kyrie2: Query Rewriting under Extensional Constraints in $\mathcal{ELHIO}$

José Mora, Riccardo Rosati and Óscar Corcho

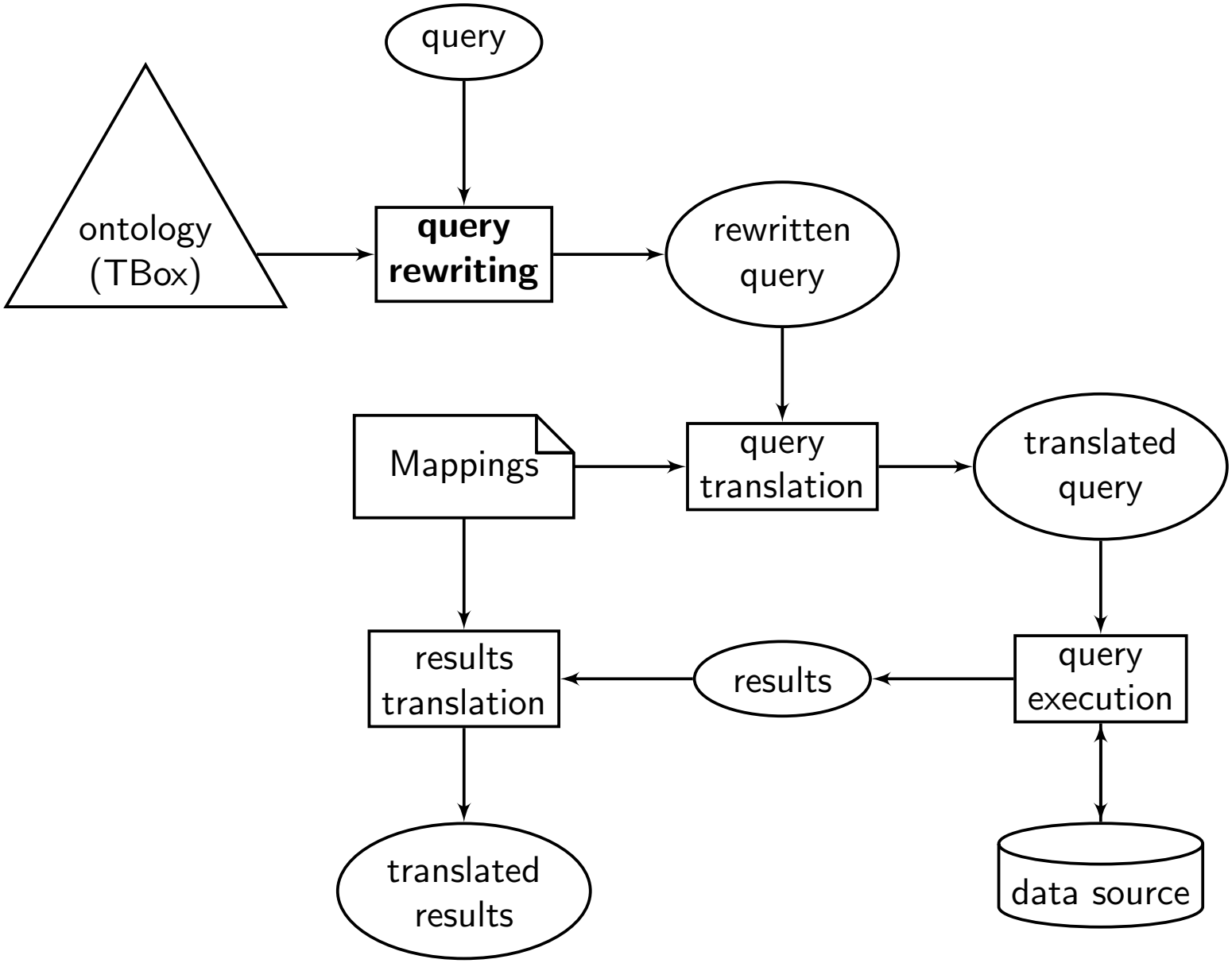
{mora, rosati}@dis.uniroma1.it, ocorcho@fi.upm.es

