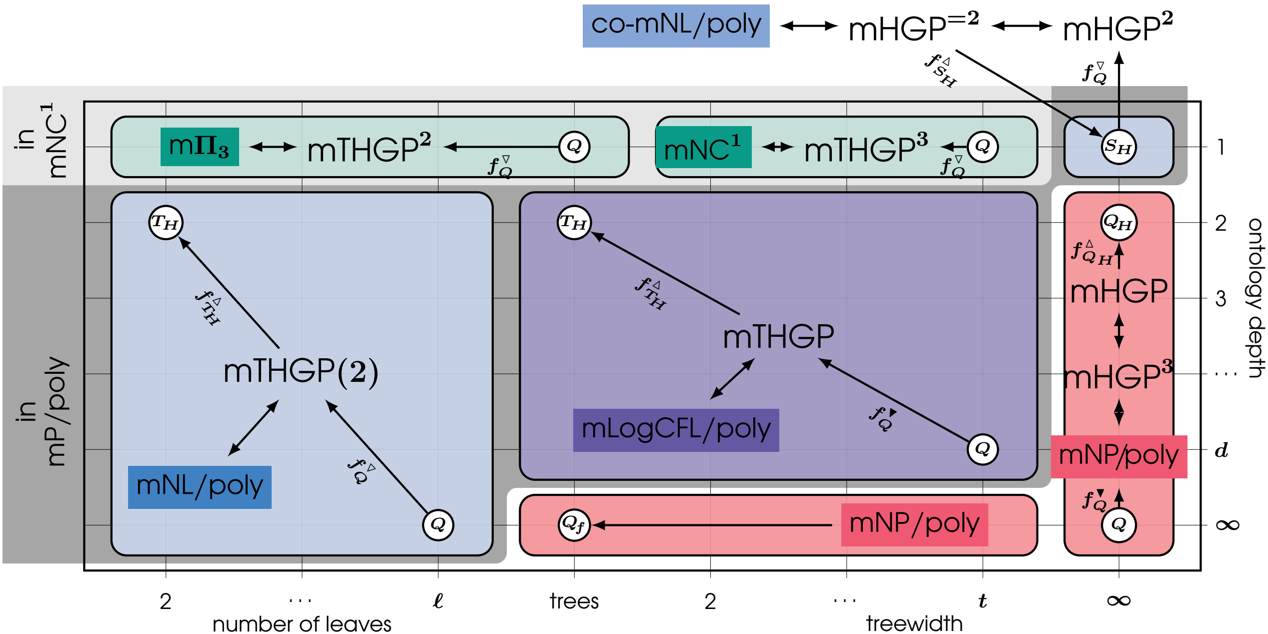
m$\Pi_3$ $\subsetneqq$ m$AC^0$ $\subsetneqq$ **m$NC^1$** $\subsetneqq$ mNL/poly $\subseteq$ mLogCFL/poly $\subsetneqq$ **mP/poly** $\subsetneqq$ **mNP/poly**

