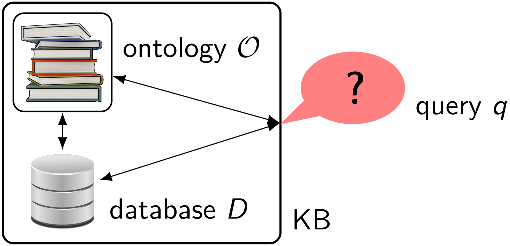
Querying the Unary Negation Fragment with Regular Path Expressions

Jean Christoph Jung Carsten Lutz Mauricio Martel
Thomas Schneider

AG Theorie der künstlichen Intelligenz, Universität Bremen
Funded by ERC Consolidator Grant 647289 CODA and DFG grant SCHN 1234/3

ICDT 2018 27 March

Ontology-mediated query answering



Ontology-mediated query (OMQ):

triple $Q = (\mathcal{O}, \Sigma, q)$ with Σ the data signature

OMQ evaluation:

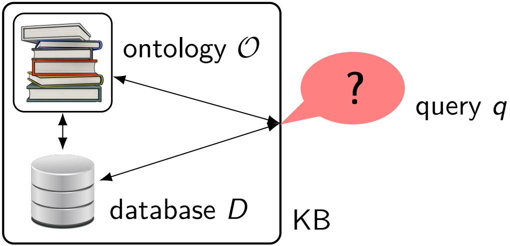
Given Q and D over Σ, does $D \cup \mathcal{O} \models q$ hold?

OMQ containment $Q_1 \subseteq Q_2$:

Given $Q_1 = (\mathcal{O}_1, \Sigma, q_1)$ and $Q_2 = (\mathcal{O}_2, \Sigma, q_2)$,

does $D \cup \mathcal{O}_1 \models q_1$ imply $D \cup \mathcal{O}_2 \models q_2$ for all D over Σ?

Ontology-mediated query answering



Typical ontology languages:

- Description logics (DLs)
- Languages based on existential rules (aka Datalog+/–)

Typical query languages: CQs, UCQs

OMQ evaluation in 2ExpTime (data complexity: coNP)

Unary Negation and regular expressions

We study: an expressive language that subsumes

- popular ontology languages
  (including transitive roles and/or regular expressions on roles)
- popular query languages
  (U)CQs, (U)C2RPQs

Unary Negation and regular expressions

We study: an expressive language that subsumes

- popular ontology languages
  (including transitive roles and/or regular expressions on roles)
- popular query languages
  (U)CQs, (U)C2RPQs

UNFO$^{\text{reg}}$ = UNFO + regular path expressions on binary relations

$$\varphi \; ::= \; P(\bar{x}) \mid \qquad x = y \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \neg\varphi(x)$$

Unary Negation and regular expressions

We study: an expressive language that subsumes

- popular ontology languages
  (including transitive roles and/or regular expressions on roles)
- popular query languages
  (U)CQs, (U)C2RPQs

UNFO$^{\text{reg}}$ = UNFO + regular path expressions on binary relations

$$\varphi \ ::= \ P(\bar{x}) \mid \mathbf{E(x,y)} \mid x = y \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \neg\varphi(x)$$

$$E \ ::= \ R \mid R^{-} \mid E \cup E \mid E \cdot E \mid E^{*} \mid \varphi(x)?$$

Unary Negation and regular expressions

We study: an expressive language that subsumes

- popular ontology languages
  (including transitive roles and/or regular expressions on roles)
- popular query languages
  (U)CQs, (U)C2RPQs

UNFO$^{\text{reg}}$ = UNFO + regular path expressions on binary relations

$$\varphi \ ::= \ P(\bar{\mathrm{x}}) \mid \mathbf{E(x,y)} \mid x = y \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \neg\varphi(x)$$

$$E \ ::= \ R \mid R^- \mid E \cup E \mid E \cdot E \mid E^* \mid \varphi(x)?$$

UC2RPQ = UCQ with atoms $E(x, y)$ as above (UC2RPQs as tests)
"union of conjunctive 2-way regular path queries"

UNFO$^{\text{reg}}$: a powerful language

UNFO$^{\text{reg}}$ can express ...

