Towards Practical Query Answering for Horn-$\mathcal{SHIQ}$

Thomas Eiter[1] Magdalena Ortiz[1] Mantas Šimkus[1]
Trung-Kien Tran [2] Guohui Xiao[1]

[1]Institute of Information Systems, Vienna University of Technology

[2]STARLab, Vrije Universiteit Brussel

25th International Workshop on Description Logics (DL 2012)

Query Answering in DLs

Ontology Based Data Access is a key application of DLs
Hence, query answering in DLs a crucial problem

Evaluate a conjunctive query q over an ABox $\mathcal{A}$, taking into account the constraints expressed by a DL TBox $\mathcal{T}$

Query Answering in DLs

Ontology Based Data Access is a key application of DLs
Hence, query answering in DLs a crucial problem

Evaluate a conjunctive query q over an ABox $\mathcal{A}$, taking into account the constraints expressed by a DL TBox $\mathcal{T}$

$$\text{hasDevelopedCapital}(x) \leftarrow \text{country}(x), \text{hasCapital}(x, y), \text{city}(y), \text{hasHDI}(y, \mathit{high})$$

$\mathcal{A}$

country(*Brazil*)
capital(*Brasilia*)
isLocatedIn(*Brasilia*, *RegiãoCentroOeste*)
isLocatedIn(*RegiãoCentroOeste*, *Brazil*)
hasHDI(*Brasilia*, high)

Query Answering in DLs

Ontology Based Data Access is a key application of DLs
Hence, query answering in DLs a crucial problem

Evaluate a conjunctive query q over an ABox $\mathcal{A}$, taking into account the constraints expressed by a DL TBox $\mathcal{T}$

$$\mathrm{hasDevelopedCapital}(x) \leftarrow \mathrm{country}(x), \mathrm{hasCapital}(x,y), \mathrm{city}(y), \mathrm{hasHDI}(y, \mathit{high})$$

$$\mathrm{country}(\textit{Brazil}) \qquad \mathrm{hasHDI}(\textit{Brasilia}, \mathit{high})$$

$\mathcal{A}$

country(*Brazil*)
capital(*Brasilia*)
isLocatedIn(*Brasilia*, *RegiãoCentroOeste*)
isLocatedIn(*RegiãoCentroOeste*, *Brazil*)
hasHDI(*Brasilia*, *high*)

Query Answering in DLs

Ontology Based Data Access is a key application of DLs
Hence, query answering in DLs a crucial problem

Evaluate a conjunctive query q over an ABox $\mathcal{A}$, taking into account the constraints expressed by a DL TBox $\mathcal{T}$

$$\text{hasDevelopedCapital}(x) \leftarrow \text{country}(x), \text{hasCapital}(x,y), \text{city}(y), \text{hasHDI}(y, \mathit{high})$$

$$\text{country}(\mathit{Brazil}) \qquad \text{hasHDI}(\mathit{Brasilia}, \mathit{high})$$

$$\text{hasCapital}(\mathit{Brazil}, \mathit{Brasilia}), \text{city}(\mathit{Brasilia}),$$

$\mathcal{A}$

country(*Brazil*)
capital(*Brasilia*)
isLocatedIn(*Brasilia*, *RegiãoCentroOeste*)
isLocatedIn(*RegiãoCentroOeste*, *Brazil*)
hasHDI(*Brasilia*, *high*)

$\mathcal{T}$

$$\begin{aligned} &\mathit{trans}(\text{isLocatedIn}) \\ \text{country} &\sqsubseteq \exists\text{hasCapital.capital} \\ \text{hasCapital} &\sqsubseteq \text{isLocatedIn} \\ \text{country} &\sqsubseteq \leqslant 1\, \text{isLocatedIn}^{-}.\text{capital} \\ \text{country} &\sqsubseteq \forall\text{hasCapital.city} \end{aligned}$$

Query Answering in DLs

Ontology Based Data Access is a key application of DLs
Hence, query answering in DLs a crucial problem