**monotone Boolean formulas (PE)**

**nondeterministic Boolean circuits (FO)**

**monotone circuits (NDL)**

# Roadmap for Succinctness Proofs

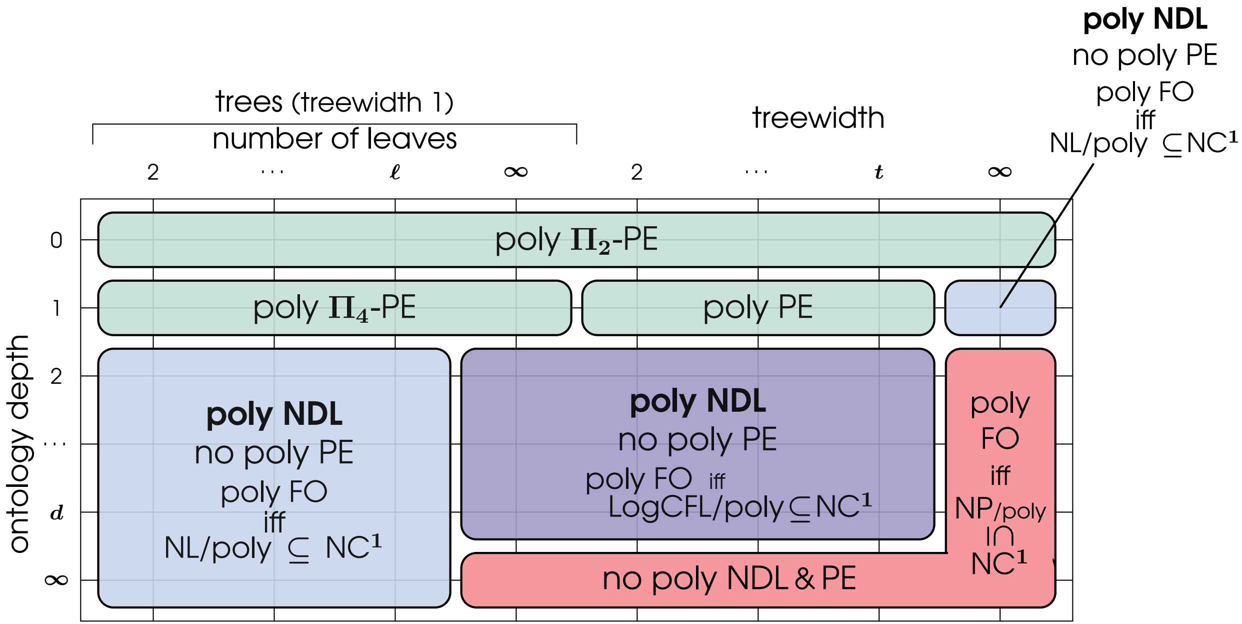



ontology depth

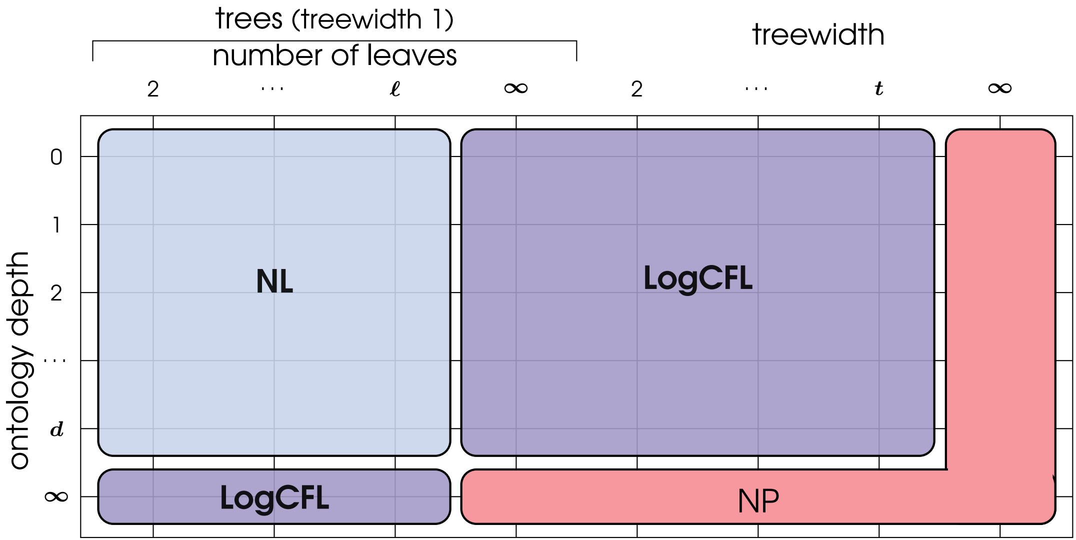
0 = no axioms with $\exists y$ on the right-hand side

$d$ $\approx$ trees $\mathfrak{C}^{\tau(a)}_{\mathcal{O}}$ of labelled nulls are of depth at most $d$