- **many popular MLs and DLs**
  (including transitive roles and/or regular expressions on roles)

- **C2RPQs $\rightsquigarrow$ queries are for free**
  OMQ evaluation and containment reduce to (un)satisfiability:

UNFO$^{\text{reg}}$: a powerful language

UNFO$^{\text{reg}}$ can express . . .

- **many popular MLs and DLs**
  (including transitive roles and/or regular expressions on roles)

- **C2RPQs $\rightsquigarrow$ queries are for free**
  OMQ evaluation and containment reduce to (un)satisfiability:
  - $D \cup \mathcal{O} \models q$ iff $\mathcal{O} \wedge D \wedge \neg q$ is unsatisfiable

UNFO$^{\text{reg}}$: a powerful language

UNFO$^{\text{reg}}$ can express . . .

- **many popular MLs and DLs**
  (including transitive roles and/or regular expressions on roles)
- **C2RPQs $\rightsquigarrow$ queries are for free**
  OMQ evaluation and containment reduce to (un)satisfiability:
  - $D \cup \mathcal{O} \models q$ iff $\mathcal{O} \wedge D \wedge \neg q$ is unsatisfiable
  - $(\mathcal{O}, \Sigma_{\text{full}}, q_1) \subseteq (\mathcal{O}, \Sigma_{\text{full}}, q_2)$ iff $\mathcal{O} \wedge q_1 \wedge \neg q_2$ is unsatisfiable

Our results

Theorem 1

(1) Satisfiability in $\mathrm{UNFO}^{\mathrm{reg}}$ is 2ExpTime-complete.

(2) Model checking is $\mathrm{P}^{\mathrm{NP}[O(\log^2 n)]}$-complete.

Same complexity as for UNFO [ten Cate & Segoufin, LMCS'13].

Our results

Theorem 1

(1) Satisfiability in UNFO^{reg} is 2ExpTime-complete.

(2) Model checking is $\text{P}^{\text{NP}[O(\log^2 n)]}$-complete.

Same complexity as for UNFO [ten Cate & Segoufin, LMCS'13].

Theorem 2

For OMQs with with $\mathcal{O}$ in UNFO^{reg} and q UC2RPQ,

(3) evaluation is 2ExpTime-compl. (combined), coNP-compl. (data);

(4) containment w.r.t. Σ_{full} is 2ExpTime-complete.

2ExpTime upper bounds follow from Theorem 1 (1).

All lower bounds inherited from $(\mathcal{ALCI}, \text{CQ})$.

[Lutz IJCAR'08; Calvanese et al. KR'06]

Overview

1. Results for Satisfiability
2. Results for OMQ Evaluation & Containment
3. Conclusion and Outlook

Next . . .

1. Results for Satisfiability
2. Results for OMQ Evaluation & Containment
3. Conclusion and Outlook

Schedule

We want to show:

Lemma 3

Satisfiability in UNFO$^{\text{reg}}$ is in 2ExpTime.

Previous approach for UNFO

Reduction to μ-calculus [ten Cate & Segoufin, LMCS'13]

✘ Does not transfer to UNFO$^{\text{reg}}$

Our approach

- show tree-like model property: treewidth $\leq$ width(φ)
- represent tree-like models via labelled trees
- characterise satisfaction of C2RPQs in labelled trees via tree-shaped witnesses
- build automata that accept the tree-like models of the input fma.

Normal form

UNFO$^{\text{reg}}$ formulas in **normal form**:

- connected C2RPQs with $= 1$ free variable and normal UNFO$^{\text{reg}}$ formulas $\psi(x)$? as tests
- $\neg\varphi(x) \quad \varphi(x) \vee \psi(x) \quad \exists x\, \varphi(x)$

Lemma 4

Every UNFO$^{\text{reg}}$ sentence can be transformed into an equivalent normal UNFO$^{\text{reg}}$ sentence φ' in exponential time.