http://j.mp/eboxevaluation

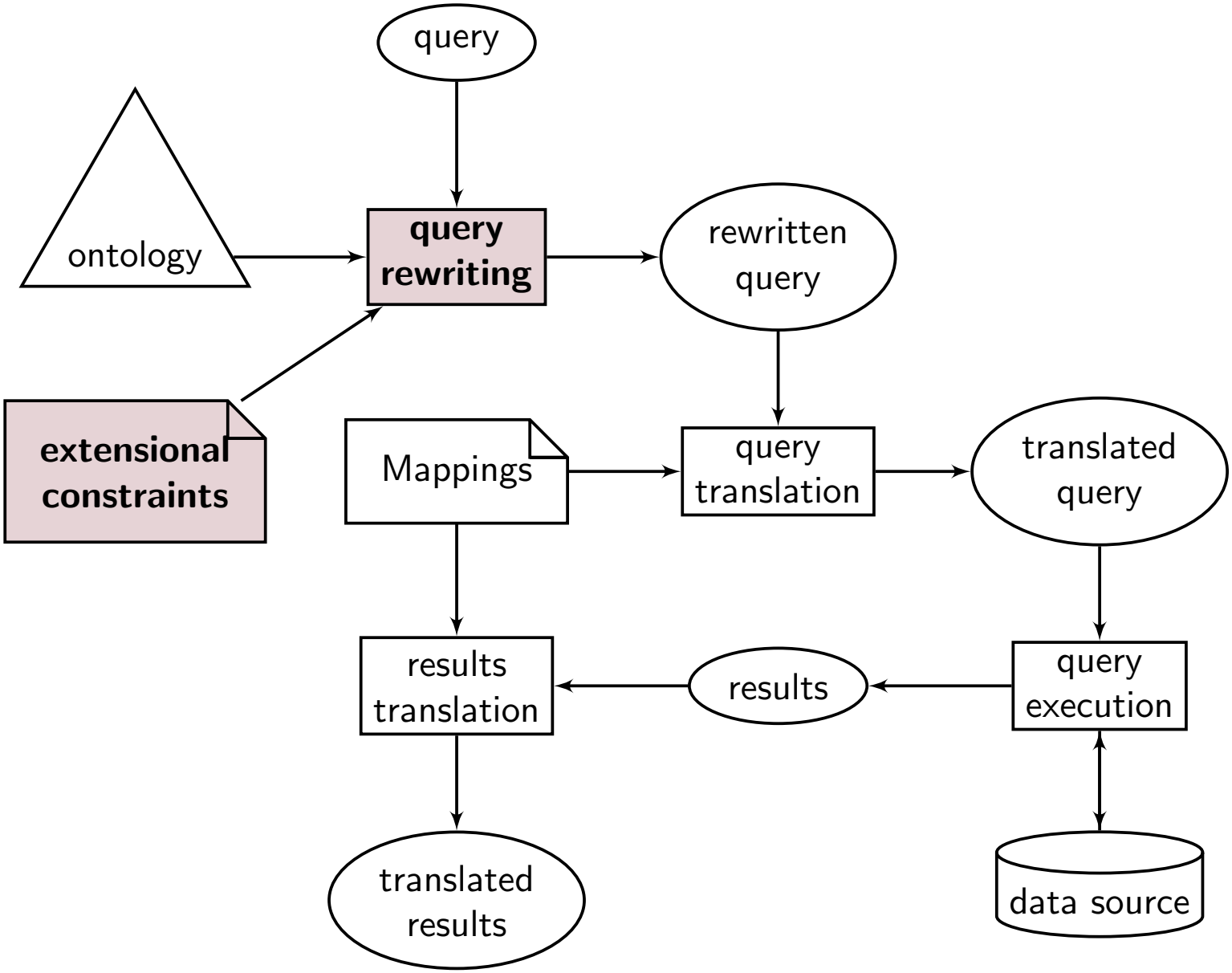
Roma - October 23, 2014

Index

Query rewriting in OBDA



Query rewriting in OBDA with extensional constraints



What are extensional constraints?

Extensional constraints or ABox dependencies[1] are:

- axioms that the data satisfies

What are extensional constraints?

Extensional constraints or ABox dependencies[1] are:

- axioms that the data satisfies
- as an "integrity constraint" on the ontology

What are extensional constraints?

Extensional constraints or ABox dependencies[1] are:

- axioms that the data satisfies
- as an "integrity constraint" on the ontology

These axioms:

- form an Extensional Box (EBox)[2]

What are extensional constraints?

Extensional constraints or ABox dependencies[1] are:

- axioms that the data satisfies
- as an "integrity constraint" on the ontology

These axioms:

- form an Extensional Box (EBox)[2]
- can be automatically derived[3]

What are extensional constraints?

Extensional constraints or ABox dependencies[1] are:

- axioms that the data satisfies
- as an "integrity constraint" on the ontology

These axioms:

- form an Extensional Box (EBox)[2]
- can be automatically derived[3]
- can be used to optimise query rewriting

What are extensional constraints?