Evaluate a conjunctive query q over an ABox $\mathcal{A}$, taking into account the constraints expressed by a DL TBox $\mathcal{T}$

$$\text{hasDevelopedCapital}(x) \leftarrow \text{country}(x), \text{hasCapital}(x, y), \text{city}(y), \text{hasHDI}(y, \textit{high})$$

$$\text{hasDevelopedCapital}(\textit{Brazil}) \leftarrow \text{country}(\textit{Brazil}) \quad \text{hasCapital}(\textit{Brazil}, \textit{Brasilia}), \text{city}(\textit{Brasilia}), \quad \text{hasHDI}(\textit{Brasilia}, \textit{high})$$

$\mathcal{A}$	$\mathcal{T}$
country(*Brazil*)	*trans*(isLocatedIn)
capital(*Brasilia*)	country $\sqsubseteq \exists$hasCapital.capital
isLocatedIn(*Brasilia*, *RegiãoCentroOeste*)	hasCapital $\sqsubseteq$ isLocatedIn
isLocatedIn(*RegiãoCentroOeste*, *Brazil*)	country $\sqsubseteq \leqslant 1$ isLocatedIn$^-$.capital
hasHDI(*Brasilia*, *high*)	country $\sqsubseteq \forall$hasCapital.city

State of the art

For lightweight DLs, successful query answering using database technologies

For $\mathcal{DL}$-*Lite*

- Query rewriting compiles q and $\mathcal{T}$ into a UCQ/FO query $q^{\mathcal{T}}$
- $q^{\mathcal{T}}$ can be evaluated over $\mathcal{A}$ only with off-the-shelf RDBMSs
- Many papers on better and shorter rewritings

For $\mathcal{EL}$

- Query rewriting into Datalog (e.g.,. Requiem), no FO rewritability
- Alternative: the combined approach
  - TBox partially materialized in $\mathcal{A}$ (polynomial expansion)
  - q rewritten into a FO query over the expanded data
  - evaluation possible with off-the-shelf RDBMSs

State of the art (cont'd)

For more expressive DLs

- (full) CQ answering not supported by reasoners
- algorithms don't seem implementable
- many bad complexity results
  - $\mathcal{ALCI}$ and $\mathcal{SH}$ are 2-ExpTime-hard in combined complexity
  - already $\mathcal{AL}$ intractable in data complexity

State of the art (cont'd)

For more expressive DLs

- (full) CQ answering not supported by reasoners
- algorithms don't seem implementable
- many bad complexity results
  - $\mathcal{ALCI}$ and $\mathcal{SH}$ are 2-ExpTime-hard in combined complexity
  - already $\mathcal{AL}$ intractable in data complexity ➥bad news!

State of the art (cont'd)

For more expressive DLs

- (full) CQ answering not supported by reasoners
- algorithms don't seem implementable
- many bad complexity results
  - $\mathcal{ALCI}$ and $\mathcal{SH}$ are 2-ExpTime-hard in combined complexity
  - already $\mathcal{AL}$ intractable in data complexity ➥bad news!

- Is query answering beyond $\mathcal{DL}$-*Lite* and $\mathcal{EL}$ practicable?
- Can we realize it using existing efficient technologies?
  (RDBMS, Datalog engines, etc.)

Query Answering in Horn-$\mathcal{SHIQ}$

Horn fragments of DLs like $\mathcal{SHIQ}$ seem promising candidates

- Horn-$\mathcal{SHIQ}$ is tractable in data complexity (PTime-complete)
- The combined complexity is not higher than for standard reasoning
  - ExpTime-complete
- It has useful features not present in $\mathcal{EL}$ and $\mathcal{DL}$-*Lite*

$trans(\text{isLocatedIn})$ $\text{country} \sqsubseteq \forall \text{hasCapital.city}$ $\text{country} \sqsubseteq\ \leqslant 1\, \text{isLocatedIn}^{-}.\text{capital}$

Query Answering in Horn-$\mathcal{SHIQ}$

Horn fragments of DLs like $\mathcal{SHIQ}$ seem promising candidates

- Horn-$\mathcal{SHIQ}$ is tractable in data complexity (PTime-complete)
- The combined complexity is not higher than for standard reasoning
  - ExpTime-complete
- It has useful features not present in $\mathcal{EL}$ and $\mathcal{DL}$-*Lite*

$$\mathit{trans}(\text{isLocatedIn}) \quad \text{country} \sqsubseteq \forall \text{hasCapital}.\text{city} \quad \text{country} \sqsubseteq\, \leqslant 1\, \text{isLocatedIn}^{-}.\text{capital}$$

- These features make the problem significantly more complex
- Incorporating them into existing techniques is not trivial

Query Answering for Horn-$\mathcal{SHIQ}$

Our Contribution

We present a novel query rewriting algorithm for Horn-$\mathcal{SHIQ}$ and a prototype implementation that shows promising results.