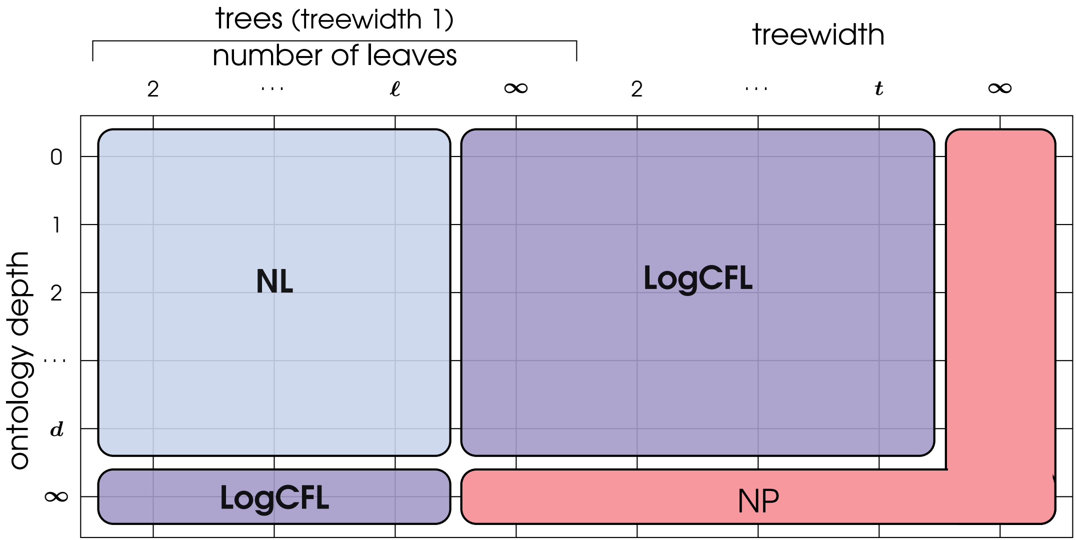
# Succinctness Landscape



# OMQ Answering Combined Complexity

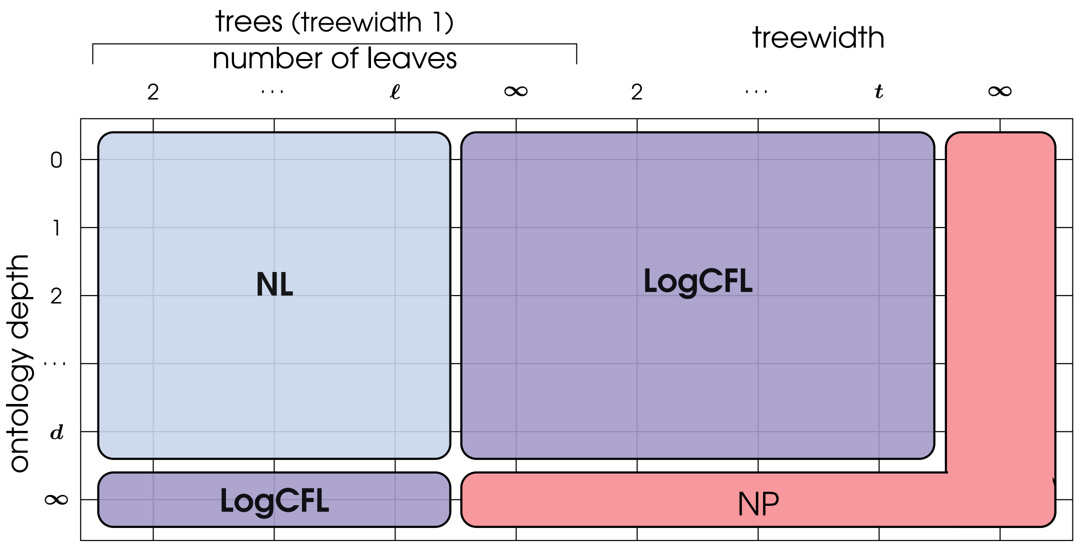


# OMQ Answering Combined Complexity




**NB:** polynomial-size NDL rewritings exist in all cases when OMQ answering is in LogCFL

# OMQ Answering Combined Complexity




**NB:** polynomial-size NDL rewritings exist in all cases when OMQ answering is in LogCFL

**constructed polynomial-size NDL rewritings that can be evaluated in LogCFL/NL**

**LogCFL** means that they are **highly parallelisable**

# References


1. M. Bienvenu, S. Kikot, R. Kontchakov, V. V. Podolskii & M. Zakharyaschev. "Ontology-mediated queries: combined complexity and succinctness of rewritings via circuit complexity". In: JACM, 2018.

2. M. Bienvenu, S. Kikot, R. Kontchakov, V. V. Podolskii, V. Ryzhikov & M. Zakharyaschev. "The complexity of ontology-based data access with OWL 2 QL and bounded treewidth queries". In: Proc. of the 36th ACM Symposium on Principles of Database Systems, PODS 2017, 2017.

3. G. Gottlob, S. Kikot, R. Kontchakov, V. V. Podolskii, T. Schwentick & M. Zakharyaschev. "The Price of Query Rewriting in Ontology-Based Data Access". In: Artificial Intelligence, 213:42–59, 2014.

4. R. Rosati & A. Almatelli. "Improving query answering over DL-Lite ontologies". In: Proc. of the 12th Int. Conf. on Principles of Knowledge Representation & Reasoning, KR'10, pp. 290–300. AAAI, 2010.

5. M. Thomazo. "Compact rewritings for existential rules". In: Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence IJCAI 2013, pp. 1125–1131. IJCAI/AAAI, 2013.