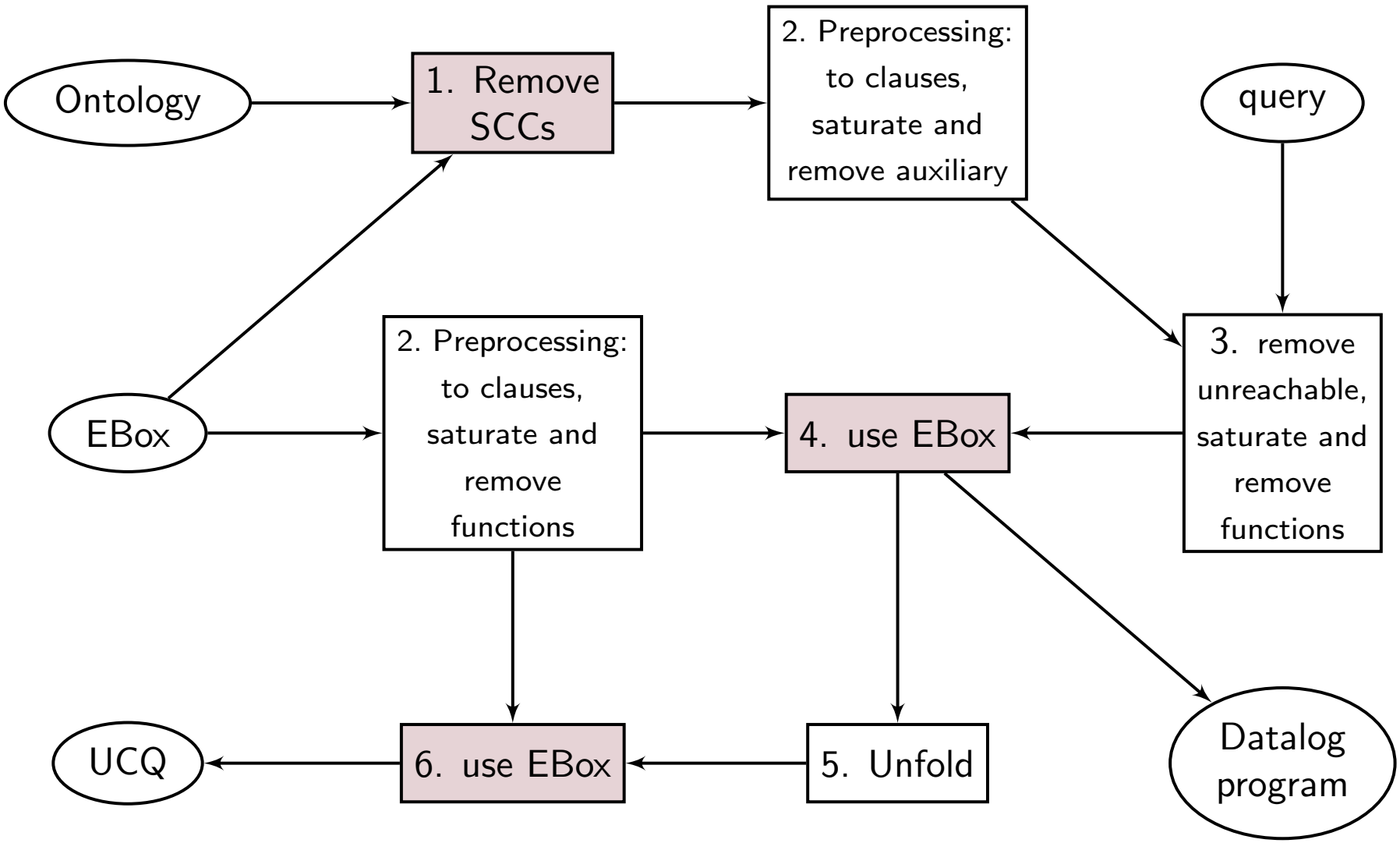
Extensional constraints or ABox dependencies [1] are:

- axioms that the data satisfies
- as an "integrity constraint" on the ontology

These axioms:

- form an Extensional Box (EBox) [2]
- can be automatically derived [3]
- can be used to optimise query rewriting on $\mathcal{ELHIO}$

Stages in kyrie2



Example – TBox and EBox

TBox

- Person
  - Professor
  - Student
    - UndergradStudent
    - GradStudent
      - MasterStudent
      - PhDStudent
- Course
  - PotentialCourse
  - $\exists hasProfessor.Professor$
    - $\exists hasStudent.Student$
      - ImpartedCourse

EBox

- PotentialCourse
  - ImpartedCourse
- Student
  - GradStudent
  - Bachelor
  - MasterStudent
  - PhDStudent
  - $\exists hasStudent^{-}$

TBox clauses - SCC removal

```
                AUX$0(?0)  :-  Student(?1), hasStudent(?0,?1)
              Course(?0)   :-  ImpartedCourse(?0)
              Course(?0)   :-  PotentialCourse(?0)
              Course(?0)   :-  Professor(?1), hasProfessor(?0,?1)
           GradStudent(?0) :-  MasterStudent(?0)
           GradStudent(?0) :-  PhDStudent(?0)
                Person(?0) :-  Professor(?0)
                Person(?0) :-  Student(?0)
         Professor(f1(?0)) :-  AUX$0(?0)
              Student(?0)  :-  Bachelor(?0)
              Student(?0)  :-  GradStudent(?0)
           Student(f0(?0)) :-  ImpartedCourse(?0)
hasProfessor(?0,f1(?0))    :-  AUX$0(?0)
  hasStudent(?0,f0(?0))    :-  ImpartedCourse(?0)
```

TBox clauses - SCC removal

```
                AUX$0(?0)  :-  Student(?1), hasStudent(?0,?1)
               Course(?0)  :-  ImpartedCourse(?0)
               Course(?0)  :-  PotentialCourse(?0)
               Course(?0)  :-  Professor(?1), hasProfes
           GradStudent(?0)  :-  MasterStudent(?0)
           GradStudent(?0)  :-  PhDStudent(?0)
               Person(?0)  :-  Professor(?0)
               Person(?0)  :-  Student(?0)
          Professor(f1(?0))  :-  AUX$0(?0)
--             Student(?0)  :-  Bachelor(?0)
               Student(?0)  :-  GradStudent(?0)
          Student(f0(?0))  :-  ImpartedCourse(?0)
hasProfessor(?0,f1(?0))  :-  AUX$0(?0)
  hasStudent(?0,f0(?0))  :-  ImpartedCourse(?0)
```