- $|\varphi'|$ is exponential in $|\varphi|$
- **Width** of φ' (max. number of variables in a sub-C2RPQ): linear!

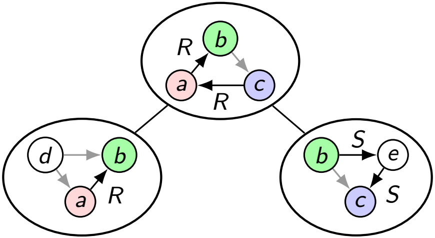
Tree-like structures

Tree-like structure of width m: pair $\mathfrak{T} = (T, \mathrm{bag})$ where

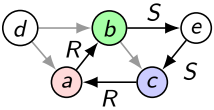
- T is a tree
- $\mathbf{bag}(w)$, $w \in T$: finite structure of domain size $\leq m$ such that for every element a, the set of nodes containing a is connected

Example

$\mathfrak{T}$ of width $m = 3$:



corresponding struc. $\mathfrak{A}_{\mathfrak{T}}$:



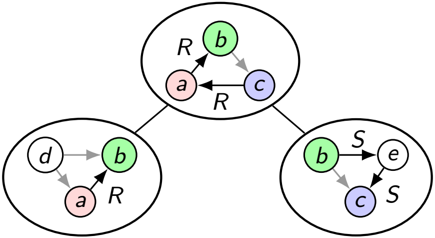
Tree-like structures

Tree-like structure of width m: pair $\mathfrak{T} = (T, \mathrm{bag})$ where

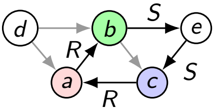
- T is a tree
- $\mathbf{bag}(w)$, $w \in T$: finite structure of domain size $\leq m$ such that for every element a, the set of nodes containing a is connected

Example

$\mathfrak{T}$ of width $m = 3$:



corresponding struc. $\mathfrak{A}_{\mathfrak{T}}$:



$$\mathfrak{A}_{\mathfrak{T}} \models \exists x \, RS^* R(x, x)$$

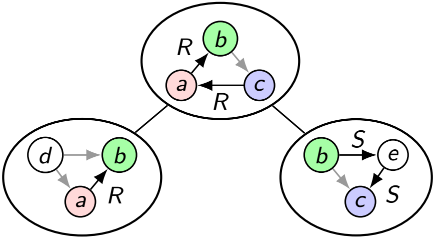
Tree-like structures

Tree-like structure of width m: pair $\mathfrak{T} = (T, \text{bag})$ where

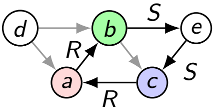
- T is a tree
- **bag(w)**, $w \in T$: finite structure of domain size $\leq m$ such that for every element a, the set of nodes containing a is connected

Example

$\mathfrak{T}$ of width $m = 3$:



corresponding struc. $\mathfrak{A}_{\mathfrak{T}}$:



$\mathfrak{A}_{\mathfrak{T}} \models \exists x\, RS^*R(x, x)$

because

$\mathfrak{A}_{\mathfrak{T}} \models RS^*R(a, a)$

Type-decorated tree-like structures

Fix UNFO$^{\text{reg}}$ formula φ_0.

Replace regular atoms $E(x, y)$ with atoms $\boldsymbol{\mathcal{A}(x, y)}$, $\mathcal{A}$ an **NFA**.

Type-decorated tree-like structure (TDTLS) for φ_0: pair $(\mathfrak{T}, \tau)$ s.t.