- We rewrite q into a UCQ $\mathsf{rew}_{\mathcal{T}}(q)$ (depends on the TBox $\mathcal{T}$)
- The (non-existential) axioms of $\mathcal{T}$ are rewritten into Datalog rules $\mathsf{cr}(\mathcal{T})$
- Answering q over $(\mathcal{T}, \mathcal{A})$ amounts to evaluating the Datalog program

$$\mathcal{A} \cup \mathsf{cr}(\mathcal{T}) \cup \mathsf{rew}_{\mathcal{T}}(q)$$

  - We can also evaluate $\mathsf{rew}_{\mathcal{T}}(q)$ over the completion of $\mathcal{A}$ (with no additional unnamed objects)
- $\mathsf{rew}_{\mathcal{T}}(q)$ can be exponential, but has manageable size for real queries and ontologies

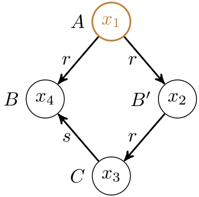
The rewriting algorithm

Main idea:

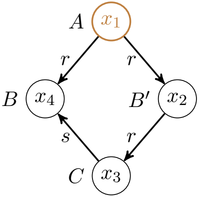
- Eliminate query variables that can be matched at unnamed objects
  - Query matches have tree-shaped parts
  - We clip off the variables x that can be leaves
  - Replace them by constraints $D(y)$ on their parent variables y
  - The added atoms $D(y)$ ensure the existence of a match for x
- In the resulting queries all variables are matched to named objects

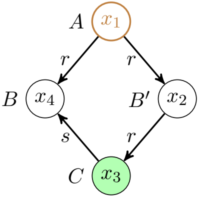
One step of query rewriting

$$q(x_1) \leftarrow r(x_1, x_2), r(x_1, x_4), r(x_2, x_3), s(x_3, x_4), A(x_1), B(x_4), B'(x_2), C(x_3)$$



One step of query rewriting



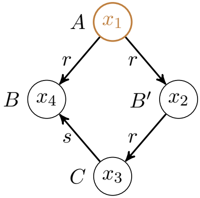


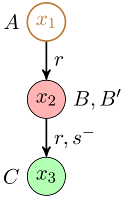
1. Select the non-distinguished variable x_3

One step of query rewriting

1. Select the non-distinguished variable x_3
2. Ensure that x_3 has only incoming edges
   ➤replace $r(x, y)$ by $r^{-}(y, x)$ as needed

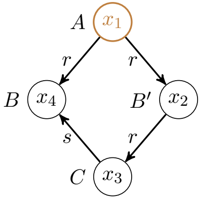
One step of query rewriting



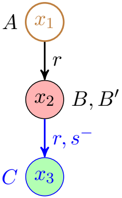


1. Select the non-distinguished variable x_3
2. Ensure that x_3 has only incoming edges
   ➤replace $r(x, y)$ by $r^-(y, x)$ as needed
3. Merge the predecessors
   ➤if x_3 is a leaf of a tree, they must be mapped together

One step of query rewriting

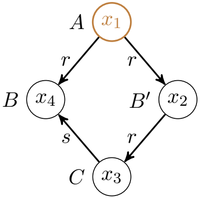


$\mathcal{T} \models D \sqsubseteq \exists(r \sqcap s^-).C$

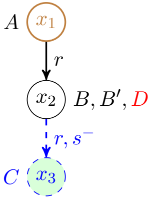


1. Select the non-distinguished variable x_3
2. Ensure that x_3 has only incoming edges
   ➤replace $r(x, y)$ by $r^-(y, x)$ as needed
3. Merge the predecessors
   ➤if x_3 is a leaf of a tree, they must be mapped together
4. Find an axiom that enforces an $(r \sqcap s^-)$-child that is C
   ➤fail if $\mathcal{T}$ does not imply such an axiom