EBox

PotentialCourse

- ImpartedCourse

Student

- GradStudent
- Bachelor
- MasterStudent
- PhDStudent
- $\exists hasStudent^-$

Datalog program - use the EBox

```
          Q(?0)  :-  Course(?0)
     Course(?0)  :-  ImpartedCourse(?0)
     Course(?0)  :-  PotentialCourse(?0)
     Course(?0)  :-  Professor(?1), hasProfessor(?0,?1)
     Course(?0)  :-  Student(?1), hasStudent(?0,?1)
GradStudent(?0)  :-  MasterStudent(?0)
GradStudent(?0)  :-  PhDStudent(?0)
    Student(?0)  :-  GradStudent(?0)
```

Datalog program - use the EBox

```
                Q(?0)  :-  Course(?0)
--          Course(?0)  :-  ImpartedCourse(?0)
            Course(?0)  :-  PotentialCourse(?0)
            Course(?0)  :-  Professor(?1), hasProfess
            Course(?0)  :-  Student(?1), hasStudent(?
      GradStudent(?0)   :-  MasterStudent(?0)
      GradStudent(?0)   :-  PhDStudent(?0)
           Student(?0)  :-  GradStudent(?0)
```

EBox

PotentialCourse

- ImpartedCourse

Student

- GradStudent
- Bachelor
- MasterStudent
- PhDStudent
- $\exists hasStudent^-$

Unfolded UCQ - use the EBox

```
Q(?0)  :-  Course(?0)
Q(?0)  :-  GradStudent(?1), hasStudent(?0,?1)
Q(?0)  :-  MasterStudent(?1), hasStudent(?0,?1)
Q(?0)  :-  PhDStudent(?1), hasStudent(?0,?1)
Q(?0)  :-  PotentialCourse(?0)
Q(?0)  :-  Professor(?1), hasProfessor(?0,?1)
Q(?0)  :-  Student(?1), hasStudent(?0,?1)
```

Unfolded UCQ - use the EBox

```
      Q(?0)   :-   Course(?0)
--    Q(?0)   :-   GradStudent(?1), hasStudent(?0
--    Q(?0)   :-   MasterStudent(?1), hasStudent(
--    Q(?0)   :-   PhDStudent(?1), hasStudent(?0,
      Q(?0)   :-   PotentialCourse(?0)
      Q(?0)   :-   Professor(?1), hasProfessor(?0
--    Q(?0)   :-   Student(?1), hasStudent(?0,?1)
++    Q(?0)   :-   hasStudent(?0,?1)
```

EBox

PotentialCourse

- ImpartedCourse

Student

- GradStudent
- Bachelor
- MasterStudent
- PhDStudent
- $\exists hasStudent^-$

Evaluation set up

We want to check whether EBoxes may be used to reduce

- the size of the rewritten queries and
- the time for query rewriting

Evaluation set up

We want to check whether EBoxes may be used (hypothesis: yes) to reduce

- the size of the rewritten queries and
- the time for query rewriting

To do this:

- synthetic EBoxes to extend a previous benchmark ([4], 11 ontologies, 5 to 9 queries each)
- EBoxes are generated with three parameters

size	EBox size wrt TBox size ($[0 - \infty]$)
	Impact of the EBox on the results
cover	part of the TBox covered by the EBox ($[0 - 1]$)
	Impact of similarity (vs randomness) on the results
reverse	axioms in the cover that get reversed (LHS $\rightleftharpoons$ RHS) $[0 - 1]$
	Impact of redundancy, upside-down vs downside-up

- $size = 0 \rightarrow cover = 0$
- $cover = 0 \rightarrow reverse = 0$
- values used: $0, 0.2, 0.4, 0.6, 0.8$ and 1, total 156 EBoxes for each ontology

Results

Query independent information					query	Datalog time(ms)				Datalog size				UCQ time(ms)				UCQ size			
EBox	I	II	III	IV		I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
PT	109	2047	24266	2859	1	0	0	157	235	15	13	14	9	0	0	516	672	15	13	14	9
PS	222	195	171	111	2	16	16	157	234	10	10	10	10	16	16	500	656	10	10	10	10
size	0.0	0.2	0.8	0.8	3	0	0	125	235	35	30	28	15	31	15	485	813	72	57	54	15
cover	0.0	0.8	0.2	0.8	4	0	16	219	188	41	38	22	16	63	94	735	719	185	170	3	42
rev	0.0	0.0	0.0	0.0	5	16	16	172	250	8	5	7	1	32	31	609	719	30	9	15	1
PT: preprocess time (ms)					6	0	15	234	188	18	14	14	11	0	15	578	641	18	14	14	11
PS: preprocessed size					7	0	0	125	172	27	23	23	20	94	125	1359	1359	180	140	140	110