- $\mathfrak{T}$ is a tree-like structure
- τ assigns a 1-type $t(x) \subseteq \mathrm{cl}(\varphi_0)$ to each element in $\mathfrak{A}_{\mathfrak{T}}$

Satisfaction for regular atoms $\boldsymbol{(\mathfrak{T}, \tau)} \models \boldsymbol{\mathcal{A}(a, b)}$ is defined via "looking up" tests $\varphi(a)$? directly in $\tau(a)$

Proper TDTLSs

Labelling τ must be "globally" consistent with all $\exists$- and C2RPQ-subformulas of φ_0:

A TDTLS is proper if

(1) $\exists x\, \varphi(x) \in \tau(a)$ are witnessed by $\varphi(x) \in \tau(b)$ for some b

(2) C2RPQs in $\tau(a)$ are witnessed by homomorphisms into $(\mathfrak{T}, \tau)$

Proper TDTLSs

Labelling τ must be "globally" consistent with all $\exists$- and C2RPQ-subformulas of φ_0:

A TDTLS is proper if

(1) $\exists x\, \varphi(x) \in \tau(a)$ are witnessed by $\varphi(x) \in \tau(b)$ for some b

(2) C2RPQs in $\tau(a)$ are witnessed by homomorphisms into $(\mathfrak{T}, \tau)$

Lemma 5 (Tree-like model property of UNFO$^{\text{reg}}$)

Let φ_0 be a normal UNFO$^{\text{reg}}$ sentence.

φ_0 is satisfiable iff there is a proper TDTLS $(\mathfrak{T}, \tau)$ for φ_0 s.t. $\varphi_0 \in \tau(a)$ for some a.

Proper TDTLSs

Labelling τ must be "globally" consistent with all $\exists$- and C2RPQ-subformulas of φ_0:

A TDTLS is proper if

(1) $\exists x\, \varphi(x) \in \tau(a)$ are witnessed by $\varphi(x) \in \tau(b)$ for some b

(2) C2RPQs in $\tau(a)$ are witnessed by homomorphisms into $(\mathfrak{T}, \tau)$

Lemma 5 (Tree-like model property of UNFO$^{\text{reg}}$)

Let φ_0 be a normal UNFO$^{\text{reg}}$ sentence **of size n**.

φ_0 is satisfiable iff there is a proper TDTLS $(\mathfrak{T}, \tau)$ for φ_0 **of outdegree $O(n^2)$** s.t. $\varphi_0 \in \tau(a)$ for some a.

Proper TDTLSs

Labelling τ must be "globally" consistent with all $\exists$- and C2RPQ-subformulas of φ_0:

A TDTLS is proper if

(1) $\exists x\, \varphi(x) \in \tau(a)$ are witnessed by $\varphi(x) \in \tau(b)$ for some b

(2) C2RPQs in $\tau(a)$ are witnessed by homomorphisms into $(\mathfrak{T}, \tau)$

Lemma 5 (Tree-like model property of UNFO$^{\text{reg}}$)

Let φ_0 be a normal UNFO$^{\text{reg}}$ sentence **of size n and width m**.

φ_0 is satisfiable iff there is a proper TDTLS $(\mathfrak{T}, \tau)$ for φ_0 **of outdegree $O(n^2)$ and width $\leq m$** s.t. $\varphi_0 \in \tau(a)$ for some a.

Automata-based decision procedure

Use 2-way alternating tree automata **(2ATAs)**

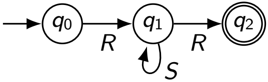
$\mathcal{A}$ accepts input $(\mathfrak{T}, \tau)$ iff $\mathfrak{T}$ proper and $\varphi_0 \in \tau(a)$ for some a

Then: φ_0 satisfiable iff $L(\mathcal{A}) \neq \emptyset$

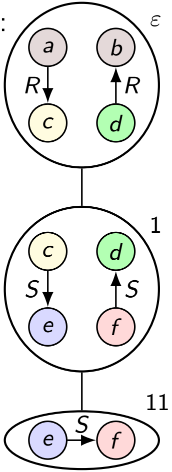
Problem: How can $\mathcal{A}$ check Condition (2) of properness?