One step of query rewriting

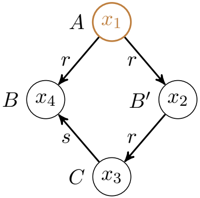


$$\mathcal{T} \models D \sqsubseteq \exists(r \sqcap s^-).C$$

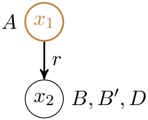


1. Select the non-distinguished variable x_3
2. Ensure that x_3 has only incoming edges
   - ➤replace $r(x, y)$ by $r^-(y, x)$ as needed
3. Merge the predecessors
   - ➤if x_3 is a leaf of a tree, they must be mapped together
4. Find an axiom that enforces an $(r \sqcap s^-)$-child that is C
   - ➤fail if $\mathcal{T}$ does not imply such an axiom
5. Drop x_3 and add $D(x_2)$

One step of query rewriting



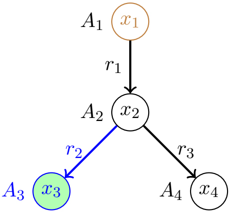
$\mathcal{T} \models D \sqsubseteq \exists(r \sqcap s^-).C$



1. Select the non-distinguished variable x_3
2. Ensure that x_3 has only incoming edges
   - ➤replace $r(x, y)$ by $r^-(y, x)$ as needed
3. Merge the predecessors
   - ➤if x_3 is a leaf of a tree, they must be mapped together
4. Find an axiom that enforces an $(r \sqcap s^-)$-child that is C
   - ➤fail if $\mathcal{T}$ does not imply such an axiom
5. Drop x_3 and add $D(x_2)$

Another step of query rewriting

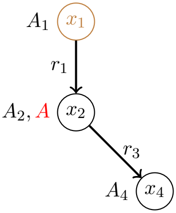
The query



using the axiom

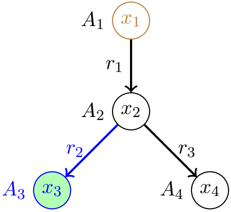
$$A \sqsubseteq \exists r_2.A_3$$

is rewritten to



Another step of query rewriting

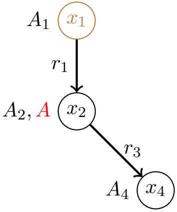
The query



using the axiom

$$A \sqsubseteq \exists r_2.A_3$$

is rewritten to



To handle transitive roles in the query:

- introduce a new variable between the eliminated variable and some of its predecessors
- eliminate sets of variables (due to variables connected in the query that may be mapped to the same object)

TBox Saturation

- We compute in advance a set $\Xi(\mathcal{T})$ of relevant axioms
  - Tailored resolution calculus for Horn-$\mathcal{ALCHIQ}^{\sqcap}$
  - Adaptation of existing *consequence driven* procedures for satisfiability

Example Rules

$$\frac{M \sqsubseteq \exists S.(N \sqcap N') \quad N \sqsubseteq A}{M \sqsubseteq \exists S.(N \sqcap N' \sqcap A)}\ \mathbf{R}^{c}_{\sqsubseteq}$$

$$\frac{M \sqsubseteq \exists (S \sqcap \mathsf{inv}(r)).(N \sqcap A) \quad A \sqsubseteq \forall r.B}{M \sqsubseteq B}\ \mathbf{R}^{-}_{\forall}$$

- The rewriting step simply searches for an axiom in $\Xi(\mathcal{T})$

The Query Answering Algorithm

Algorithm 1: Answering CQs via Query Rewriting

Input: normal Horn-$\mathcal{SHIQ}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, Conjunctive Query q
Output: query answers
$\Xi(\mathcal{T}) \leftarrow \texttt{Saturate}(\mathcal{T})$;
$\mathsf{rew}_{\mathcal{T}}(q) \leftarrow \texttt{Rewrite}(q, \Xi(\mathcal{T}))$;
$\mathsf{cr}(\mathcal{T}) \leftarrow \texttt{CompletionRules}(\mathcal{T})$;
$\mathcal{P} \leftarrow \mathcal{A} \cup \mathsf{cr}(\mathcal{T}) \cup \mathsf{rew}_{\mathcal{T}}(q)$;
$ans \leftarrow \{\vec{u} \mid q(\vec{u}) \in \texttt{MinimalModel}(\mathcal{P})\}$; $\triangleright$ `call Datalog reasoner`