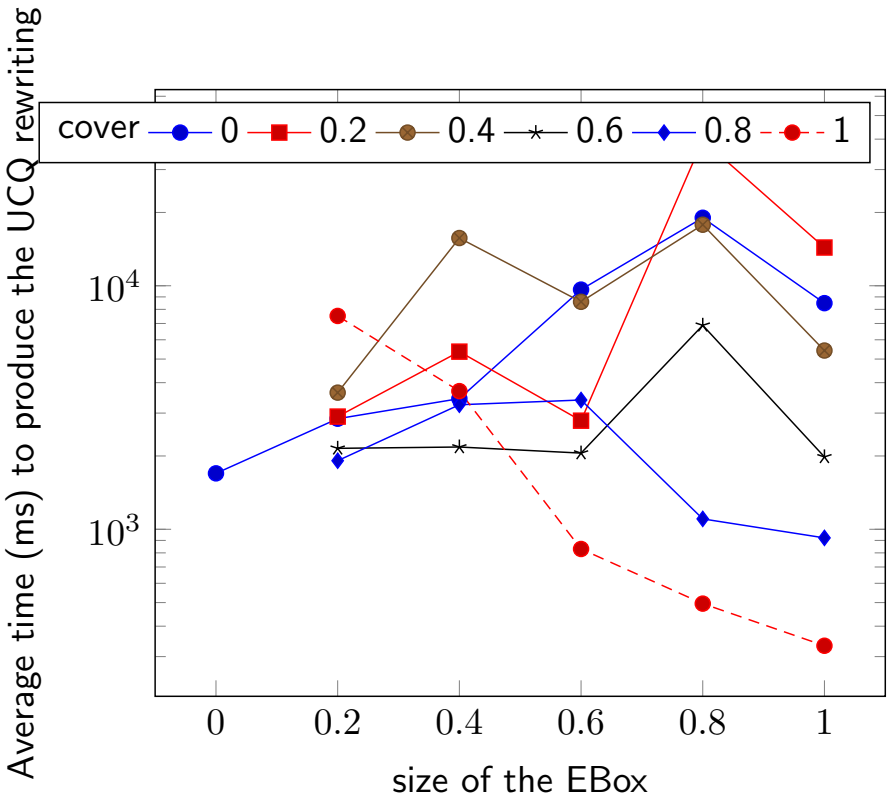
Table: Results obtained for ontology V (original size 222 clauses) with EBoxes I, II, III and IV. Full results at `http://j.mp/eboxevaluation`

Results

Query independent information					query	Datalog time(ms)				Datalog size				UCQ time(ms)				UCQ size			
EBox	I	II	III	IV		I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
PT	109	2047	24266	2859	1	0	0	157	235	15	13	14	9	0	0	516	672	15	13	14	9
PS	222	195	171	111	2	16	16	157	234	10	10	10	10	16	16	500	656	10	10	10	10
size	0.0	0.2	0.8	0.8	3	0	0	125	235	35	30	28	15	31	15	485	813	72	57	54	15
cover	0.0	0.8	0.2	0.8	4	0	16	219	188	41	38	22	16	63	94	735	719	185	170	3	42
rev	0.0	0.0	0.0	0.0	5	16	16	172	250	8	5	7	1	32	31	609	719	30	9	15	1
PT: preprocess time (ms)					6	0	15	234	188	18	14	14	11	0	15	578	641	18	14	14	11
PS: preprocessed size					7	0	0	125	172	27	23	23	20	94	125	1359	1359	180	140	140	110

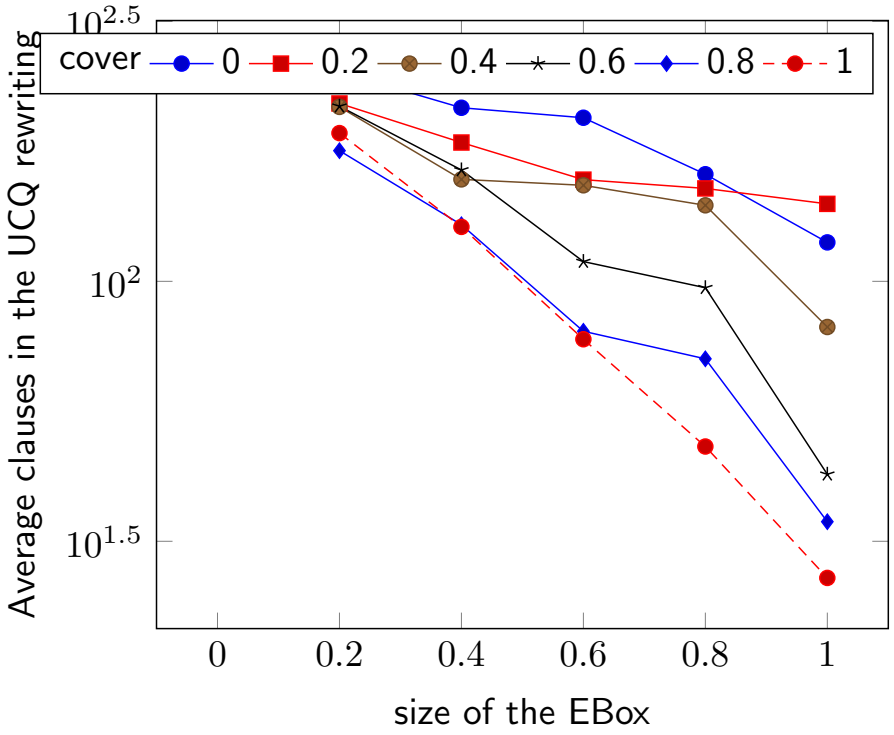
Table: Results obtained for ontology V (original size 222 clauses) with EBoxes I, II, III and IV. Full results at `http://j.mp/eboxevaluation`

Results summary – Time to UCQ



- Time needed to produce the UCQ rewriting
- All queries and ontologies averaged
- Less time needed to produce the rewriting if
  - the EBox is big and
  - the EBox is similar to the TBox
- Please note the logarithmic scale