(2) C2RPQs in $\tau(a)$ are witnessed by $\underbrace{\text{homomorphisms into } (\mathfrak{T}, \tau)}_{\textbf{need not be local!}}$

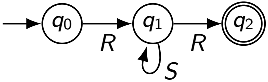
Non-locality of regular atoms

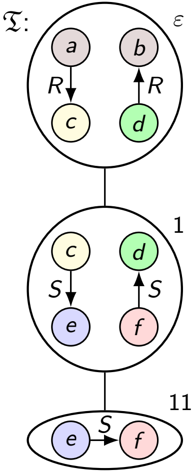
Example: C2RPQ $\mathcal{A}(x, y)$ with $\mathcal{A} =$ 

$\mathfrak{T}$:



Non-locality of regular atoms

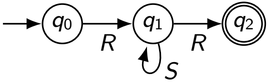
Example: C2RPQ $\mathcal{A}(x, y)$ with $\mathcal{A} =$ 

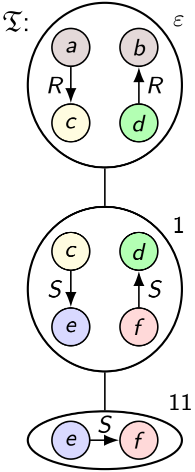


$\mathfrak{A}_{\mathfrak{T}} \models \mathcal{A}(a, b)$; h targets bags $\varepsilon, 1, 11$

Non-locality of regular atoms

Example: C2RPQ $\mathcal{A}(x, y)$ with $\mathcal{A} =$





$\mathfrak{A}_{\mathfrak{T}} \models \mathcal{A}(a, b)$; h targets bags $\varepsilon, 1, 11$

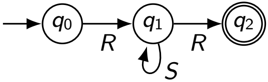
(1) Subdivide atom $\mathcal{A}(a, b)$

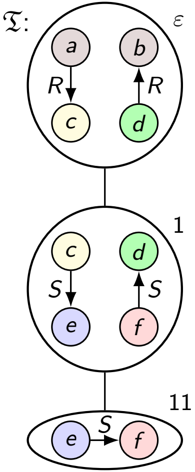
$$\mathcal{A}[q_0, q_1](a, c) \quad \wedge \quad \mathcal{A}[q_1, q_2](d, b) \quad \wedge$$

$$\mathcal{A}[q_1, q_1](c, e) \quad \wedge \quad \mathcal{A}[q_1, q_1](f, d) \quad \wedge$$

$$\mathcal{A}[q_1, q_1](e, f)$$

Non-locality of regular atoms

Example: C2RPQ $\mathcal{A}(x, y)$ with $\mathcal{A} =$ 



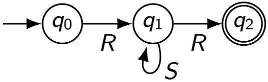
$\mathfrak{A}_{\mathfrak{T}} \models \mathcal{A}(a, b)$; h targets bags $\varepsilon, 1, 11$

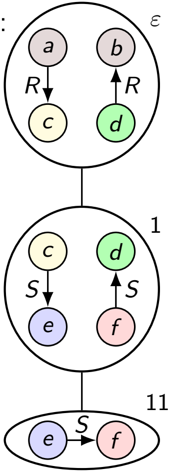
(1) Subdivide atom $\mathcal{A}(a, b)$

$$\mathcal{A}[q_0, q_1](a, c) \quad \wedge \quad \mathcal{A}[q_1, q_2](d, b) \quad \wedge$$
$$\mathcal{A}[q_1, q_1](c, e) \quad \wedge \quad \mathcal{A}[q_1, q_1](f, d) \quad \wedge$$
$$\mathcal{A}[q_1, q_1](e, f)$$

Subdivision into $O(|\varphi_0|^2)$ atoms suffices

Non-locality of regular atoms

Example: C2RPQ $\mathcal{A}(x,y)$ with $\mathcal{A} =$ 

$\mathfrak{T}$: 

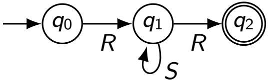
$\mathfrak{A}_{\mathfrak{T}} \models \mathcal{A}(a,b)$; h targets bags $\varepsilon, 1, 11$