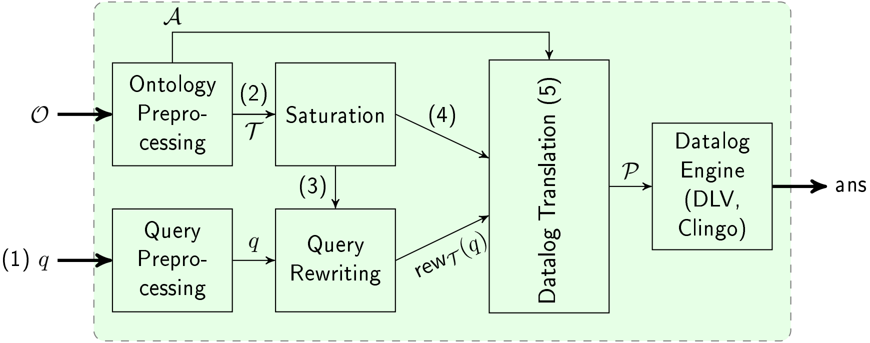
The completion rules $\mathsf{cr}(\mathcal{T})$ are straightforward, e.g.

$$B(y) \leftarrow A(x), r(x,y) \quad \text{for each } A \sqsubseteq \forall r.B \in \mathcal{T}$$

$$r(x,y) \leftarrow r_1(x,y), \ldots, r_n(x,y) \quad \text{for each } r_1 \sqcap \ldots \sqcap r_n \sqsubseteq r \in \mathcal{T}$$

System Architecture: Clipper

We have implemented a prototype system called Clipper (http://www.kr.tuwien.ac.at/research/systems/clipper)



(1) CQ with simple roles only

(2) Normalized TBox

(3) Existential axioms in $\Xi(\mathcal{T})$

(4) Axioms for ABox completion

(5) $\mathcal{P} \leftarrow \mathcal{A} \cup \mathsf{cr}(\mathcal{T}) \cup \mathsf{rew}_{\mathcal{T}}(q)$

Experiments

We carried out the following experiments:

- Downscaling test
  - We compared Clipper with state of the art query rewriting systems over $\mathcal{DL}$-*Lite* ontologies

Experiments

We carried out the following experiments:

- Downscaling test
  - We compared Clipper with state of the art query rewriting systems over $\mathcal{DL}$-*Lite* ontologies
- Full Horn-$\mathcal{SHIQ}$ test
  - We tested Clipper over a full Horn-$\mathcal{SHIQ}$ ontology

Downscaling test

- We used the ontologies of the Requiem test suite
  Adolena (A), Stock Exchange (S), Vicodi (V) and University (U)
- We added 2 queries to the 5 existing ones for each ontology
- We compared Clipper with Presto and Requiem
- We compared rewriting times and size of the rewritten queries
- For U, we also evaluated the resulting Datalog program over 4 ABoxes, 67k to 320k

Downscaling test (2)

$\mathcal{O}$	Query	# Rules/CQs			Rewriting time, ms (avg. eval. time, DLV)		
		RequiemG	Presto	Clipper	RequiemG	Presto	Clipper
A	Q1	27	53	42	281	45	50
	Q2	50	32	31	184	46	62
	Q3	104	32	31	292	27	65
	Q4	224	43	36	523	32	71
	Q5	624	37	36	1177	25	70
	Q6	364	35	30	523	31	65
	Q7	2548	43	32	7741	61	64
S	Q1	6	7	10	14	7	19
	Q2	2	3	22	263	9	22
	Q3	4	4	9	1717	10	21
	Q4	4	4	24	1611	9	23
	Q5	8	5	10	18941	10	22
	Q6	4	8	5	204	11	21
	Q7	8	6	7	1733	11	17
U	Q1	2	4	2	14 (1247)	12 (1252)	27 (1255)
	Q2	1	2	45	201 (1247)	23 (1262)	36 (1637)
	Q3	4	8	17	477 (2055)	26 (2172)	29 (1890)
	Q4	2	56	63	2431 (1260)	20 (1235)	28 (1735)
	Q5	10	8	16	7216 (1267)	26 (1305)	36 (1372)
	Q6	10	13	10	13 (1272)	14 (1260)	27 (1262)
	Q7	960	24	19	1890 (1730)	15 (1310)	35 (1322)