Results summary – UCQ number of clauses



- Number of clauses produced in the UCQ rewriting
- All queries and ontologies averaged
- Shorter rewritings produced when
  - the EBox is big and
  - the EBox is similar to the TBox
- Please note the logarithmic scale

Conclusions

- Significant reductions in rewritten query size are achievable
- Query rewriting time can be reduced, especially when most needed
- Greater EBoxes have a greater impact
- EBoxes similar to TBoxes have greater impact
- Random EBoxes are less predictable (depends on the query)

Conclusions and future lines

Conclusions:

- Significant reductions in rewritten query size are achievable
- Query rewriting time can be reduced, especially when most needed
- Greater EBoxes have a greater impact
- EBoxes similar to TBoxes have greater impact
- Random EBoxes are less predictable (depends on the query)

Future lines:

- Extending to Datalog$\pm$
- Considering SPARQL
- More and better experiments

kyrie2: Query Rewriting under Extensional Constraints in $\mathcal{ELHIO}$

José Mora, Riccardo Rosati and Óscar Corcho

{mora, rosati}@dis.uniroma1.it, ocorcho@fi.upm.es

http://j.mp/eboxevaluation

Roma - October 23, 2014

"Removability"

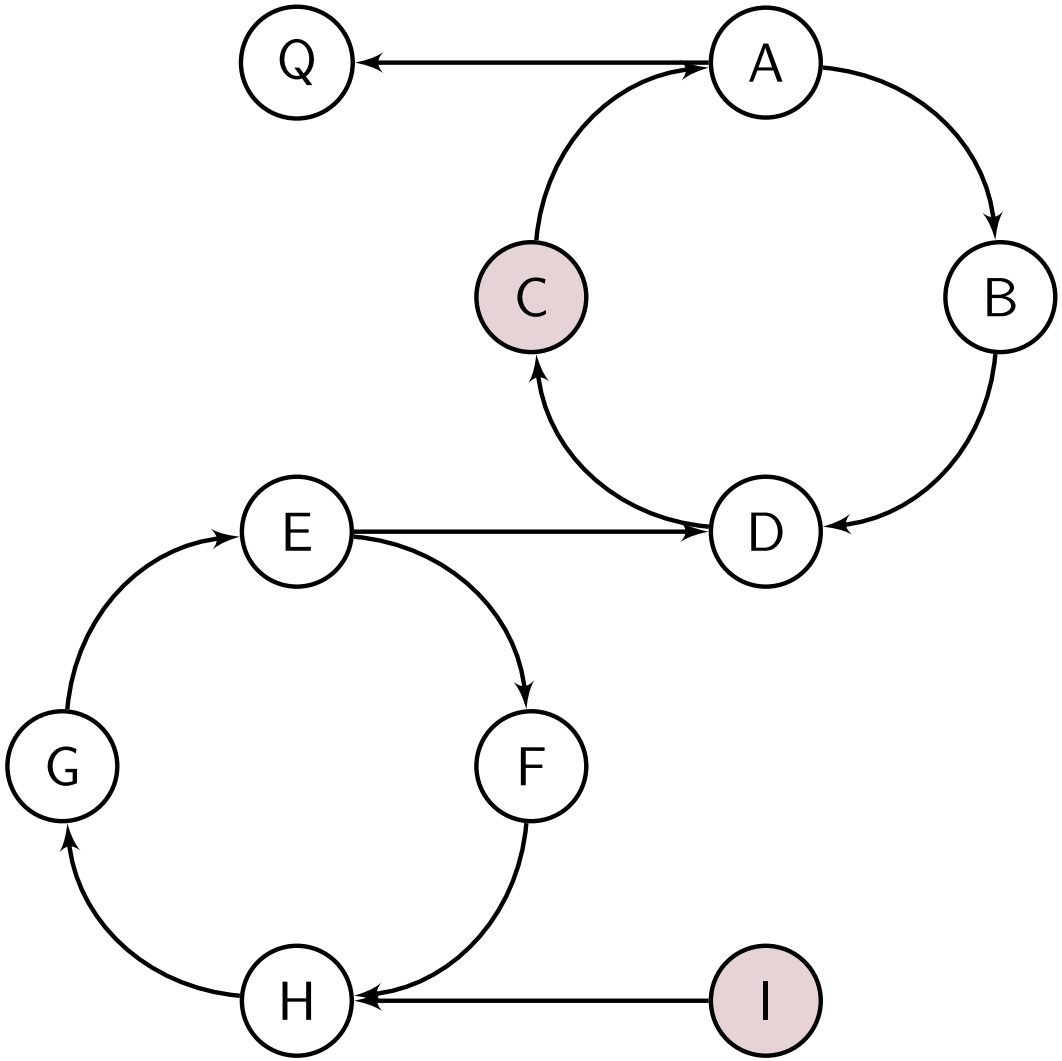
Let Γ be an OBDA system or a Datalog program.

Let E_e be the EBox after removing the predicates with some intensional definition in Γ.