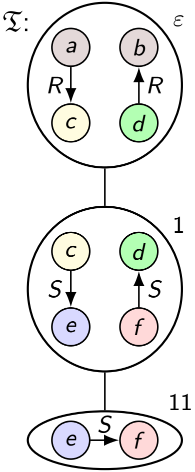
(2) Split the resulting set of atoms

$$\begin{array}{c}
\mathcal{A}[q_0,q_1](a,c) \quad \wedge \quad \mathcal{A}[q_1,q_2](d,b) \\
\hline
\mathcal{A}[q_1,q_1](c,e) \quad \wedge \quad \mathcal{A}[q_1,q_1](f,d) \\
\hline
\mathcal{A}[q_1,q_1](e,f)
\end{array}$$

Non-locality of regular atoms

Example: C2RPQ $\mathcal{A}(x,y)$ with $\mathcal{A} =$





$\mathfrak{A}_{\mathfrak{T}} \models \mathcal{A}(a,b)$; h targets bags $\varepsilon, 1, 11$

Resulting witness tree:

$$\Big(\varepsilon,\ \mathcal{A}[q_0,q_1](a,c) \ \wedge\ \mathcal{A}[q_1,q_2](d,b)\Big)$$

$$\Big(1,\ \mathcal{A}[q_1,q_1](c,e) \ \wedge\ \mathcal{A}[q_1,q_1](f,d)\Big)$$

$$\Big(11,\ \mathcal{A}[q_1,q_1](e,f)\Big)$$

Automata-friendly properness condition

Good news:

Lemma 6

Subdivisions and splittings introduce at most exponentially many new sub-C2RPQs.

Condition (2) of properness is equivalent to:

(2′) C2RPQs in $\tau(a)$ are witnessed by witness trees in $(\mathfrak{T}, \tau)$

$\rightsquigarrow$ construct 2ATA with exponentially many states

Next ...

1. Results for Satisfiability
2. Results for OMQ Evaluation & Containment
3. Conclusion and Outlook

Overview

We want to show:

Lemma 7

OMQ evaluation is in coNP (data complexity).

Observation: $D \cup \mathcal{O} \models q$ iff $\underbrace{D \text{ is unsatisfiable with } \mathcal{O} \wedge \neg q}$

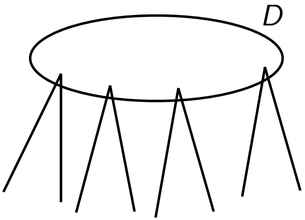
Lemma 8

Satisfiability of databases with UNFO^{reg} sentences is in NP.

Database satisfiability in NP

Fix UNFO$^{\text{reg}}$ sentence φ_0 and database D

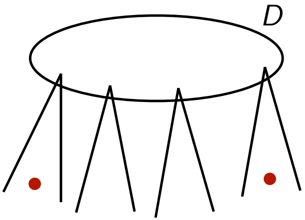
Idea: guess & check consistent decoration of dom(D) with 1-types



Database satisfiability in NP

Fix $\mathrm{UNFO}^{\mathrm{reg}}$ sentence φ_0 and database D

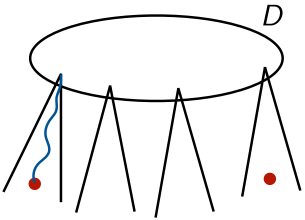
Idea: guess & check consistent decoration of dom(D) with 1-types



Database satisfiability in NP

Fix UNFO$^{\text{reg}}$ sentence φ_0 and database D

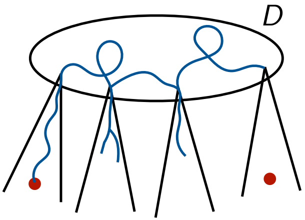
Idea: guess & check consistent decoration of dom(D) with 1-types