Full Horn-$\mathcal{SHIQ}$ test

- We modified the $\mathcal{SHOIN}(\mathrm{D})$ ontology UOBM
- We dropped or strengthened (in case of disjunctions) non-Horn-$\mathcal{SHIQ}$ TBox axioms
- The final ontology has 196 TBox axioms
- We used ABoxes with 20k, 80k, 140k and 200k assertions.
- 10 small, hand-tailored queries, with an average of 5 atoms

Full Horn-$\mathcal{SHIQ}$ test (2)

Query	# Rules	Rewriting Time (ms)	Datalog (DLV) Time (ms)
Q1	2	68	80 / 320 / 560 / 830
Q2	3	63	90 / 330 / 560 / 830
Q3	9	96	90 / 320 / 570 / 810
Q4	172	143	230 / 830 / 1430 / 1580
Q5	16	91	90 / 330 / 570 / 820
Q6	255	177	250 / 890 / 1530 / 1800
Q7	8	89	80 / 320 / 570 / 820
Q8	175	146	230 / 830 / 1430 / 1580
Q9	175	145	230 / 820 / 1400 / 1600
Q10	2	64	80 / 330 / 570 / 830

Conclusions

- We have proposed a query rewriting technique for Horn-$\mathcal{SHIQ}$
- It allows us to reduce CQs answering over Horn-$\mathcal{SHIQ}$ to evaluating a Datalog program over the ABox

Conclusions

- We have proposed a query rewriting technique for Horn-$\mathcal{SHIQ}$
- It allows us to reduce CQs answering over Horn-$\mathcal{SHIQ}$ to evaluating a Datalog program over the ABox
- The prototype system Clipper shows promising results

Conclusions

- We have proposed a query rewriting technique for Horn-$\mathcal{SHIQ}$
- It allows us to reduce CQs answering over Horn-$\mathcal{SHIQ}$ to evaluating a Datalog program over the ABox
- The prototype system Clipper shows promising results

➤Version with transitive roles may be available soon!

Conclusions

- We have proposed a query rewriting technique for Horn-$\mathcal{SHIQ}$
- It allows us to reduce CQs answering over Horn-$\mathcal{SHIQ}$ to evaluating a Datalog program over the ABox
- The prototype system Clipper shows promising results

  ➤Version with transitive roles may be available soon!
- Possible extensions:
  - weakly DL-safe rules (algorithm developed, implementation pending)
  - other DLs: regular $\mathcal{EL}^{++}$, Horn-$\mathcal{SRIQ}$; datatypes
  - other query languages: *regular path queries*

Conclusions

- We have proposed a query rewriting technique for Horn-$\mathcal{SHIQ}$
- It allows us to reduce CQs answering over Horn-$\mathcal{SHIQ}$ to evaluating a Datalog program over the ABox
- The prototype system Clipper shows promising results
  ➤Version with transitive roles may be available soon!
- Possible extensions:
  - weakly DL-safe rules (algorithm developed, implementation pending)
  - other DLs: regular $\mathcal{EL}^{++}$, Horn-$\mathcal{SRIQ}$; datatypes
  - other query languages: *regular path queries*
- Far more testing needed

Conclusions

- We have proposed a query rewriting technique for Horn-$\mathcal{SHIQ}$
- It allows us to reduce CQs answering over Horn-$\mathcal{SHIQ}$ to evaluating a Datalog program over the ABox
- The prototype system Clipper shows promising results

  ➤Version with transitive roles may be available soon!
- Possible extensions:
  - weakly DL-safe rules (algorithm developed, implementation pending)
  - other DLs: regular $\mathcal{EL}^{++}$, Horn-$\mathcal{SRIQ}$; datatypes
  - other query languages: *regular path queries*
- Far more testing needed

  but lack of realistic test cases is a big issue!