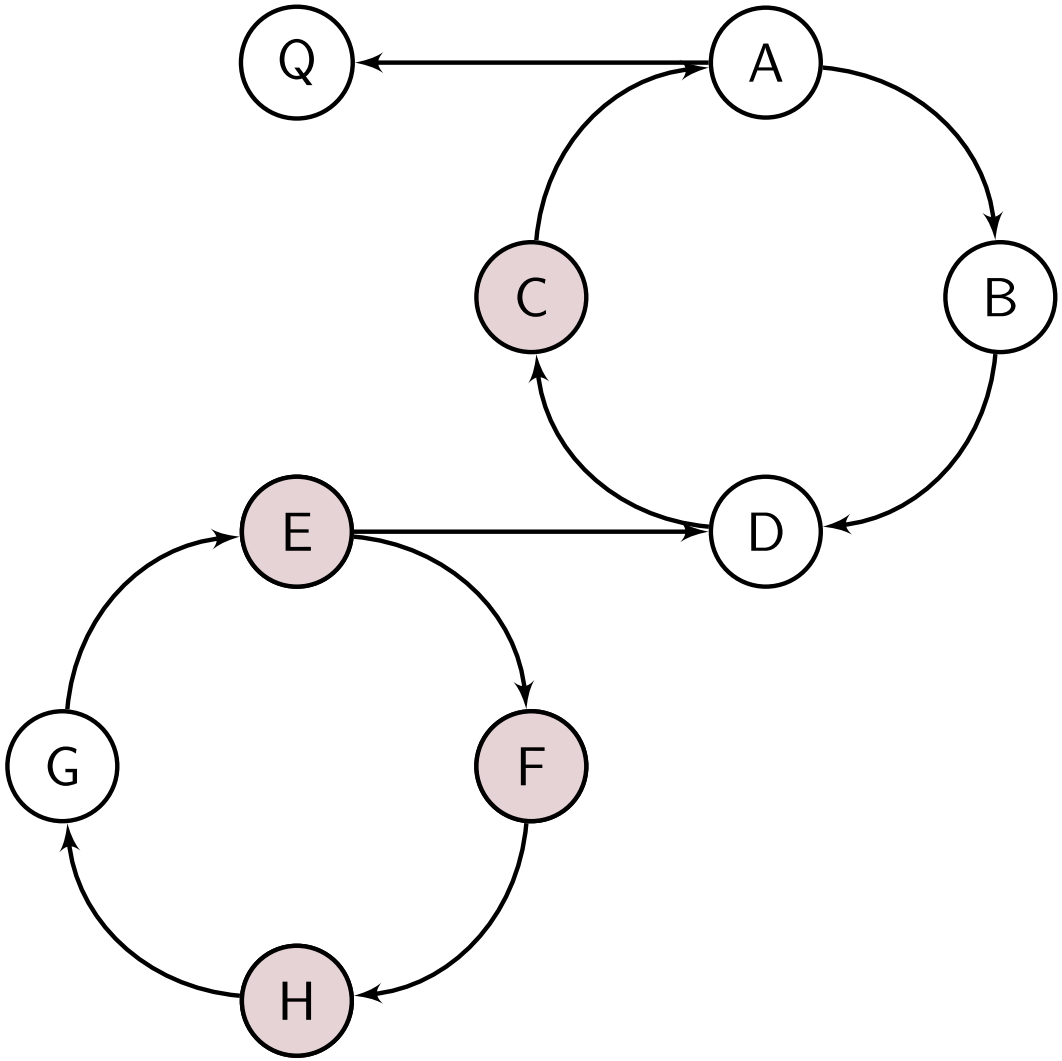
For any clause γ_e such that $\Gamma \cup E_e \vdash \gamma_e$ and any clause $\gamma \in \Gamma$:

- If $\gamma_e = \gamma$ then we can remove γ from Γ
- If γ_e subsumes γ then we can replace γ with γ_e
  (thus removing from Γ all other clauses subsumed by γ_e)