Database satisfiability in NP

Fix UNFO$^{\text{reg}}$ sentence φ_0 and database D

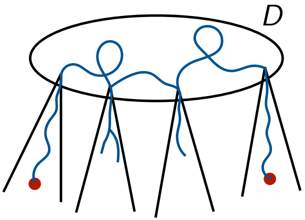
Idea: guess & check consistent decoration of dom(D) with 1-types



Database satisfiability in NP

Fix UNFO$^{\text{reg}}$ sentence φ_0 and database D

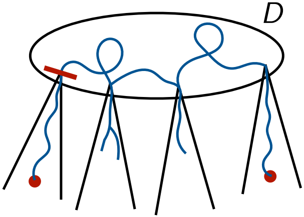
Idea: guess & check consistent decoration of dom(D) with 1-types



Database satisfiability in NP

Fix $\mathrm{UNFO}^{\mathrm{reg}}$ sentence φ_0 and database D

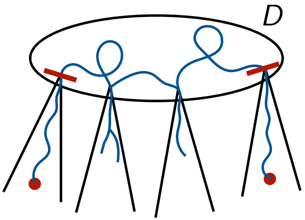
Idea: guess & check consistent decoration of $\mathrm{dom}(D)$ with 1-types



Database satisfiability in NP

Fix UNFO$^{\text{reg}}$ sentence φ_0 and database D

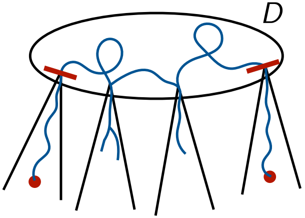
Idea: guess & check consistent decoration of dom(D) with 1-types



Database satisfiability in NP

Fix $\mathrm{UNFO}^{\mathrm{reg}}$ sentence φ_0 and database D

Idea: guess & check consistent decoration of $\mathrm{dom}(D)$ with 1-types



$\rightsquigarrow$ 3-subdivisions of regular atoms $\mathcal{A}(x, y)$ suffice

Database satisfiability in NP

Type decoration τ is **proper** if

- each $\tau(a)$ is satisfiable
- similar conditions as (1) and (2) for proper TDTLSs

Lemma 9

D is satisfiable with φ_0 iff D has a proper type decoration τ with $\varphi_0 \in \tau(a)$ for some $a \in \mathrm{dom}(D)$.

$\rightsquigarrow$ **NP decision procedure:**

- Guess candidate decoration
- Precompute satisfaction of all regular atoms
- Verify properness

Next . . .

1. Results for Satisfiability
2. Results for OMQ Evaluation & Containment
3. Conclusion and Outlook

Conclusion

UNFO$^{\text{reg}}$ is computationally well-behaved:

- SAT and model checking are not harder than in UNFO.
- OMQ evaluation and OMQ containment for (UNFO$^{\text{reg}}$, C2RPQ) are not harder than for ($\mathcal{ALCI}$, CQ).

Outlook

- Finite-model variants of SAT and OMQ evaluation
- Extensions:
  - Base case in regular path expressions with *any* UNFO^{reg} formula $\varphi(x, y)$ instead of $R(x, y)$ $\rightsquigarrow$ contains ICPDL
  - Replace C2RPQs with linear Datalog
  - Add constants, fixed points, role inclusions ... ("DL world")
- OMQ containment, general case: $(\mathcal{O}_1, \mathbf{\Sigma}, q_1) \overset{?}{\subseteq} (\mathcal{O}_2, \mathbf{\Sigma}, q_2)$

Outlook

- Finite-model variants of SAT and OMQ evaluation
- Extensions:
  - Base case in regular path expressions with *any* $\mathrm{UNFO}^{\mathrm{reg}}$ formula $\varphi(x, y)$ instead of $R(x, y)$ $\rightsquigarrow$ contains ICPDL
  - Replace C2RPQs with linear Datalog
  - Add constants, fixed points, role inclusions . . . (“DL world”)
- OMQ containment, general case: $(\mathcal{O}_1, \mathbf{\Sigma}, q_1) \overset{?}{\subseteq} (\mathcal{O}_2, \mathbf{\Sigma}, q_2)$

Thank you.