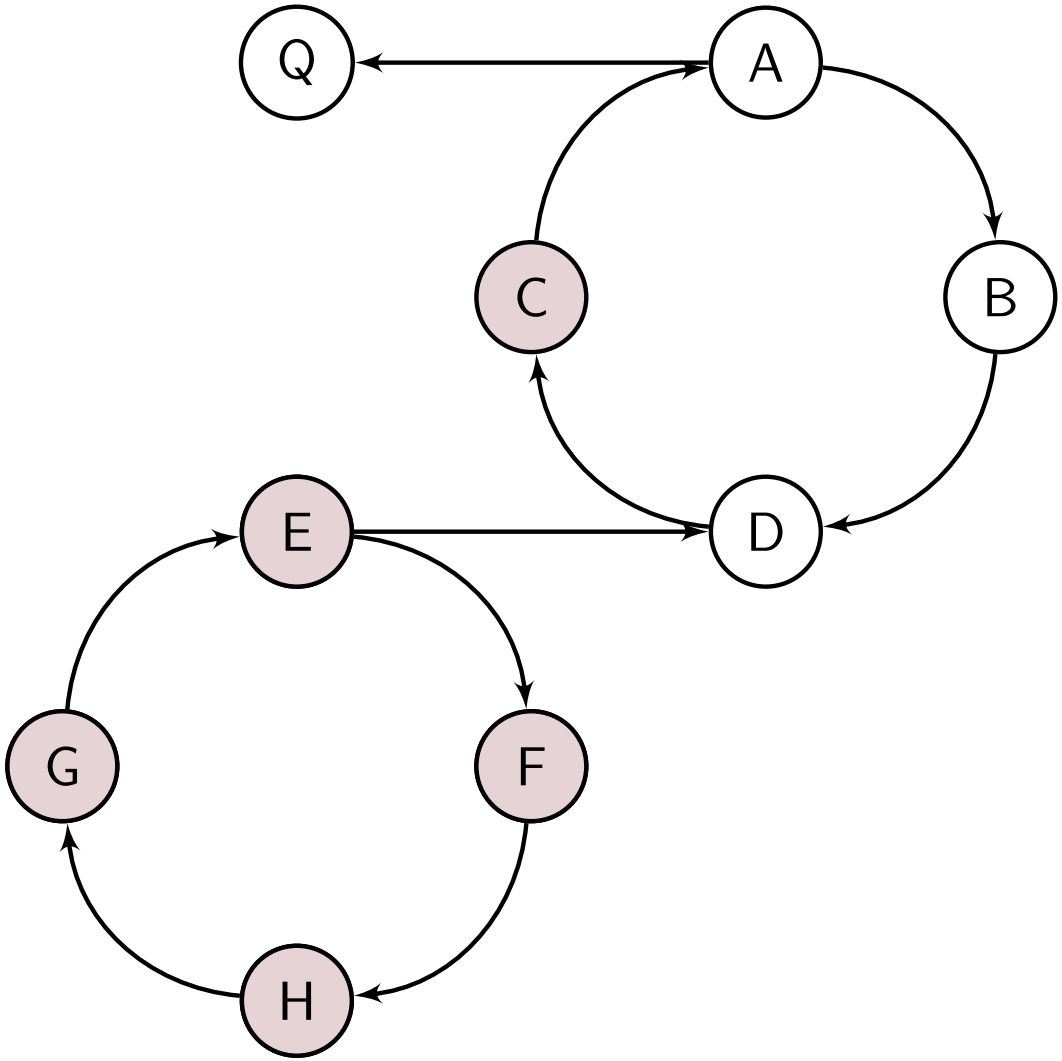
Strongly connected components (of clauses or axioms)



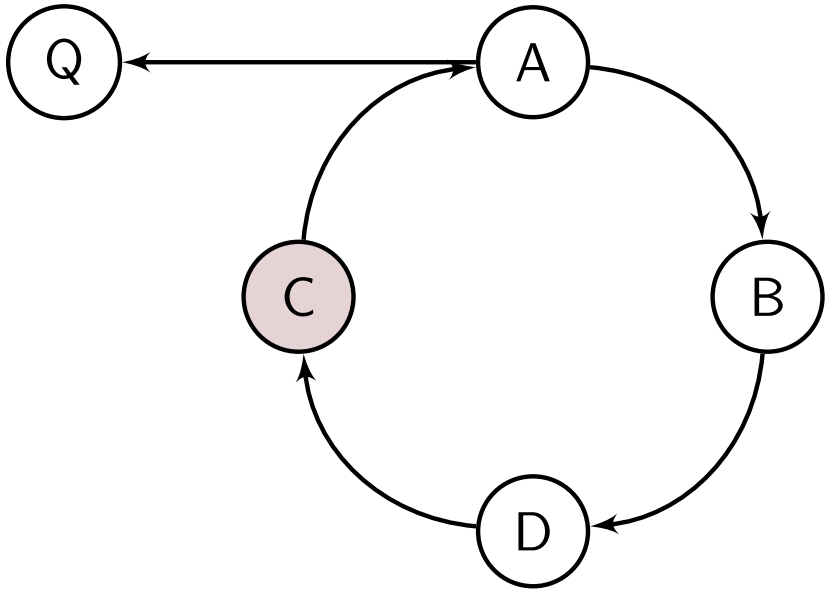
Strongly connected components (of clauses or axioms)



Strongly connected components (of clauses or axioms)



Strongly connected components (of clauses or axioms)



References I

[1] M. Rodríguez-Muro and D. Calvanese. Dependencies: Making ontology based data access work in practice. Proc. of the 5th Alberto Mendelzon Int. Workshop on Foundations of Data Management, 2011. URL http://web.inf.unibz.it/~calvanese/papers/rodr-calv-AMW-2011.pdf.

[2] Riccardo Rosati. Prexto: Query rewriting under extensional constraints in DLLite. In Elena Simperl, Philipp Cimiano, Axel Polleres, Oscar Corcho, and Valentina Presutti, editors, The Semantic Web: Research and Applications, volume 7295 of Lecture Notes in Computer Science, pages 360–374. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-30283-1. URL http://www.springerlink.com/content/1j61g52j78525175/abstract/.

[3] Marco Console, Maurizio Lenzerini, Riccardo Mancini, Riccardo Rosati, and Marco Ruzzi. Synthesizing extensional constraints in ontology-based data access. In Description Logics, pages 628–639, 2013.

[4] Jose Mora and Oscar Corcho. Towards a systematic benchmarking of ontology-based query rewriting systems. In The 12th International Semantic Web Conference (ISWC2013), pages 369–384, Sydney, Australia, 2013. URL http://data.semanticweb.org/conference/iswc/2013/proceedings-2/paper-24.

kyrie2: Query Rewriting under Extensional Constraints in $\mathcal{ELHIO}$

José Mora, Riccardo Rosati and Óscar Corcho

{mora, rosati}@dis.uniroma1.it, ocorcho@fi.upm.es

http://j.mp/eboxevaluation

Roma - October 23, 2014